



**Hochschule für Technik  
und Wirtschaft Berlin**

**University of Applied Sciences**

**Dokumentation der Softwareentwicklung für die  
Gfal Dongle-verwaltungssoftware**

*Autor des Dokuments: Houssem Hfasa, Yassin Sahnoun, Rama Abazied*

*Ort, Datum Berlin 09.07.2023*

## ***Inhalt***

1. Einleitung
2. Anforderungen
3. Architektur
4. Datenmodell
5. Benutzerhandbuch
6. Installationshandbuch
7. Teststrategie und Testfälle
8. API-Dokumentation
9. Frontend-Dokumenation
10. Glossar
11. Lizenz

## **Einleitung**

Die vorliegende Dokumentation beschreibt das Tool zur Verwaltung von Dongle-Lizenzen. Das Tool hat zum Ziel, die Verwaltung von Lizenzen auf Hardware-Dongles zu vereinfachen und einen standardisierten Workflow für die Registrierung und Verwaltung von Dongles und Lizenzen bereitzustellen. Es soll die bestehende Excel-Tabelle als Lösung ablösen und eine effiziente und sichere Verwaltung der Dongle und ihrer Lizenzen ermöglichen.

# Anforderungen

## 1. Visionen und Ziele:

- Das Tool soll die Verwaltung von Lizenzen auf Hardware-Dongles vereinfachen.
- Es soll einen standardisierten Workflow für die Registrierung und Verwaltung von Dongles und Lizenzen bieten.
- Benachrichtigungen sollen an Benutzer für zeitnah ablaufende Lizenzen gesendet werden.

## 2. Rahmenbedingungen:

- Die Softwareentwicklung kann in einer Entwicklungsumgebung nach Wahl erfolgen.
- Das Tool wird als Webanwendung umgesetzt.
- Es basiert auf einer MySQL Datenbank.
- Es verwendet eine Client-Server-Architektur mit eigener Nutzerverwaltung.
- Kostenfreie Bibliotheken werden vorrangig zur Einbindung von externen Funktionen genutzt.

## 3. Kontext und Überblick:

- Das Tool bildet den Prozess der Erstellung und Verwaltung von Dongles und ihren Lizenzen ab.
- Das Tool bietet eine Gesamtübersicht aller Dongles und eine detaillierte Ansicht der Lizenzen auf einem Dongle.

## 4. Funktionale Anforderungen:

- Das Tool verwaltet und speichert die Daten in einer zentralen Datenbank.
- Es implementiert eine sichere Authentifizierungsmethode für den Zugriff auf autorisierte Benutzer.

- Benutzern wird eine übersichtliche Gesamtübersicht seiner vorhandenen Dongles/Lizenzen angezeigt.
- Die Funktionen und Ansichten des Tools sind rollenbasiert, mit verschiedenen Rechten für Benutzer, Dongle-Verwalter und Administratoren.
- Dongle-Verwalter können Dongle- und Lizenzdatensätze verwalten, bearbeiten und hinzufügen.
- Der Administrator hat erweiterte Berechtigungen zur Benutzerverwaltung und Sicherstellung der Systemsicherheit.

## **5. Nichtfunktionale Anforderungen (Leistungen, Daten):**

- Die Gestaltung der Eingabefelder richtet sich nach dem Datentyp.
- Die Benutzeroberfläche unterstützt verschiedene Bildschirmauflösungen.
- Die Passwortverschlüsselung nutzt vorhandene Algorithmen .

## **6. Qualitätsanforderungen:**

- Benutzerfreundlichkeit und Nutzbarkeit: Das System bietet eine benutzerfreundliche Oberfläche, die einfach zu bedienen und intuitiv ist.
- Sicherheit: Das System schützt Daten vor unbefugtem Zugriff durch geeignete Sicherheitsmaßnahmen wie Verschlüsselung, Authentifizierung und Autorisierung.
- Skalierbarkeit und Erweiterbarkeit: Das System ist skalierbar, um wachsenden Nutzerzahlen und größeren Datenvolumen gerecht zu werden. Es ermöglicht eine hohe Performance und ist leicht erweiterbar für zukünftige Anforderungen.
- Wartbarkeit und Dokumentation: Der Quellcode ist nachvollziehbar und gut strukturiert. Die Dokumentation enthält klare Erläuterungen des Systems, Anleitungen zur Installation, Konfiguration und Wartung sowie Beschreibungen der Komponenten und deren Funktionalität.

## Architektur

### Architekturübersicht:

Die Gfal Dongle-Verwaltungssoftware basiert auf einer mehrschichtigen Client-Server-Architektur. Sie besteht aus einem Front-End, das als Webanwendung implementiert ist, einem Back-End, das die Geschäftslogik und Datenverwaltung enthält, und einer Datenbank zur Speicherung der Dongle- und Lizenzdaten.

### Komponenten und Module:

- Front-End: Das Front-End der Webanwendung wird mit den Webtechnologien HTML, CSS und JavaScript entwickelt. Für die Implementierung der Benutzeroberfläche und der interaktiven Funktionen wird das Webapplikationsframework React verwendet.
- Back-End: Das Back-End wird mit Python entwickelt und beinhaltet die Geschäftslogik, Datenverarbeitung und Datenbankverbindung. Es werden Python-Bibliotheken wie Django und Django REST Framework verwendet, um die Entwicklung zu erleichtern und eine effiziente API-Verbindung zur Datenbank herzustellen. Die API-Zugriffe laufen auf einem separaten Server.
- Datenbank: Zur Speicherung und Verwaltung der Dongle- und Lizenzdaten wird eine MySQL-Datenbank verwendet. Für die Datenbankverbindung wird `django.db.backends.mysql` verwendet.

**Datenflüsse und Schnittstellen:** Die Kommunikation zwischen dem Front-End und dem Back-End erfolgt über eine RESTful API, die auf JSON basiert. Das Front-End sendet Anfragen an die API, um Daten abzurufen, zu erstellen, zu bearbeiten oder zu löschen. Das Back-End verarbeitet die Anfragen, führt die entsprechenden Aktionen durch und liefert die entsprechenden Daten als JSON-Antwort zurück.

### Verwendung von Frameworks und Bibliotheken:

- Front-End: React( HTML, CSS, JavaScript)
- Back-End: Django(Django REST Framework, Python)
- Datenbankverbindung: MySQL
- Verschlüsselung: Bibliothek `secret`
- Benachrichtigungen: SMTP

## Datenmodell

### Dongle

**Beschreibung:** Diese Tabelle speichert Informationen über die Dongles.

**Attribute:**

- Lfd. Nr. (Primärschlüssel)
- Serien-Nr
- Name
- Gültig von
- Gültig bis
- Projekt/Produkt
- Kunde
- Standort
- Händler
- Datum letzte Änderung
- Datum Erstausgabe
- Firmcode

### Lizenz

**Beschreibung:** Diese Tabelle speichert Informationen über die Lizenzen.

**Attribute:**

- Lfd. Nr. (Primärschlüssel)
- FirmCode
- ProductCode
- LizenzName
- Einheiten
- Gültig von
- Gültig bis
- LizenzAnzahl
- Mitarbeiter
- Projekt
- Kunde
- Dongle\_Serien-Nr

### Ticket

**Beschreibung:** Diese Tabelle speichert Informationen über die Tickets bzw. Die Anfrage, die eine Kunde anfordert.

**Attribute:**

- ID\_Ticket (Primärschlüssel)
- Titel
- Beschreibung
- Status
- Erstellungsdatum
- Schließungsdatum
- Dongle/Lizenz
- Dongle\_Name
- Dongle\_seriennummer
- LizenzName
- Firmcode
- Gültig von
- Gültig bis
- Einheiten
- Projekt
- Grund\_der\_Ablehnung
- Admin/Verwalter\_Email
- Händler
- Standort
- Productcode
- Lizenzanzahl

### **UserLoginCustomuser**

**Beschreibung:** Diese Tabelle speichert Informationen über den Benutzer.

**Attribute:**

- id (Primärschlüssel)
- last\_login
- is\_superuser
- email
- name
- password
- role
- firm\_code

### **Beziehungen**

#### **Datenbankdesign und -schema:**

Das Datenbankschema basiert auf den bereitgestellten Django-Modellen. Jede Tabelle hat einen Primärschlüssel, der eindeutige Werte für jede Zeile gewährleistet. Die



Beziehungen zwischen den Tabellen sind durch Fremdschlüssel definiert, die auf die entsprechenden Primärschlüssel in anderen Tabellen verweisen.

