

# **Technische Dokumentation**

Autor: Pascal Förber, Luis Hutterli

Datum: 24.01.2025

Version: 2.0

# 1 Inhaltsverzeichnis

1	Inhaltsverzeichnis .....	2
2	Analyse .....	4
2.1	Projektbeschreibung .....	4
2.2	Anforderungen .....	4
2.3	Realisierungstechnologie .....	5
2.4	Use-Cases .....	5
2.5	ERM .....	6
3	Design .....	6
3.1	Layout .....	6
3.1.1	Login/Signup .....	6
3.1.2	Main .....	7
3.1.3	Admin – Alle Benutzer .....	7
3.1.4	Admin – Ausleihen eines Benutzers .....	8
3.2	Datenbankschema .....	8
3.3	API-Endpunkte .....	9
3.4	Testliste .....	9
4	Realisation .....	12
4.1	Backend .....	12
4.2	Frontend .....	12
4.2.1	Routing .....	12
5	Test .....	13
5.1	Known Bugs .....	14
5.1.1	iOS Mobile Audio Playback .....	14
6	Deployment .....	14
6.1	Benutzeranleitung .....	14
6.1.1	Backend .....	14
6.1.2	Frontend .....	15
6.2	Live Deployment .....	15
7	Benutzeranleitung .....	15
7.1	Landing .....	15
7.2	Account .....	16
7.2.1	Erstellen .....	16
7.2.2	Anmelden .....	16

7.2.3	E-Mail-Verifizieren .....	17
7.2.4	Einstellungen .....	18
7.3	Hauptseite – Desktop .....	18
7.3.1	Song Ausleihen .....	18
7.3.2	Song Zurückgeben .....	19
7.3.3	Song Abspielen / Steuern.....	19
7.3.4	Song Steuern über OS-Integration.....	20
7.4	Hauptseite – Mobile .....	21
7.4.1	Song Ausleihen .....	21
7.4.2	Song Zurückgeben .....	21
7.4.3	Song Abspielen / Steuern.....	22
7.4.4	Song Steuern über OS-Integration.....	22
7.5	Admin.....	23
7.5.1	Startseite – Liste aller Benutzer.....	23
7.5.2	Benutzer – Liste der Ausleihen .....	23
7.5.3	Ausleihe – Zurückgeben.....	24
7.5.4	Ausleihe – Dauer ändern.....	24

## 2 Analyse

### 2.1 Projektbeschreibung

Es soll eine Web-App entwickelt werden, mit welcher man eine elektronische JukeBox mit NFT-Songs (Non-Fungible Tokens) benutzen und verwalten kann.

### 2.2 Anforderungen

Ein jukeStack Benutzer (z.B. Steven Hatch) soll sich mit seiner E-Mail Adresse (Steven.Hatch@gmx.net) und einem Passwort (z.B. djfhU78&wR) in der jukeStack-App registrieren können.

Sobald er sich registriert hat, kann er sich mittels E-Mail und Passwort in der jukeStack-App anmelden. Gelingt ihm dies, so sieht er einerseits eine Liste mit NFT-Songs (z.B. Ian Kelly, Speak your mind, 4:08, 2008), die er ausleihen kann und eine zweite Liste mit bereits ausgeliehenen Songs, die aber im Moment noch leer ist.

Nun kann er aus der ersten Liste einen Titel anwählen (z.B. Speak your mind) und ihn ausleihen. Sind die unten angeführten Bedingungen erfüllt, so wird die Ausleihe gewährt und der Song auf die Liste der ausgeliehenen Songs gesetzt. Die Liste der ausgeliehenen Songs enthält nun diesen Song, welcher um das Ausleihdatum ergänzt wurde (z.B. Ian Kelly, Speak your mind, 4:08, 2008, 11.12.2024 14:03) und kann nun abgespielt werden.

Die Bedingungen für eine Ausleihe lauten wie folgt:

- Ein Benutzer darf maximal nur 5 Songs gleichzeitig ausleihen.
- Ein Song kann, da jeder Song in dieser App ein NFT ist, gleichzeitig maximal nur einmal ausgeliehen sein.
- Ein Song kann für maximal n Tage ausgeliehen werden.

Sind die ersten zwei Bedingungen nicht erfüllt, so soll eine entsprechende Warnung ausgegeben werden.

Ist ein Song ausgeliehen, kann er so oft angehört werden, wie der Benutzer es möchte.

Ein ausgeliehener Song, kann, wenn er nicht mehr gewünscht wird, wieder zurückgegeben werden.

Zudem gibt es noch einen Administrator:in (z.B. Liza Suller, Liza.Suller@tokentec.ch, kljht564xc). Logt sie sich ein, sieht sie statt der Songs, eine Liste mit allen Benutzern. Wählt sie einen Benutzer an, sieht sie wiederum dessen Ausleihen und kann diese auch rückgängig machen. Zudem kann der Admin die maximale Ausleihdauer n (siehe Bedingung 3) festlegen (default = 1).

Die Passwörter müssen unbedingt gehasht und gesalzen (kein MD5!) in der DB abgelegt werden.

## 2.3 Realisierungstechnologie

Die Web-App wird folgende mit folgenden Technologien realisiert:

Front-End: React (vite, TypeScript), (UI und weitere Util Library) *laufend auf Google Firebase*

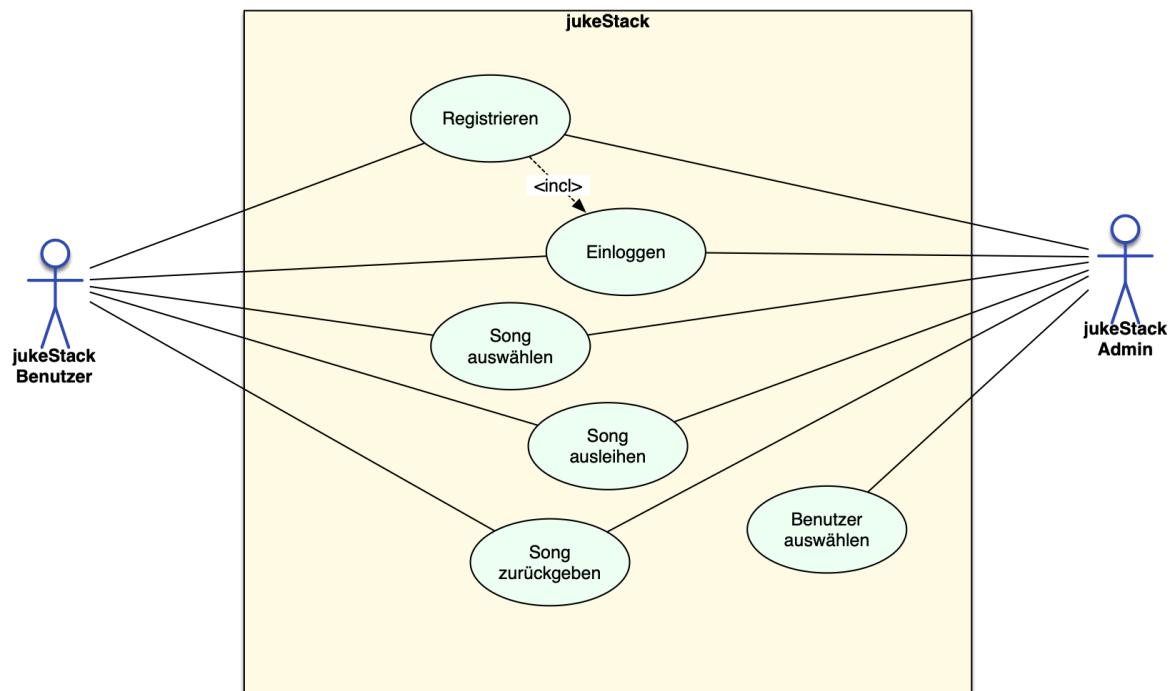
Back-End: MySQL, Java Vert.X Server *laufend auf Google Cloud Run*

Hinzu kommen diese Anbieter:

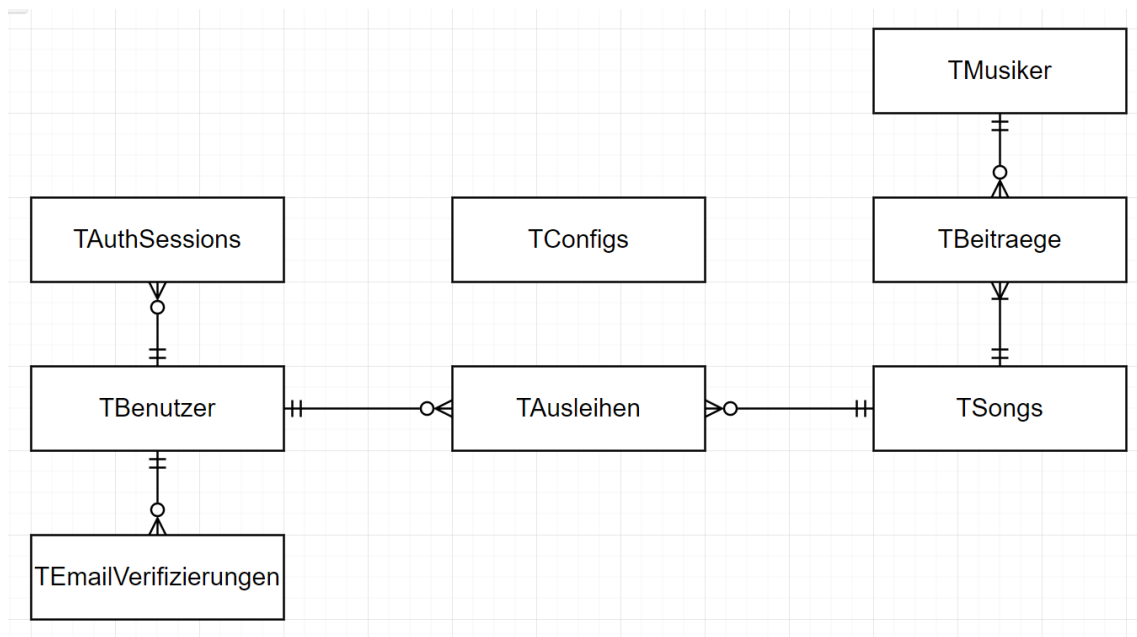
- Google Firebase (Frontend Deployment)
- Google Cloud Run (Backend Deployment)
- Cloudflare DNS (DNS, Proxy & Cache für die Webseite)
- Cloudflare R2 (Object Storage für Bilder und Song Dateien)
- Zoho Mail (Für automatische Mails und Test Accounts)