- Beziehung zwischen "Dongle" und "Lizenz":

Ein Dongle kann mehrere Lizenzen haben (One-to-Many-Beziehung). Die Tabelle "Lizenz" enthält eine Fremdschlüsselbeziehung zu "Dongle" über das Feld "Dongle\_Serien-Nr".

- Beziehung zwischen "Dongle" und "Ticket":

Ein Dongle kann in mehreren Tickets referenziert werden (One-to-Many-Beziehung). Sie könnten eine Fremdschlüsselbeziehung in der Tabelle "Ticket" hinzufügen, die auf den Primärschlüssel "Lfd. Nr." in der Tabelle "Dongle" verweist.

- Beziehung zwischen "Lizenz" und "Ticket":

Eine Lizenz kann in mehreren Tickets referenziert werden (One-to-Many-Beziehung). Sie könnten eine Fremdschlüsselbeziehung in der Tabelle "Ticket" hinzufügen, die auf den Primärschlüssel "Lfd. Nr." in der Tabelle "Lizenz" verweist.

- Beziehung zwischen "UserLogginCustomuser" und "Ticket":

Ein Benutzer kann mehrere Tickets erstellt haben (One-to-Many-Beziehung). Sie könnten eine Fremdschlüsselbeziehung in der Tabelle "Ticket" hinzufügen, die auf den Primärschlüssel "id" in der Tabelle "UserLogginCustomuser" verweist.

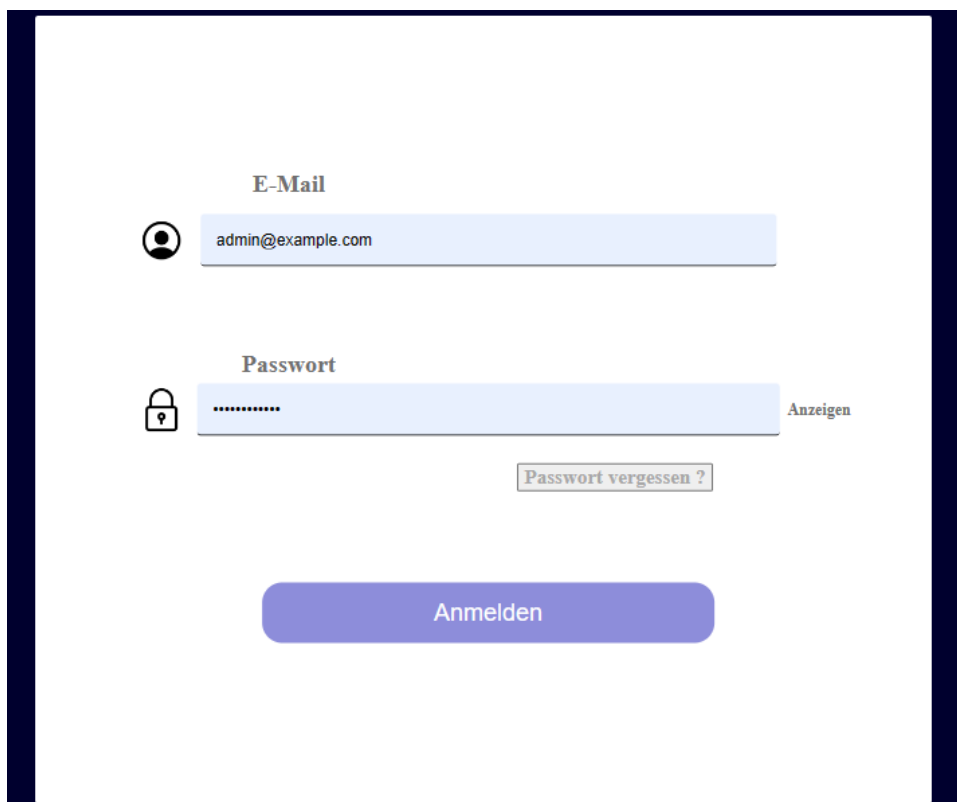
- Beziehung zwischen "UserLogginCustomuser" und "Dongle":

Ein Benutzer kann mehrere Dongles besitzen (One-to-Many-Beziehung). Sie könnten eine Fremdschlüsselbeziehung in der Tabelle "Dongle" hinzufügen, die auf den Primärschlüssel "id" in der Tabelle "UserLogginCustomuser" verweist.

## Benutzerhandbuch

### Anmeldung:

- Öffnen Sie die Login-Seite des Systems.
- Geben Sie Ihre E-Mail-Adresse und Ihr Passwort in die entsprechenden Felder ein.
- Klicken Sie auf "Anmelden", um sich einzuloggen.





The image shows a login form within a dark blue border. At the top, the text "E-Mail" is centered. Below it is a light blue input field containing the email address "admin@example.com", preceded by a user icon. Below the email field, the text "Passwort" is centered. Below it is a light blue input field containing a series of dots, preceded by a lock icon. To the right of the password field is a small link labeled "Anzeigen". Below the password field is a link labeled "Passwort vergessen ?". At the bottom of the form is a large, rounded purple button labeled "Anmelden".

### Dongle anfordern:

- Nach dem erfolgreichen Login befinden Sie sich auf der Übersichtsseite.
- Klicken Sie auf den Button "Dongle anfordern".
- Füllen Sie das Formular mit den erforderlichen Informationen aus.
- Klicken Sie auf "Speichern", um Ihre Anfrage abzuschicken.
- Ihre Anfrage wird an den Administrator gesendet und auf Genehmigung oder Ablehnung überprüft.



## Dongle anfordern

Dongle-ID (Seriennummer)	<input type="text" value="Dongle-ID (Seriennummer)"/>	Titel	<input type="text" value="Titel"/>
Beschreibung	<input type="text" value="Beschreibung"/>	Dongle-Name	<input type="text" value="Dongle-Name"/>
Händler	<input type="text" value="Händler"/>	Standort	<input type="text" value="Standort"/>
Gültig von	<input type="text" value="TT.MM.JJJJ"/> 	Gültig bis	<input type="text" value="TT.MM.JJJJ"/> 
FirmCode	<input type="text" value="FirmCode"/>	Projekt	<input type="text" value="Projekt"/>
<input type="button" value="Stornieren"/> <input type="button" value="Speichern"/>			

### Lizenz anfordern:

- Öffnen Sie die Lizenzübersichtsseite.
- Klicken Sie auf den Button "Lizenz anfordern".
- Füllen Sie das Formular mit den erforderlichen Informationen aus.
- Klicken Sie auf "Speichern", um Ihre Anfrage abzuschicken.
- Ihre Anfrage wird an den Administrator gesendet und auf Genehmigung oder Ablehnung überprüft.