## 2.4 Use-Cases

Aus dem folgenden Diagramm sind die gewünschten Anwendungsfälle erkennbar.



## 2.5 ERM



## 3 Design

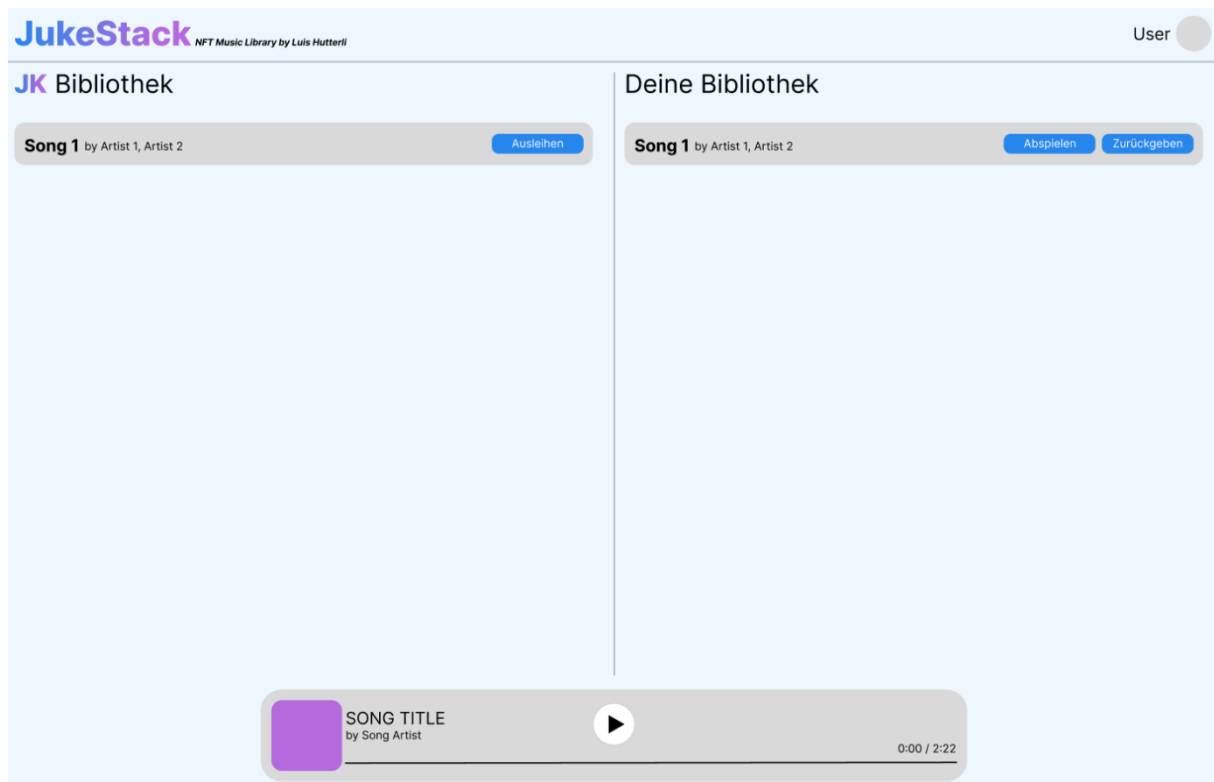
### 3.1 Layout

Die Design-Dateien finden sich im Ordner /diagrams/figma-wireframes/

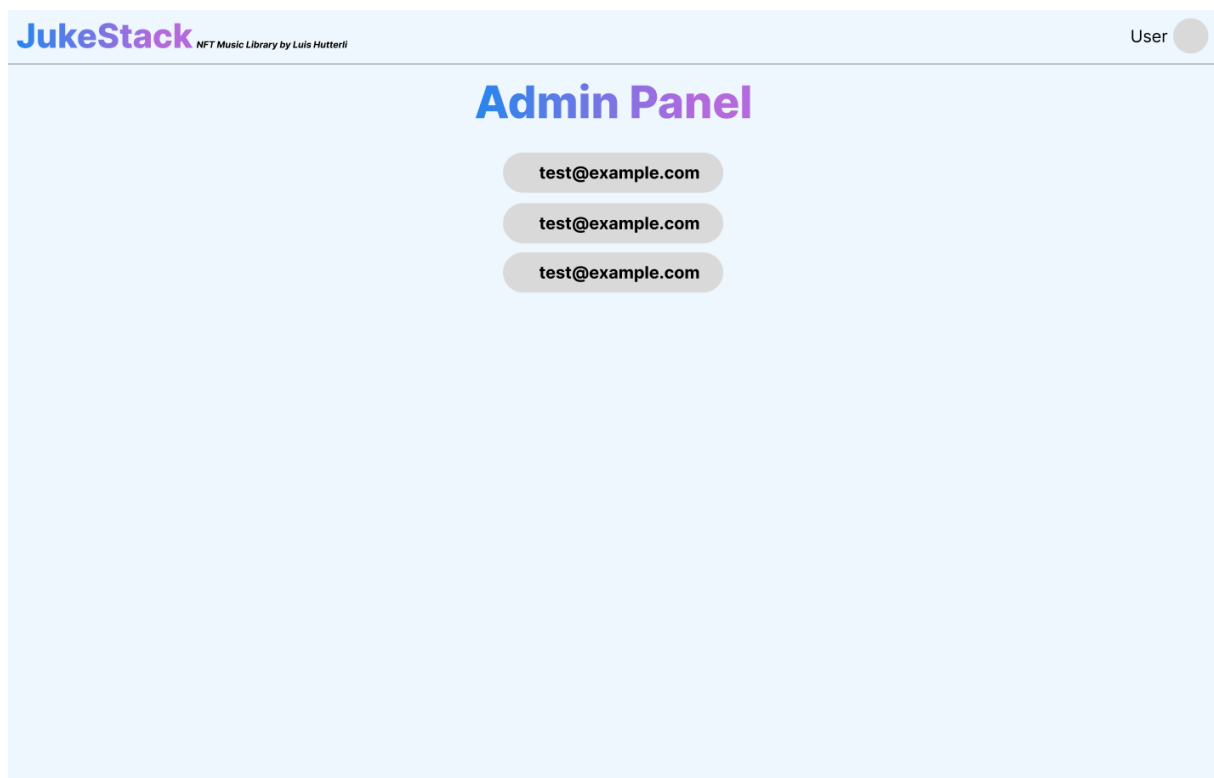
#### 3.1.1 Login/Signup

The wireframe shows a login page for "JukeStack" (NFT Music Library by Luis Hutterli). The page has a light blue background. In the top right corner, there is a "Login" button. In the center, there is a login form with the title "Login" in purple. The form contains two input fields: "Email:" and "Password:". Below the input fields is a blue "Login" button.