## Lizenz anfordern

Dongle-ID(Seriennummer)	<input type="text" value="Dongle-ID (Seriennummer)"/>	Lizenzname	<input type="text" value="Lizenzname"/>
Titel	<input type="text" value="Titel"/>	Beschreibung	<input type="text" value="Beschreibung"/>
Einheiten	<input type="text" value="Einheiten"/>	ProductCode	<input type="text" value="ProductCode"/>
Gültig von	<input type="text" value="TT.MM.JJJJ"/> 	Gültig bis	<input type="text" value="TT.MM.JJJJ"/> 
Projekt	<input type="text" value="Projekt"/>	FirmCode	<input type="text" value="FirmCode"/>
Lizenzanzahl	<input type="text" value="Lizenzanzahl"/>	<input type="button" value="Stornieren"/> <input type="button" value="Speichern"/>	

### Admin-Funktionalität - Hinzufügen von Dongles:

- Als Administrator, nach dem erfolgreichen Login befinden Sie sich auf der Übersichtsseite.
- Klicken Sie auf den Button "Dongle hinzufügen".
- Füllen Sie das Formular mit den erforderlichen Informationen über den Dongle aus.
- Klicken Sie auf "Speichern", um den Dongle direkt in die Datenbank hinzuzufügen.



## Dongle hinzufügen

Dongle-ID(Seriennummer)	<input type="text" value="Dongle-ID (Seriennum)"/>	Produktname	<input type="text" value="Produktname"/>
Händler	<input type="text" value="Händler"/>	Standort	<input type="text" value="Standort"/>
Gültig von	<input type="text" value="TT.MM.JJJJ"/> 	Gültig bis	<input type="text" value="TT.MM.JJJJ"/> 
FirmCode	<input type="text" value="FirmCode"/>	Projekt	<input type="text" value="Projekt"/>
<input type="button" value="Stornieren"/> <input type="button" value="Speichern"/>			

### Admin-Funktionalität - Hinzufügen von Lizenzen:

- Als Administrator öffnen Sie die Lizenzübersicht Seite.
- Klicken Sie auf den Button "Lizenz hinzufügen".
- Füllen Sie das Formular mit den erforderlichen Informationen über die Lizenz aus.
- Klicken Sie auf "Speichern", um die Lizenz direkt in die Datenbank hinzuzufügen.

## Lizenz hinzufügen

Mitarbeiter	<input type="text" value="Mitarbeiter"/>	Lizenzname	<input type="text" value="Lizenzname"/>
Einheiten	<input type="text" value="Einheiten"/>	ProductCode	<input type="text" value="ProductCode"/>
Gültig von	<input type="text" value="TT.MM.JJJJ"/> 	Gültig bis	<input type="text" value="TT.MM.JJJJ"/> 
Projekt	<input type="text" value="Projekt"/>	FirmCode	<input type="text" value="FirmCode"/>
Lizenzanzahl	<input type="text" value="Lizenzanzahl"/>	Dongle Seriennummer	<input type="text" value="Dongle Seriennumm"/>
<input type="button" value="Stornieren"/> <input type="button" value="Speichern"/>			

### Bearbeitung von Anfragen:

- Als Administrator öffnen Sie die Anfrageseite.
- Überprüfen Sie die eingegangenen Anfragen für Lizenzen oder Dongles.
- Klicken Sie auf "Annehmen" oder "Ablehnen" und geben Sie bei Bedarf den Ablehnungsgrund ein.

### Anfrage für neue Dongles

igsdatum	Schließungsdatum	Firmcode	Gültig von	Gültig bis	Mitarbeiter	Donglename	Dongleseriennummer	Projekt / Produkt	Standort	Händler	Grund der Ablehnung	Aktionen
-21	2020-05-21	FIRM123	2020-05-21	2020-05-21		h	54	h	h	h	fdgdfgh	
-21	2020-05-21	FIRM123	2020-05-21	2020-05-21		h	54	h	h	h		
-21	2020-05-21	FIRM123	2020-05-21	2020-05-21		h	54	h	h	h		<div>Akzeptieren</div> <div>Ablehnen</div>

## Passwort vergessen:

- Auf der Login-Seite klicken Sie auf "Passwort vergessen".
- Geben Sie Ihre E-Mail-Adresse ein.
- Klicken Sie auf "Passwort zurücksetzen".
- Überprüfen Sie Ihr E-Mail-Postfach auf eine Nachricht mit einem neuen Passwort.
- Verwenden Sie das erhaltene neue Passwort, um sich einzuloggen.

## Passwort ändern:

- Nach dem Login klicken Sie auf "Profil".
- Geben Sie Ihr aktuelles Passwort und das neue Passwort in die entsprechenden Felder ein.
- Klicken Sie auf "Passwort ändern".
- Überprüfen Sie Ihr E-Mail-Postfach auf eine Bestätigungsnachricht über die erfolgreiche Passwortänderung.

Altes Passwort

Neues Passwort

Neues Passwort bestätigen

Passwort ändern

## Konto erstellen:

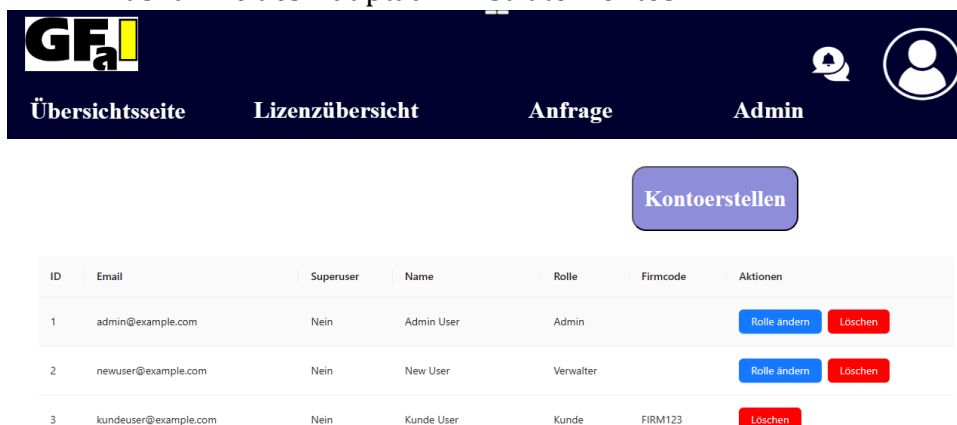
- Der Administrator hat die Möglichkeit, über die Admin-Seite ein Benutzerkonto zu erstellen.
- Auf der Admin-Seite klickt der Administrator auf "Konto erstellen".
- Der Administrator gibt die E-Mail-Adresse und den Namen des Benutzers ein und wählt die Rolle aus (Kunde, Administrator, Verwalter).
- Nach dem Speichern des Kontos wird automatisch eine E-Mail an den Benutzer gesendet, in der ihm mitgeteilt wird, dass ein Konto für ihn erstellt wurde. Die E-Mail enthält ein zufällig generiertes Passwort.
- Beim ersten Login wird der Benutzer dazu aufgefordert, das Passwort über ein Pop-up-Fenster zu ändern.



The screenshot shows the 'Admin' page of the GFaI system. The navigation bar includes 'Übersichtsseite', 'Lizenzübersicht', 'Anfrage', and 'Admin'. The 'Admin' page contains a form to create a new user account. The form has the following fields: 'Rolle wählen' (a dropdown menu), 'E-Mail', 'E-Mail bestätigen', and 'Name'. Below these fields is a button labeled 'Benutzer erstellen'.