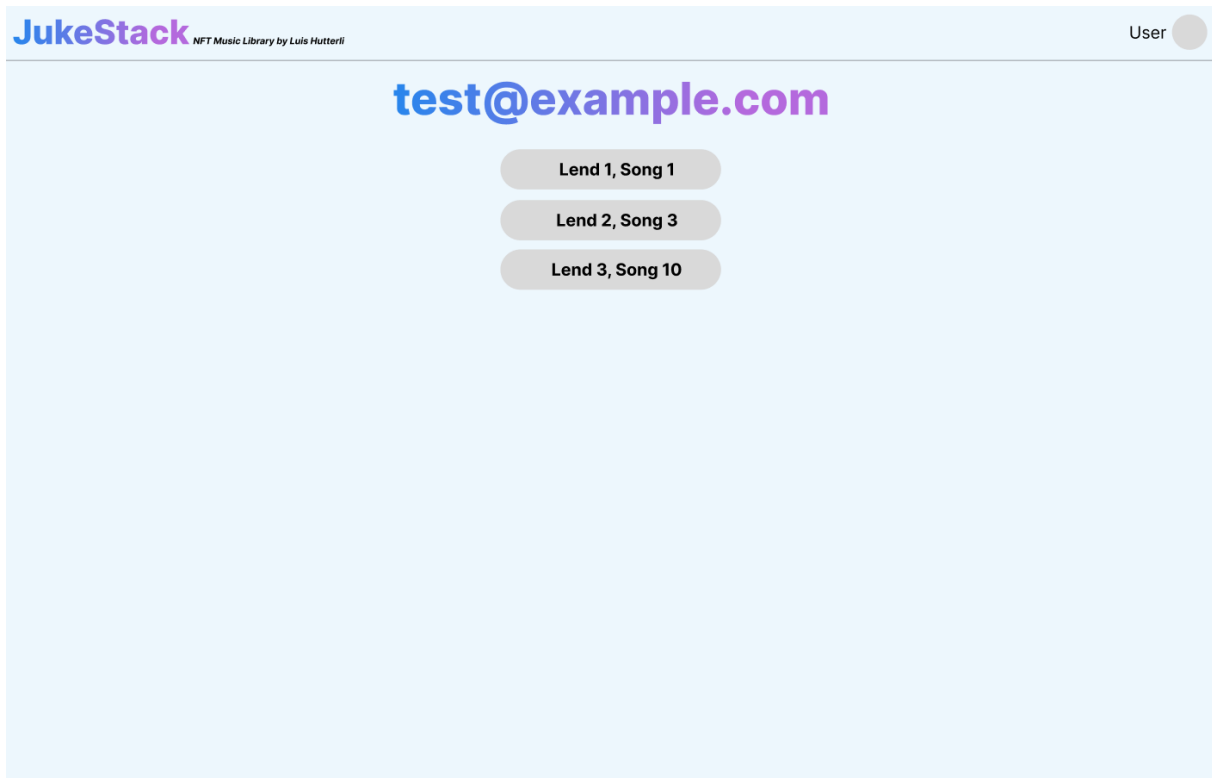
### 3.1.2 Main



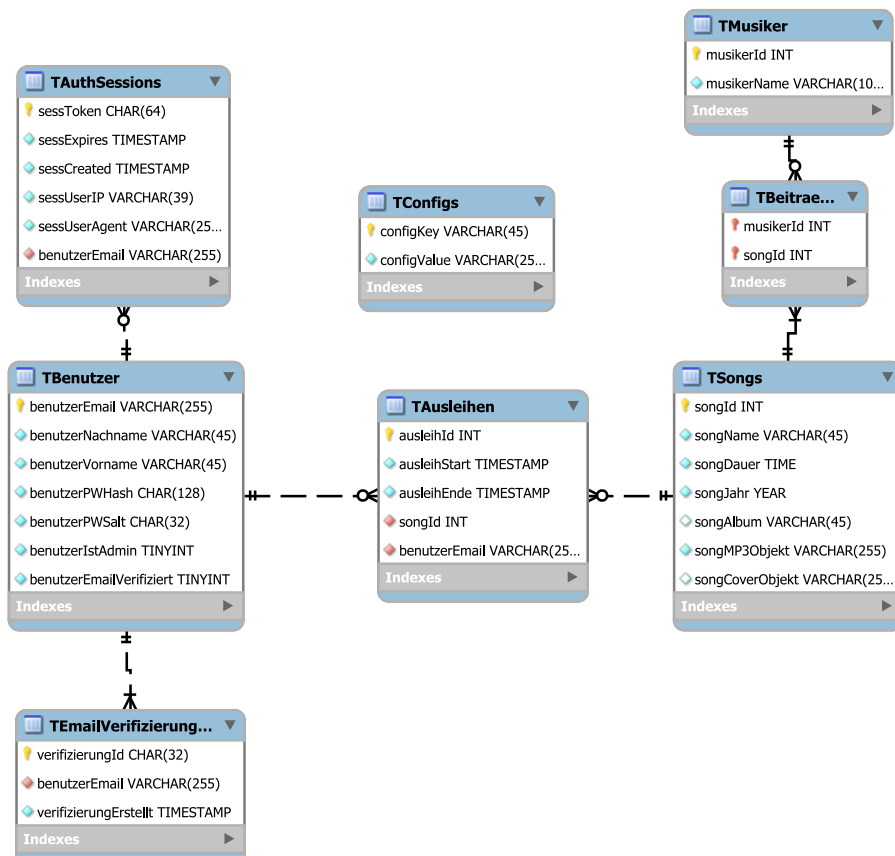
### 3.1.3 Admin – Alle Benutzer



### 3.1.4 Admin – Ausleihen eines Benutzers



### 3.2 Datenbankschema





### 3.3 API-Endpunkte

Basis: jukestack.ch/api, (APIs nach ch.lsh.ims.jukestack.MainVerticle.java)

Route	Method	Description	Response	Implemented
/user	POST	Create user	201	Fully Implemented
/user	GET	Get user info	200	Fully Implemented
/user	PUT	Update user info	200	Fully Implemented
/user	DELETE	Delete user	-	Not Implemented
/auth/login	POST	Login	201	Fully Implemented
/auth/logout	POST	Logout	200	Fully Implemented
/auth/verify	POST	Check Token validity	201	Fully Implemented
/auth/refresh	POST	Refresh Token	201	Fully Implemented
/auth/sendVerify	POST	Send verification email	200	Fully Implemented
/auth/verifyEmail	GET	Verify email via tokenized link	302	Fully Implemented
/songs	GET	List all songs	200	Fully Implemented
/lend	GET	List all borrowed songs	200	Fully Implemented
/lend/{id}	POST	Lend a song	200	Fully Implemented
/lend/{id}	DELETE	Return a song	200	Fully Implemented
/lend/{id}/listen	GET	Generate a link to song file	200	Fully Implemented
/admin/users	GET	List all users	200	Fully Implemented
/admin/users/{email}/lend	GET	List a users borrowed songs	200	Fully Implemented
/admin/lend/{id}	PUT	Update a users lend by lendId	200	Fully Implemented
/admin/lend/{id}	DELETE	Return a users lend by lendId	200	Fully Implemented