## Kontoverwaltung:

- Auf der Admin-Seite werden alle Benutzerkonten angezeigt.
- Der Administrator kann die Rolle eines Benutzers ändern, z. B. von Verwalter zu Administrator oder umgekehrt. Dies ermöglicht es dem Administrator, die Zugriffsrechte für einzelne Benutzer anzupassen.
- Der Administrator hat die Möglichkeit, Benutzerkonten zu löschen, mit Ausnahme des Hauptadministratorkontos.



The screenshot shows the 'Admin' page with a table listing all user accounts. Above the table is a button labeled 'Konto erstellen'. The table has the following columns: ID, Email, Superuser, Name, Rolle, Firmcode, and Aktionen. The table contains three rows of data.

ID	Email	Superuser	Name	Rolle	Firmcode	Aktionen
1	admin@example.com	Nein	Admin User	Admin		<a href="#">Rolle ändern</a> <a href="#">Löschen</a>
2	newuser@example.com	Nein	New User	Verwalter		<a href="#">Rolle ändern</a> <a href="#">Löschen</a>
3	kundeuser@example.com	Nein	Kunde User	Kunde	FIRM123	<a href="#">Löschen</a>

## **Installationshandbuch:**

In diesem Installationshandbuch finden Sie eine Übersicht über die notwendigen Schritte, um das Projekt auf Ihrem Computer zum Laufen zu bringen. Beachten Sie, dass eine **detailliertere Anleitung und weitere Informationen in der README .md-Datei** des Projekts enthalten sind.

Die grundlegenden Schritte zur Installation des Projekts sind:

12. Stellen Sie sicher, dass Python auf Ihrem System installiert ist.
13. Installieren Sie Anaconda (falls noch nicht geschehen).
14. Erstellen Sie eine Anaconda-Umgebung und installieren Sie die benötigten Pakete.
15. Installieren Sie Node.js und npm.
16. Installieren Sie React und die erforderlichen Node-Module.
17. Starten Sie das Django-Projekt und das React-Frontend.

Bitte konsultieren Sie die README .md-Datei für weitere Details und Anweisungen zu jedem Schritt.

## **Teststrategie und Testfälle:**

Unsere Teststrategie umfasst sowohl automatisierte Tests für das Backend als auch manuelle Tests für das Frontend. Wir haben Unit-Tests für die meisten Funktionen des Backends implementiert, um sicherzustellen, dass einzelne Komponenten ordnungsgemäß funktionieren. Für die restlichen Backend-Funktionen und das Frontend haben wir manuelle Tests durchgeführt, um das Verhalten der Anwendung in verschiedenen Szenarien zu überprüfen.

Hier ist ein Überblick über unsere Teststrategie:

### **Unit-Tests:**

- Wir haben umfangreiche Unit-Tests für das Backend entwickelt, um einzelne Funktionen und Komponenten isoliert zu testen.
- Diese Tests werden automatisiert ausgeführt, um die korrekte Funktionalität sicherzustellen und potenzielle Fehler oder Fehler in den einzelnen Modulen zu identifizieren.
- Durch die Durchführung von Unit-Tests können wir sicherstellen, dass einzelne Funktionen unabhängig voneinander korrekt funktionieren und die erwarteten Ergebnisse liefern.

### **Manuelle Tests:**

- Wir haben manuelle Tests für das Frontend entwickelt, um das Verhalten der Anwendung aus Sicht des Benutzers zu überprüfen.
- Diese Tests werden manuell durchgeführt, um sicherzustellen, dass die Benutzeroberfläche, die Navigation und die Funktionen der Anwendung reibungslos funktionieren.
- Durch manuelle Tests können wir verschiedene Szenarien und Benutzereingaben simulieren, um sicherzustellen, dass die Anwendung den Anforderungen und Spezifikationen entspricht.



## API-Dokumentation

### Authentifizierung

#### Benutzer einloggen API:

**URL:** /login/

**Methode:** POST

**Parameter:**

- **email** (string): Die E-Mail-Adresse des Benutzers.
- **password** (string): Das Passwort des Benutzers.

**Antwortcodes:**

- 200: Erfolgreiche Anmeldung.
- 401: Ungültige E-Mail oder Passwort.

#### Adminzugriffstoken abrufen API:

**URL:** /admin-access-token/

**Methode:** POST

**Parameter:**

- **email** (string): Die E-Mail-Adresse des Administrators.
- **password** (string): Das Passwort des Administrators.

**Antwortcodes:**

- 200: Erfolgreiches Abrufen des Adminzugriffstokens.
- 403: Benutzer ist kein Administrator.
- 401: Ungültige E-Mail oder Passwort.

### **Zugriffstoken abrufen API:**

**URL:** /user-access-token/

**Methode:** POST

**Parameter:**

- **email** (string): Die E-Mail-Adresse des Benutzers.
- **password** (string): Das Passwort des Benutzers.

**Antwortcodes:**

- 200: Erfolgreiches Abrufen des Zugriffstokens.
- 401: Ungültige E-Mail oder Passwort.

### **Benutzerverwaltung**

#### **Benutzer erstellen API:**

**URL:** /users/

**Methode:** POST

**Parameter:**

- **email** (string): Die E-Mail-Adresse des Benutzers.
- **name** (string): Der Name des Benutzers.
- **role** (string): Die Rolle des Benutzers.
- **firm\_code** (string): Der Firmencode des Benutzers.

**Antwortcodes:**

- 201: Benutzer erfolgreich erstellt.
- 400: Fehlende Pflichtfelder.

#### **Passwort ändern API:**

**URL:** /passwortverwalten/

**Methode: POST**

**Parameter:**

- **email** (string): Die E-Mail-Adresse des Benutzers.
- **oldPassword** (string): Das alte Passwort des Benutzers.
- **newPassword** (string): Das neue Passwort des Benutzers.

**Antwortcodes:**

- 200: Passwort erfolgreich geändert.
- 400: Fehlende Pflichtfelder oder falsches altes Passwort.
- 404: Benutzer nicht gefunden.

### **Passwort zurücksetzen API:**

**URL: /Passwortzuruecksetzung/**

**Methode: POST**

**Parameter:**

- **email** (string): Die E-Mail-Adresse des Benutzers.

**Antwortcodes:**

- 200: Passwort erfolgreich zurückgesetzt.
- 400: Fehlende Pflichtfelder.
- 404: Benutzer nicht gefunden.

### **Lizenzseite API:**

- **URL:** /api/Lizenzseite/
- **Methode:** GET
- **Beschreibung:** Diese API liefert eine Liste von Lizenzen basierend auf der Rolle und dem FirmCode des authentifizierten Benutzers.
- **Authentifizierung:** Basic Authentication
- **Parameter:** Keine zusätzlichen Parameter erforderlich.
- **Antwort:** Die API gibt eine JSON-Antwort zurück, die eine Liste von Lizenzen enthält.

### Lizenzhinzufügen API:

- **Endpoint:** /api/Lizenzticket/create/
- **Methode:** POST
- **Beschreibung:** Diese API erstellt eine neue Lizenz basierend auf den bereitgestellten Daten.
- **Authentifizierung:** Basic Authentication
- **Parameter:**

Feldname	Typ	Erforderlich	Beschreibung
dongle_seriennummer	String	Ja	Dongle-Seriennummer
titel	String	Nein	Titel der Lizenz
beschreibung	String	Nein	Beschreibung der Lizenz
lizenzname	String	Ja	Name der Lizenz
erstellungsdatum	Date	Ja	Erstellungsdatum
schliessungsdatum	Date	Ja	Schließungsdatum
gueltig_von	Date	Ja	Gültig von
gueltig_bis	Date	Ja	Gültig bis
projekt	String	Ja	Projekt
einheiten	Integer	Ja	Einheiten
productcode	Integer	Ja	Productcode
firmcode	String	Ja	FirmCode
lizenzanzahl	Integer	Ja	Lizenzanzahl

- **Antwort:** Die API gibt eine JSON-Antwort zurück, die den Status der Anfrage enthält.

### LizenzAnfordern API:

- **URL:** /api/LizenzAnfordern/create/
- **Methode:** POST
- **Beschreibung:** Diese API ermöglicht es einem Kunden, eine Lizenz anzufordern.
- **Authentifizierung:** Basic Authentication
- **Parameter:**

Feldname	Typ	Erforderlich	Beschreibung
dongle_seriennummer	String	Ja	Dongle-Seriennummer
titel	String	Nein	Titel der Lizenz

beschreibung	String	Nein	Beschreibung der Lizenz
lizenzname	String	Ja	Name der Lizenz
erstellungsdatum	Date	Ja	Erstellungsdatum
schliessungsdatum	Date	Ja	Schließungsdatum
gueltig_von	Date	Ja	Gültig von
gueltig_bis	Date	Ja	Gültig bis
projekt	String	Ja	Projekt
einheiten	Integer	Ja	Einheiten
productcode	Integer	Ja	Productcode
firmcode	String	Ja	FirmCode
lizenzanzahl	Integer	Ja	Lizenzanzahl

- **Antwort:** Die API gibt eine JSON-Antwort zurück, die den Status der Anfrage enthält.

### **Dongle hinzufügen API:**

- **URL:** /api/dongle/create/
- **Methode:** POST
- **Beschreibung:** Erstellt einen neuen Dongle.

### **Parameter**

Name	Typ	Beschreibung
serien_nr	string	Die Seriennummer des Dongles.
name	string	Der Name des Dongles.
gueltig_von	string	Das Gültigkeitsdatum von.
gueltig_bis	string	Das Gültigkeitsdatum bis.
projekt_produk	string	Das Projekt oder Produkt des Dongles.

<b>standort</b>	string	Der Standort des Dongles.
<b>haendler</b>	string	Der Händler des Dongles.
<b>datum_letzte_aenderung</b>	string	Das Datum der letzten Änderung.
<b>datum_erstausgabe</b>	string	Das Datum der Erstausgabe.
<b>firmcode</b>	string	Der Firmencode des Dongles.

#### Erfolgreiche Antwort

- Statuscode: 201 Created
- Inhalt: { "success": "Dongle erfolgreich erstellt." }

#### Fehlerantwort

- Statuscode: 400 Bad Request
- Inhalt: { "error": "Fehlerhafte Anfrage." }

#### Dongle-Anfordern API:

- **URL:** /api/Dongleticket/create/
- **Methode:** POST
- **Beschreibung:** Erstellt ein neues Ticket für die Anforderung eines Dongles.
- **Parameter**

Name	Typ	Beschreibung
<b>dongle_seriennummer</b>	string	Die Seriennummer des angeforderten Dongles.
<b>dongle_name</b>	string	Der Name des angeforderten Dongles.
<b>gueltig_von</b>	string	Das Gültigkeitsdatum von.
<b>gueltig_bis</b>	string	Das Gültigkeitsdatum bis.
<b>projekt</b>	string	Das Projekt des angeforderten Dongles.
<b>standort</b>	string	Der Standort des angeforderten Dongles.
<b>haendler</b>	string	Der Händler des angeforderten Dongles.
<b>firmcode</b>	string	Der Firmencode des angeforderten Dongles.
<b>titel</b>	string	Der Titel des Tickets.

<b>beschreibung</b>	string	Die Beschreibung des Tickets.
<b>erstellungsdatum</b>	string	Das Erstellungsdatum des Tickets.
<b>schliessungsdatum</b>	string	Das Schließungsdatum des Tickets.

- **Erfolgreiche Antwort**
- Statuscode: 201 Created
- Inhalt: { "success": "Ihre Anfrage wurde gesendet!" }
- **Fehlerantwort**
- Statuscode: 401 Unauthorized
- Inhalt: { "error": "Fehler beim Erstellen der Lizenz: {Fehlermeldung}" }

### **homepage API:**

- **URL:** /homepage/
- **Methode:** GET
- **Beschreibung:** Diese API-Anfrage gibt eine Liste von Dongles basierend auf der Rolle und dem Firmencode des authentifizierten Benutzers zurück.
- **Authentifizierung:** Basic Authentication
- **Parameter:** Keine zusätzlichen Parameter erforderlich.
- **Antwort:** Die API gibt eine JSON-Antwort zurück, die eine Liste von Dongles enthält.

## **Frontend-Dokumentation**

### **CustomuserTable**

CustomuserTable ist eine React-Komponente, die eine Tabelle mit benutzerdefinierten Benutzerdaten anzeigt. Die Tabelle ermöglicht das Bearbeiten der Benutzerrolle und das Löschen von Benutzern.

#### **Zustandsvariablen**

- data: Eine Zustandsvariable, die die Daten für die Tabelle speichert.
- editingId: Eine Zustandsvariable, die die ID des gerade bearbeiteten Benutzers speichert.
- form: Eine Instanz von Form, die zur Verwaltung der Formulardaten verwendet wird.
- email und setRole: Werte aus dem AuthContext, die die E-Mail und die Rolle des aktuellen Benutzers speichern.

#### **useEffect-Hook**

Der useEffect-Hook wird verwendet, um Daten von der API abzurufen und in der data-Zustandsvariablen zu speichern. Der Hook wird bei der Initialisierung der Komponente ausgeführt.

#### **EditableCell-Komponente**

EditableCell ist eine Komponente, die eine editierbare Tabellenzelle darstellt. Sie ermöglicht das Bearbeiten der Benutzerrolle in der Tabelle und zeigt einen Bestätigungsdialog an, wenn der Benutzer die Rolle ändern möchte.

#### **Funktionen**

- handleConfirm: Eine Funktion, die den Bestätigungsdialog für die Rollenänderung anzeigt.
- handleDelete: Eine Funktion zum Löschen eines Benutzers aus der Tabelle und der Datenbank.
- isEditing: Eine Funktion, die überprüft, ob eine Zeile in der Tabelle gerade bearbeitet wird.
- edit: Eine Funktion zum Aktivieren des Bearbeitungsmodus für eine Zeile.
- cancel: Eine Funktion zum Abbrechen des Bearbeitungsmodus.
- save: Eine Funktion zum Speichern der geänderten Benutzerdaten.



## Spalten

Die columns-Variable enthält die Spaltenkonfigurationen für die Tabelle. Jede Spalte zeigt bestimmte Informationen aus den Benutzerdaten an. Die "Rolle"-Spalte ist editierbar und zeigt ein Dropdown-Menü zum Ändern der Benutzerrolle an. Die "Aktionen"-Spalte enthält Schaltflächen zum Bearbeiten und Löschen von Benutzern.

## DongleTable-Komponente

Die DongleTable-Komponente ist eine React-Komponente, die eine Tabelle mit Informationen zu Dongles anzeigt. Die Tabelle verfügt über eine Filterfunktion, um Dongles anhand des Namens zu suchen.