### 3.4 Testliste

Funktionale Testfälle			
Test	Erwartetes Ergebnis	Bestanden	Bemerkung
Layout Registrierung: Benutzer erstellen	Der Benutzer wird erfolgreich in der Datenbank gespeichert.		
Layout Anmelden: Benutzer Anmelden	Ein Session-Cookie wird erstellt, der Benutzer wird eingeloggt und zur App weitergeleitet.		
Layout Einstellungen: Benutzer Informationen werden Angezeigt	Benutzer Informationen werden in die vorgefertigten Felder eingesetzt		
Layout Einstellungen: Benutzer Informationen ändern	Die gewählten Benutzerinformationen werden aktualisiert. Bei Änderung der Email muss die Verifikation erneut durchgeführt werden.		
Alle Layouts: Im Header: Abmelden	Der Benutzer wird abgemeldet und auf die Start Seite geleitet. Die Session wird Serverseitig deaktiviert.		
Layout Main: Neuen Account erstellt oder Email geändert	Der Benutzer kann sich eine Verifizierungsmail schicken lassen.		
Verifizierungslink in der erhaltenen Email anklicken	Die Email des Benutzers wird verifiziert und er kann die App jetzt normal Benutzen.		
Layout Main: Songs anzeigen	Alle verfügbaren Songs werden korrekt angezeigt.		

Layout Main: Ausgeliehene Songs anzeigen	Alle ausgeliehenen Songs werden korrekt angezeigt.		
Layout Main: Song ausleihen (auf verfügbaren Song klicken)	Der Song wird in die persönliche Bibliothek hinzugefügt.		
Layout Main: Song zurückgeben (auf ausgeliehenen Song klicken)	Der Song wird aus der persönlichen Bibliothek entfernt.		
Layout Main: Song abspielen	Der Song wird abgespielt, und Ton ist hörbar.		
Layout Main: Maximal 5 Songs ausleihen	Eine Fehlermeldung wird angezeigt, und der Song wird nicht hinzugefügt.		
Admin - Alle Benutzer anzeigen	Alle Benutzer (außer andere Admins) werden angezeigt.		
Admin - Ausleihen eines Benutzers anzeigen	Alle ausgeliehenen Songs des Benutzers werden korrekt angezeigt.		
Admin - Ausleihe eines Benutzers zurückgeben	Der Song wird aus der Bibliothek des Benutzers entfernt.		
Admin - Ausleihe eines Benutzers ändern	Die neuen Ausleihtage werden korrekt gespeichert.		

Destruktive Testfälle			
Test	Erwartetes Ergebnis	Bestanden	Bemerkung
Benutzer erstellen - Unvollständige Eingaben	Eine Fehlermeldung wird angezeigt, und der Benutzer wird nicht erstellt.		
Anmelden - Falsche Zugangsdaten	Eine Fehlermeldung wird angezeigt, und der Benutzer wird nicht eingeloggt.		
Song ausleihen - Nicht vorhandener Song	Eine Fehlermeldung wird angezeigt.		
Song abspielen - Kein Zugriff	Der Song wird nicht abgespielt, und eine Fehlermeldung wird angezeigt.		
Admin - Ungültige Änderungen an Ausleihen	Eine Fehlermeldung wird angezeigt, und die Änderung wird nicht übernommen.		
Generell - Zugriff auf API Endpunkte mit Falscher oder Fehlender Authentifizierung	Der Zugriff sollte nicht erlaubt werden und ein Status 401 Unauthorized soll zurückgegeben werden.		
Generell - Zugriff auf bestimmte API Endpunkte ohne die Email bestätigt zu haben	Der Zugriff sollte nicht erlaubt werden und der Benutzer sollte zu einer entsprechenden Verifizierungsseite weitergeleitet werden.		

## 4 Realisation

### 4.1 Backend

Ich habe mich für ein Java-Backend entschieden, da ich auf der suche nach einer guten Serverless Cloud Plattform gesucht habe. Hierzu habe ich diese Webseite verwendet:

<https://www.srvlss.io/>

Ursprünglich wollte ich das Backend in Rust schreiben und auf Cloudflare Workers deployen, jedoch war die Integration mit dritten SQL-Datenbank zu aufwändig und ich musste eine andere Plattform suchen.

Ich habe dann schliesslich Google Cloud Run gefunden und ausgewählt, da es sehr flexibel ist, ich ein wenig Erfahrung mit der Google Cloud Plattform habe und es für meinen Anwendungsfall gratis ist. GCR (Google Cloud Run) kann direkt mit Java verwendet werden, jedoch mit SpringBoot, zum glück kann man jedoch sein eigenes Deployment mit Docker erstellen und jede Sprache und Framework deployen. Ich habe mich also für einen Vert.X Server entschieden, da er nach meiner Recherche als einer der schnellsten und flexibelsten herausging.

### 4.2 Frontend

Das Frontend ist eine React Webapplikation, statt JavaScript habe ich TypeScript verwendet.

Als Packager habe ich die Vite Framework verwendet, da sie das Entwickeln und Deployen einfach etwas schneller macht.

Um das Frontend zu Deployen habe ich Google Firebase Hosting verwendet, da ich Firebase schon sehr oft verwendet habe und gute Erfahrungen damit gemacht habe.