## Zustandsvariablen

- `dataSource`: Eine Zustandsvariable, die die Daten für die Tabelle speichert.
- `email` und `password`: Werte aus dem AuthContext, die die E-Mail und das Passwort des aktuellen Benutzers speichern.
- `useEffect`-Hooks:
- Der erste `useEffect`-Hook speichert die E-Mail des Benutzers im Local Storage, wenn sie sich ändert.
- Der zweite `useEffect`-Hook speichert das Passwort des Benutzers im Local Storage, wenn es sich ändert.
- Der dritte `useEffect`-Hook wird verwendet, um Daten von der API abzurufen und in der `dataSource`-Zustandsvariablen zu speichern. Der Hook wird bei der Initialisierung der Komponente ausgeführt und bei Änderungen von `email` und `password`.

## fetchData-Funktion

Die `fetchData`-Funktion ist eine asynchrone Funktion, die die Dongle-Daten von der API abrufen und in der `dataSource`-Zustandsvariablen speichert. Die Funktion verwendet `Axios`, um einen GET-Antrag an die API zu senden und die erwarteten Daten im JSON-Format zu erhalten. Die Funktion wird im dritten `useEffect`-Hook aufgerufen.

## Spaltenkonfiguration

Die columns-Variable enthält die Spaltenkonfigurationen für die Tabelle. Jede Spalte zeigt bestimmte Informationen aus den Dongle-Daten an. Die "Name"-Spalte verfügt über eine Filterfunktion, die ein Eingabefeld zum Suchen von Dongles anhand des Namens anzeigt.

## Filter-Dropdown-Konfiguration

Die "Name"-Spalte verwendet eine benutzerdefinierte Dropdown-Komponente für die Filterfunktion. Die Dropdown-Komponente besteht aus einem Input-Element, das verwendet wird, um den Namen des Dongles einzugeben, nach dem gesucht werden soll. Die `onPressEnter`- und `onBlur`-Ereignishandler lösen die Filterfunktion aus.

## Filterfunktion

Die Filterfunktion für die "Name"-Spalte überprüft, ob der Name des Dongles den eingegebenen Suchbegriff enthält. Die Funktion wird für jeden Datensatz in der Tabelle aufgerufen und gibt true zurück, wenn der Name des Dongles den Suchbegriff enthält. Andernfalls gibt sie false zurück. Die Filterfunktion ist case-insensitiv.

## Tabellenkomponente

Die Tabellenkomponente verwendet die Table-Komponente von Ant Design. Die columns- und dataSource-Eigenschaften werden an die Table-Komponente übergeben, um die Spalten und Daten für die Tabelle zu konfigurieren. Die pagination-Eigenschaft wird verwendet, um die Anzahl der angezeigten Zeilen pro Seite auf 7 zu begrenzen.

## EinloggenSeite

Die Einloggen-Komponente ist eine React-Komponente, die ein Login-Formular darstellt. Benutzer können ihre E-Mail und Passwort eingeben, um sich anzumelden. Die Komponente verwendet Zustandsvariablen, Effekte und Funktionen, um den Anmeldeprozess zu verwalten und Fehlermeldungen anzuzeigen.

## Hauptmerkmale

- Benutzer können ihre E-Mail und Passwort eingeben, um sich anzumelden.
- Die Passwort-Sichtbarkeit kann umgeschaltet werden.
- Es gibt einen Link zum Navigieren zur Passwortzurücksetzungsseite.
- Fehlermeldungen werden angezeigt, wenn die Anmeldung fehlschlägt.

## Zustandsvariablen

- inputEmail: Speichert den aktuellen Wert des E-Mail-Eingabefelds.
- inputPassword: Speichert den aktuellen Wert des Passwort-Eingabefelds.
- errorMessage: Speichert die aktuelle Fehlermeldung, falls vorhanden.
- passwordInputKey: Wird verwendet, um das Passwordeingabefeld bei Bedarf zurückzusetzen.
- showPassword: Speichert den aktuellen Sichtbarkeitsstatus des Passworts.

## useEffect-Hooks

Der erste useEffect-Hook wird verwendet, um den Benutzer zur Übersichtsseite weiterzuleiten, wenn er bereits angemeldet ist. Dies geschieht basierend auf der Rolle des Benutzers.

## Funktionen

- togglePasswordVisibility: Ändert den Sichtbarkeitsstatus des Passworts im Passwordeingabefeld.
- handleOkClick: Setzt das Passwort und die Fehlermeldung zurück.
- handlePasswordResetClick: Navigiert zur Passwortzurücksetzungsseite.

- `handleLoginSubmit`: Verarbeitet das Einreichen des Login-Formulars, indem eine Anfrage an die API gesendet wird, um den Benutzer anzumelden. Bei erfolgreicher Anmeldung werden die Benutzerrolle, E-Mail und Passwort gesetzt und der Benutzer wird zur Übersichtsseite weitergeleitet. Bei fehlgeschlagener Anmeldung wird eine Fehlermeldung angezeigt.

## AdminSeite

Dieser Code definiert die Adminseite-Komponente, die eine Benutzeroberfläche für Administratoren darstellt. Die Komponente enthält eine Navigationsleiste, einen "Konto erstellen"-Button und eine benutzerdefinierte Tabelle (`CustomuserTable`). Die Adminseite-Komponente verwendet den `useAdminAccess`-Hook, um zu überprüfen, ob der aktuelle Benutzer ein Administrator ist. Wenn der Benutzer kein Administrator ist, wird er zur Übersichtsseite weitergeleitet.

### Hauptmerkmale

Zeigt die Administrationsseite mit einer Navigationsleiste, einem "Konto erstellen"-Button und einer benutzerdefinierten Tabelle an (`CustomuserTable`).

Verwendet den `useAdminAccess`-Hook, um zu überprüfen, ob der aktuelle Benutzer ein Administrator ist.

Leitet Benutzer, die keine Administratoren sind, zur Übersichtsseite weiter.

### `useEffect`-Hooks

Der `useEffect`-Hook überprüft, ob der Benutzer ein Administrator ist und leitet ihn gegebenenfalls zur Übersichtsseite weiter.

## Kontoerstellen

Kontoerstellen ist eine React-Komponente, die ein Formular zum Erstellen eines neuen Benutzerkontos anzeigt. Diese Komponente wird typischerweise vom Administrator verwendet, um neue Benutzer für die Anwendung hinzuzufügen. Die Komponente verwendet den `AuthContext` und die `useAdminAccess`-Hook, um sicherzustellen, dass nur Administratoren Zugriff auf diese Seite haben.

### Zustände und Effekte

- `role1`: Der ausgewählte Wert für die Benutzerrolle im Formular.
- `email1`: Der eingegebene Wert für die E-Mail-Adresse des Benutzers.
- `confirmEmail`: Der eingegebene Wert für die E-Mail-Bestätigung des Benutzers.
- `name1`: Der eingegebene Wert für den Benutzernamen.
- `firmCode`: Der eingegebene Wert für den Firmencode des Benutzers (nur für Kunden).
- `showPopup`: Boolean Zustand, der steuert, ob das Erfolgspopup angezeigt wird oder nicht.

- Die useEffect-Hooks werden verwendet, um E-Mail und Passwort des eingeloggten Benutzers im localStorage zu speichern und um eine Weiterleitung zur Übersichtseite durchzuführen, wenn der Benutzer kein Admin ist.