#### 4.2.1 Routing

Die Applikation verwendet ReactRouter und die BrowserRouter Komponente, dass bedeutet, dass Routes bzw. Pfade gleich funktionieren wie bei einer herkömmlichen Webseite (z.B. <https://jukestack.ch/app/login>). Die andere Variante wäre die HashRouter komponente gewesen, jedoch wird hier vor dem Pfad auch immer ein # Platziert (z.B. <https://jukestack.ch/#/app/login>). Jedoch funktioniert der direkte Zugriff auf einen Unterpfad mit dem Browserrouter nicht direkt, da zuerst alle Anfragen auf alle Unterpfade zum Hauptpfad und dem index.html weitergeleitet werden müssen.

Das erziele ich mit einer speziellen Konfiguration in [firebase.json](#);

Dort «rewrite» ich alle Anfragen auf allen Pfaden zu index.html.

Auch kann ich durch die direkte Integration von Google Firebase in die Google Cloud Plattform den GCR Container (Backend) auf den Pfad /api binden, da ich alle Anfragen auf diesen Pfad an den GCR Container weiterleite.

## 5 Test

Funktionale Testfälle			
Test	Erwartetes Ergebnis	Bestanden	Bemerkung
Layout Registrierung: Benutzer erstellen	Der Benutzer wird erfolgreich in der Datenbank gespeichert.	Ja	-
Layout Anmelden: Benutzer Anmelden	Ein Session-Cookie wird erstellt, der Benutzer wird eingeloggt und zur App weitergeleitet.	Ja	-
Layout Einstellungen: Benutzer Informationen werden Angezeigt	Benutzer Informationen werden in die vorgefertigten Felder eingesetzt	Ja	-
Layout Einstellungen: Benutzer Informationen ändern	Die gewählten Benutzerinformationen werden aktualisiert. Bei Änderung der Email muss die Verifikation erneut durchgeführt werden.	Ja	-
Alle Layouts: Im Header: Abmelden	Der Benutzer wird abgemeldet und auf die Start Seite geleitet. Die Session wird Serverseitig deaktiviert.	Ja	-
Layout Main: Neuen Account erstellt oder Email geändert	Der Benutzer kann sich eine Verifizierungsmail schicken lassen.	Ja	-
Verifizierungslink in der erhaltenen Email anklicken	Die Email des Benutzers wird verifiziert und er kann die App jetzt normal Benutzen.	Ja	-
Layout Main: Songs anzeigen	Alle verfügbaren Songs werden korrekt angezeigt.	Ja	-
Layout Main: Ausgeliehene Songs anzeigen	Alle ausgeliehenen Songs werden korrekt angezeigt.	Ja	-
Layout Main: Song ausleihen (auf verfügbaren Song klicken)	Der Song wird in die persönliche Bibliothek hinzugefügt.	Ja	-
Layout Main: Song zurückgeben (auf ausgeliehenen Song klicken)	Der Song wird aus der persönlichen Bibliothek entfernt.	Ja	-
Layout Main: Song abspielen	Der Song wird abgespielt, und Ton ist hörbar.	Ja	-
Layout Main: Maximal 5 Songs ausleihen	Eine Fehlermeldung wird angezeigt, und der Song wird nicht hinzugefügt.	Ja	-
Admin - Alle Benutzer anzeigen	Alle Benutzer (außer andere Admins) werden angezeigt.	Ja	-
Admin - Ausleihen eines Benutzers anzeigen	Alle ausgeliehenen Songs des Benutzers werden korrekt angezeigt.	Ja	-
Admin - Ausleihe eines Benutzers zurückgeben	Der Song wird aus der Bibliothek des Benutzers entfernt.	Ja	-
Admin - Ausleihe eines Benutzers ändern	Die neuen Ausleihtage werden korrekt gespeichert.	Ja	-
Tests durchgeführt am 24.01.2025 von Luis Hutterli		LH	

Destruktive Testfälle			
Test	Erwartetes Ergebnis	Bestanden	Bemerkung
Benutzer erstellen - Unvollständige Eingaben	Eine Fehlermeldung wird angezeigt, und der Benutzer wird nicht erstellt.	Ja	-
Anmelden - Falsche Zugangsdaten	Eine Fehlermeldung wird angezeigt, und der Benutzer wird nicht eingeloggt.	Ja	-
Song ausleihen - Nicht vorhandener Song	Eine Fehlermeldung wird angezeigt.	Ja	-
Song abspielen - Kein Zugriff	Der Song wird nicht abgespielt, und eine Fehlermeldung wird angezeigt.	Ja	-
Admin - Ungültige Änderungen an Ausleihen	Eine Fehlermeldung wird angezeigt, und die Änderung wird nicht übernommen.	Ja	-
Generell - Zugriff auf API Endpunkte mit Falscher oder Fehlender Authentifizierung	Der Zugriff sollte nicht erlaubt werden und ein Status 401 Unauthorized soll zurückgegeben werden.	Ja	-
Generell - Zugriff auf bestimmte API Endpunkte ohne die Email bestätigt zu haben	Der Zugriff sollte nicht erlaubt werden und der Benutzer sollte zu einer entsprechenden Verifizierungsseite weitergeleitet werden.	Ja	-
Tests durchgeführt am 24.01.2025 von Luis Hutterli		LH5	

## 5.1 Known Bugs

### 5.1.1 iOS Mobile Audio Playback

Das Abspielen von Songs funktioniert leider nicht auf mobilen Apple-Geräten in der WebKit-Engine. Grund dafür ist ein Feature das ursprünglich zum Datensparen entwickelt wurde, und es erst erlaubt Audio-Dateien zu laden, wenn dies durch eine Benutzer-Interaktion ausgelöst wird. Bedeutet: Audio-Dateien können nur abgespielt werden, wenn das direkt durch einen Knopfdruck geschieht.