## Funktionen

- isInputOnlySpaces(input): Überprüft, ob die Eingabe nur Leerzeichen enthält.
- isValidEmail(emailAddress): Überprüft, ob die E-Mail-Adresse gültig ist.
- getAdminAccessToken(): Ruft das Admin-Zugriffstoken ab, das zum Erstellen von neuen Benutzern benötigt wird.
- handleSubmit(e): Formular-Handler, der das Erstellen eines neuen Benutzers auslöst.
- Rendern der Komponente
- Die Komponente besteht aus verschiedenen Eingabefeldern, die den Benutzer dazu auffordern, Informationen wie Rolle, E-Mail, Name und Firmencode (falls erforderlich) einzugeben. Nach der Validierung der Eingaben und erfolgreicher Erstellung des neuen Benutzers wird

## KundeTicketTable

KundeTicketTable ist eine React-Komponente, die eine Tabelle mit Ticketdaten für Kunden anzeigt. Die Tabelle verwendet die Ant Design-Bibliothek und zeigt die Daten in einer paginierten Tabelle an. Die Komponente nimmt zwei optionale Props filter und customColumns entgegen, um die angezeigten Daten zu filtern und benutzerdefinierte Spalten hinzuzufügen.

## Zustände und Effekte

- dataSource: Der Zustand, der die Datenquelle für die Tabelle speichert.
- modalVisible: Boolean Zustand, der steuert, ob das Modal angezeigt wird oder nicht.
- selectedTicketId: Der Zustand, der die ausgewählte Ticket-ID speichert.
- Die useEffect-Hooks werden verwendet, um E-Mail und Passwort des eingeloggten Benutzers im localStorage zu speichern und um die Ticketdaten asynchron abzurufen.

## Funktionen

fetchData(): Eine asynchrone Funktion, die Ticketdaten von der API abrufen und den dataSource-Zustand aktualisiert.

## Rendern der Komponente

Die Komponente rendert eine Ant Design Table-Komponente mit den angegebenen Spalten und Daten. Die Tabelle ist paginiert, sodass jeweils nur eine begrenzte Anzahl von Zeilen angezeigt wird.

## **Lizenzübersicht**

Lizenzübersicht ist eine React-Komponente, die eine Übersichtsseite für Lizenzen anzeigt. Die Seite enthält eine Tabelle mit Lizenzinformationen, die von der LizenzTable-Komponente gerendert wird, und einen Button zum Hinzufügen oder Anfordern von Lizenzen, abhängig von der Rolle des eingeloggtten Benutzers. Die Komponente verwendet den AuthContext, um die Benutzerrolle abzurufen, und den useNavigate-Hook, um die Navigation zu ermöglichen.

### **Zustände und Effekte**

Es gibt keine lokalen Zustände oder Effekte in der Komponente. Der AuthContext wird jedoch verwendet, um die Benutzerrolle abzurufen.

### **Funktionen**

Es gibt keine expliziten Funktionen in der Komponente, der Button zum Hinzufügen oder Anfordern von Lizenzen verwendet jedoch den onClick-Handler, um die Navigation zu ermöglichen.

### **Rendern der Komponente**

Die Komponente rendert eine Navbar, einen Hintergrund und die Lizenzübersicht, die aus einer Tabelle und einem Button zum Hinzufügen oder Anfordern von Lizenzen besteht. Der Button wird basierend auf der Rolle des Benutzers angezeigt.

## **LizenzTable-Komponente**

LizenzTable ist eine React-Komponente, die eine Tabelle mit Lizenzinformationen anzeigt. Die Tabelle verwendet die Ant Design-Bibliothek und zeigt die Daten in einer paginierten Tabelle an. Die Komponente verwendet den AuthContext, um die Anmeldeinformationen des eingeloggtten Benutzers abzurufen und die Daten von der API abzurufen.

### **Zustände und Effekte**

dataSource: Der Zustand, der die Datenquelle für die Tabelle speichert.

Der useEffect-Hook wird verwendet, um die Lizenzdaten asynchron abzurufen und den dataSource-Zustand zu aktualisieren.

### **Funktionen**

fetchData(): Eine asynchrone Funktion, die Lizenzdaten von der API abrufen und den dataSource-Zustand aktualisiert.

### **Spaltenkonfiguration**

Die Spaltenkonfiguration definiert die Spalten, die in der Tabelle angezeigt werden sollen. Jede Spalte hat einen Titel, einen dataIndex und optional einen key.

### **Rendern der Komponente**

Die Komponente rendert eine Ant Design Table-Komponente mit den angegebenen Spalten und Daten. Die Tabelle ist paginiert, sodass jeweils nur eine begrenzte Anzahl von Zeilen angezeigt wird.

### **Passwortzurücksetzung**

Passwortzurücksetzung ist eine React-Komponente, die ein Formular zum Zurücksetzen des Passworts bereitstellt. Die Benutzer müssen ihre E-Mail-Adresse eingeben, sie bestätigen und ein Captcha lösen, um das Formular abzusenden.

### **Zustände**

- inputEmail: Der Zustand für die E-Mail-Eingabe
- confirmEmail: Der Zustand für die Bestätigungs-E-Mail-Eingabe
- captchaInput: Der Zustand für die Captcha-Eingabe
- captcha: Der Zustand für das zufällige Captcha
- showPopup: Der Zustand für das Anzeigen des Erfolgspopups

### **Funktionen**

- generateRandomCaptcha(): Generiert ein zufälliges Captcha und gibt es zurück.
- refreshCaptcha(): Aktualisiert das Captcha, indem ein neues zufälliges Captcha generiert und in den Zustand gesetzt wird.
- showSuccessPopup(): Zeigt das Erfolgspopup an, indem der Zustand showPopup auf true gesetzt wird.
- SuccessPopup(): Eine Komponente, die ein Erfolgspopup anzeigt, wenn das Passwort erfolgreich zurückgesetzt wurde.
- handlePasswordResetSubmit(): Eine asynchrone Funktion, die das Passwortzurücksetzungsformular verarbeitet und die Passwortzurücksetzungsanforderung an die API sendet.

### **Rendern der Komponente**

Die Komponente rendert ein Formular mit Eingabefeldern für E-Mail, Bestätigungs-E-Mail und Captcha sowie einen Button zum Absenden des Formulars. Wenn das Formular erfolgreich abgesendet wurde, wird ein Erfolgspopup angezeigt.

### **Profilseite**

Das ist eine React-Komponente für eine Profilseite, auf der Benutzer ihr Passwort ändern können. Die Hauptfunktionalität der Komponente ist in der Funktion handleSubmit

enthalten, die eine Anfrage an die API sendet, um das Passwort zu ändern. Die Komponente verwendet auch verschiedene Zustände und Nebeneffekte, um die Benutzerinteraktion und das Rendering zu handhaben.

### Hauptzustände

- `oldPassword`: Speichert das aktuelle Passwort des Benutzers.
- `newPassword`: Speichert das neue Passwort, das der Benutzer eingegeben hat.
- `confirmNewPassword`: Speichert die Bestätigung des neuen Passworts, das der Benutzer eingegeben hat.
- `isFormValid`: Gibt an, ob das Formular gültig ist und abgesendet werden kann.
- `showPopup`: Steuert die Anzeige des Erfolgspopups.

### Hauptfunktionen

- `showSuccessPopup()`: Zeigt das Erfolgspopup an, indem der Zustand `showPopup` auf `true` gesetzt wird.
- `SuccessPopup()`: Eine Komponente, die ein Erfolgspopup anzeigt, wenn das Passwort erfolgreich geändert wurde.
- `checkFormValidity()`: Überprüft die Gültigkeit des Formulars und setzt den Zustand `isFormValid` entsprechend.
- `getUserAccessToken()`: Ruft das Benutzerzugriffstoken von der API ab.
- `handleSubmit()`: Eine asynchrone Funktion, die das Passwortänderungsformular verarbeitet und die Passwortänderungsanforderung an die API sendet.

## TicketTable

Dies ist eine React-Komponente namens `TicketTable`, die eine Tabelle zur Anzeige und Verwaltung von Tickets in einer Anwendung darstellt. Die Tabelle zeigt Ticketdaten an und ermöglicht es Benutzern, Tickets entweder anzunehmen oder abzulehnen. Die Komponente verwendet Hooks, um den Zustand und Nebeneffekte zu verwalten, und nutzt die `antd`-Bibliothek für die Gestaltung von UI-Komponenten.

### Hauptzustände

- `dataSource`: Speichert die Ticketdaten, die aus der API abgerufen wurden.
- `modalVisible`: Steuert die Anzeige des Modal-Dialogs zur Eingabe von Ablehnungsgründen.
- `ablehnungText`: Speichert den eingegebenen Ablehnungsgrund.
- `selectedTicketId`: Speichert die ID des ausgewählten Tickets, das abgelehnt werden soll.

### Hauptfunktionen

- `handleAccept(ticketId)`: Eine asynchrone Funktion, die aufgerufen wird, wenn der Benutzer auf "Akzeptieren" klickt. Sie sendet eine POST-Anfrage an die API, um das Ticket anzunehmen, und aktualisiert die lokale Datenquelle.
- `handleAblehnenClick(ticketId)`: Diese Funktion wird aufgerufen, wenn der Benutzer auf "Ablehnen" klickt. Sie setzt die ausgewählte Ticket-ID und zeigt das Modal zur Eingabe von Ablehnungsgründen an.
- `handleModalSubmit()`: Eine asynchrone Funktion, die aufgerufen wird, wenn der Benutzer den Modal-Dialog zur Ablehnung bestätigt. Sie sendet eine PUT-Anfrage an die API, um das Ticket abzulehnen, und aktualisiert die lokale Datenquelle.

## Übersichtseite

Dies ist eine React-Komponente namens Übersichtseite (engl. Overview Page), die den Hauptinhalt der Übersichtsseite in einer Anwendung darstellt. Die Komponente enthält eine Tabelle zum Anzeigen von Dongle-Daten (`DongleTable`) und einen Button zum Anfordern oder Hinzufügen von Dongles, je nach Rolle des Benutzers. Außerdem gibt es ein Popup zum Ändern des Passworts, das beim ersten Login des Benutzers angezeigt wird.

## Hauptzustände

- `showPasswordChangePopup`: Steuert die Anzeige des Popups zum Ändern des Passworts.
- `newPassword`: Speichert das vom Benutzer eingegebene neue Passwort.
- `confirmPassword`: Speichert das vom Benutzer eingegebene Bestätigungspasswort.
- `errorMessage`: Speichert eine Fehlermeldung bei Problemen beim Ändern des Passworts.
- `showPopup`: Steuert die Anzeige des Erfolgspopups nach erfolgreicher Passwortänderung.

## Hauptfunktionen

- `showSuccessPopup()`: Zeigt das Erfolgspopup an, wenn das Passwort erfolgreich geändert wurde.
- `SuccessPopup()`: Eine Funktion, die das Erfolgspopup rendert.
- `handlePasswordChange(newPassword, confirmPassword)`: Eine asynchrone Funktion, die aufgerufen wird, wenn der Benutzer auf "Passwort ändern" klickt. Sie sendet eine POST-Anfrage an die API, um das Passwort zu ändern, und aktualisiert den Wert von `last_login` im `SessionStorage`.

## useAdminAccess

Dies ist ein benutzerdefinierter React-Hook namens `useAdminAccess`, der verwendet wird, um den Adminstatus und den Ladestatus für einen Benutzer zu ermitteln. Er sendet eine POST-Anfrage an die API-Adresse "<http://127.0.0.1:8000/admin-access-token/>" mit den E-Mail- und Passwortanmeldedaten des Benutzers. Wenn die Antwort erfolgreich ist und der Benutzer ein Administrator ist, wird der Adminstatus auf `true` gesetzt.



## Hauptzustände

- isAdmin: Speichert, ob der Benutzer ein Administrator ist oder nicht.
- isLoading: Speichert den Ladestatus der Anfrage.

## Glossar

**Dongle:** Ein Hardware-Dongle mit einer eindeutigen Seriennummer, der zur Speicherung von Lizenzen verwendet wird. [wibu]

**Lizenz:** Ein Datensatz, der eine Lizenznummer oder Produktnummer sowie den Namen des Produkts enthält. Die Lizenz wird verwendet, um den Zugriff auf bestimmte Funktionen oder Module eines Produkts zu ermöglichen.

**Rolle:** Es gibt drei Rollen im System: Benutzer, Dongle-Verwalter und Administrator. Die Benutzer haben die geringsten Rechte, während die Dongle-Verwalter mehr Rechte haben, um Dongle und Lizenzen zu verwalten. Die Administratoren haben die höchsten Rechte und können alle Aspekte des Systems verwalten.

**Ablaufdatum:** Das Ablaufdatum gibt das Datum an, bis zu dem die Lizenz auf dem Dongle gültig ist. Falls kein Datum angegeben wurde, gilt die Lizenz unbefristet.

**Rechte:** Die Aktionsmöglichkeiten innerhalb einer Rolle werden durch Rechte definiert. Beispiele für solche Rechte sind unter anderem das Ändern der Datenbank, welches ausschließlich der Rolle des Administrators vorbehalten ist. Benutzer hingegen haben das Recht zur Erstellung von Anfragen zur Dongle-Änderung oder -Erstellung sowie zur Einsicht aller vorhandenen Dongle. Der Dongle-Verwalter hingegen besitzt das Recht, Dongle-Einträge zu ändern oder neue hinzuzufügen.

## **Lizenz**

Copyright (c) 2023 HTW Berlin Softwareentwicklung Gruppe 1 (Rama Abazied, Yassin Sahnoun, Houssein Hfasa)

### MIT-Lizenz

Hiermit wird jeder Person, die eine Kopie dieser Software und der zugehörigen Dokumentationsdateien (die "Software") erhält, die Erlaubnis erteilt, kostenlos und ohne Einschränkungen mit der Software zu handeln. Dies umfasst die Rechte zur Nutzung, zum Kopieren, Ändern, Zusammenführen, Veröffentlichen, Verteilen, Unterlizenzieren und/oder Verkaufen von Kopien der Software.

Die folgenden Bedingungen müssen erfüllt sein, um diese Erlaubnis zu erhalten:

Der oben genannte Urheberrechtshinweis und dieser Genehmigungshinweis müssen in allen Kopien oder wesentlichen Teilen der Software enthalten sein.

DIE SOFTWARE WIRD "WIE SIE IST" UND OHNE JEDGLICHE AUSDRÜCKLICHE ODER STILLSCHWEIGENDE GEWÄHRLEISTUNG BEREITGESTELLT. DIES BEINHÄLTET, ABER IST NICHT BESCHRÄNKT AUF, GEWÄHRLEISTUNGEN DER MARKTGÄNGIGKEIT, EIGNUNG FÜR EINEN BESTIMMTEN ZWECK UND NICHTVERLETZUNG VON RECHTEN DRITTER. IN KEINEM FALL SIND DIE URHEBER ODER URHEBERRECHTSINHABER HAFTBAR FÜR ANSPRÜCHE, SCHÄDEN ODER ANDERE VERBINDLICHKEITEN, OB IN EINER VERTRAGS- ODER HAFTUNGSKLAGE, EINER UNERLAUBTEN HANDLUNG ODER ANDERWEITIG, DIE SICH AUS, AUS ODER IN VERBINDUNG MIT DER SOFTWARE ODER DER NUTZUNG ODER ANDEREN HANDLUNGEN MIT DER SOFTWARE ERGEBEN.

Diese Software ist Open Source und wird unter der MIT-Lizenz vergeben.