Diese Limitierung scheint erstmals kein Problem zu sein, jedoch erkennt WebKit das Drücken eines Knopfes in React nicht als Benutzeraktion. Das liegt sehr wahrscheinlich an der zusätzlichen Komplexität die React mit sich bringt. Auch das Setzen einer onclick Methode über Vanilla JavaScript funktioniert in diesem Kontext nicht.

Aufgrund dieser Umstände unterstützt die Applikation zurzeit keine mobilen Apple-Geräte.

## 6 Deployment

### 6.1 Benutzeranleitung

Mehr über dieses Thema findet man im README.

#### 6.1.1 Backend

Das Backend kann mit dem deploy.sh Script in ./backend/ direkt zu Google Cloud Run Deployed werden. Hierzu braucht es lediglich ein Google Cloud Projekt mit den richtigen Berechtigungen und die entsprechenden Umgebungsvariablen im Google Secrets Manager.

Lokal kann das Backend mit den zwei Scripts build-docker.sh und run-docker.sh in ./backend/ ausgeführt werden. Hierzu braucht man auch keine jdk, nur Docker.

### 6.1.2 Frontend

Das Frontend wird mit dem Befehl «npm run build» gebündelt und dann mit dem Befehl «firebase deploy» auf eine Google Firebase Seite deployed.

Lokal kann man die webseite wie folgt ausführen: «npm run dev»

(für beides müssen zuerst die Dependencies installiert werden: «npm i»)

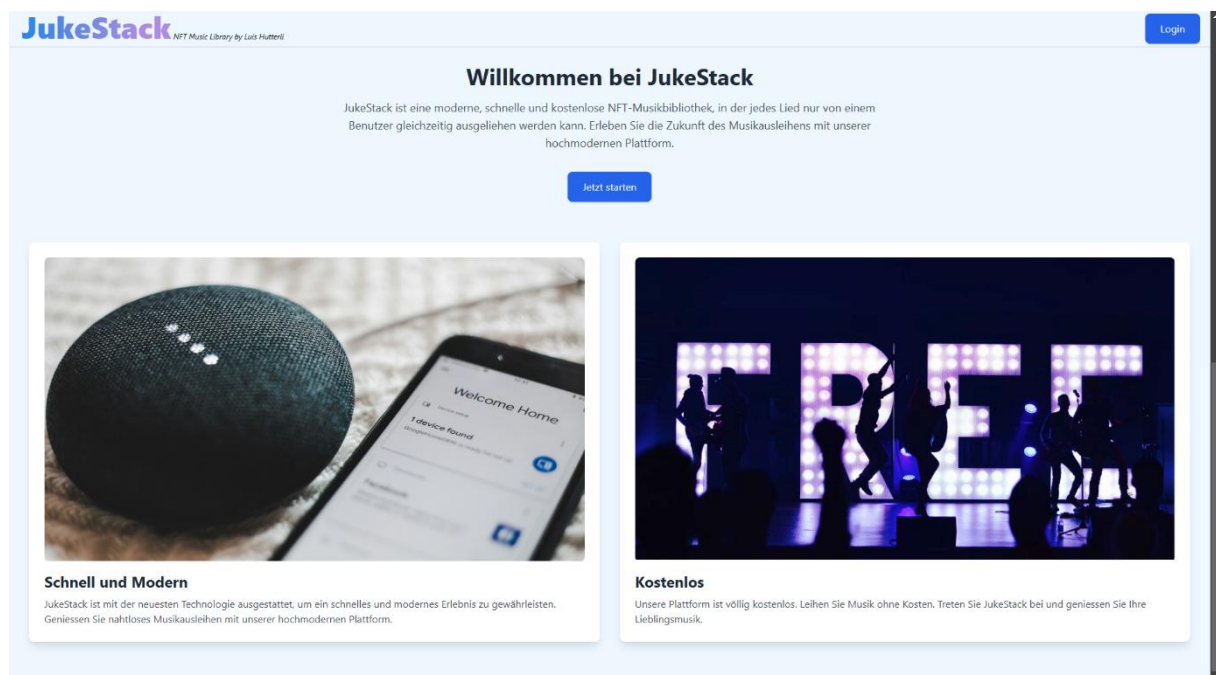
## 6.2 Live Deployment

Auf <https://jukestack.ch/>

Login auf <https://jukestack.ch/app/login>

## 7 Benutzeranleitung

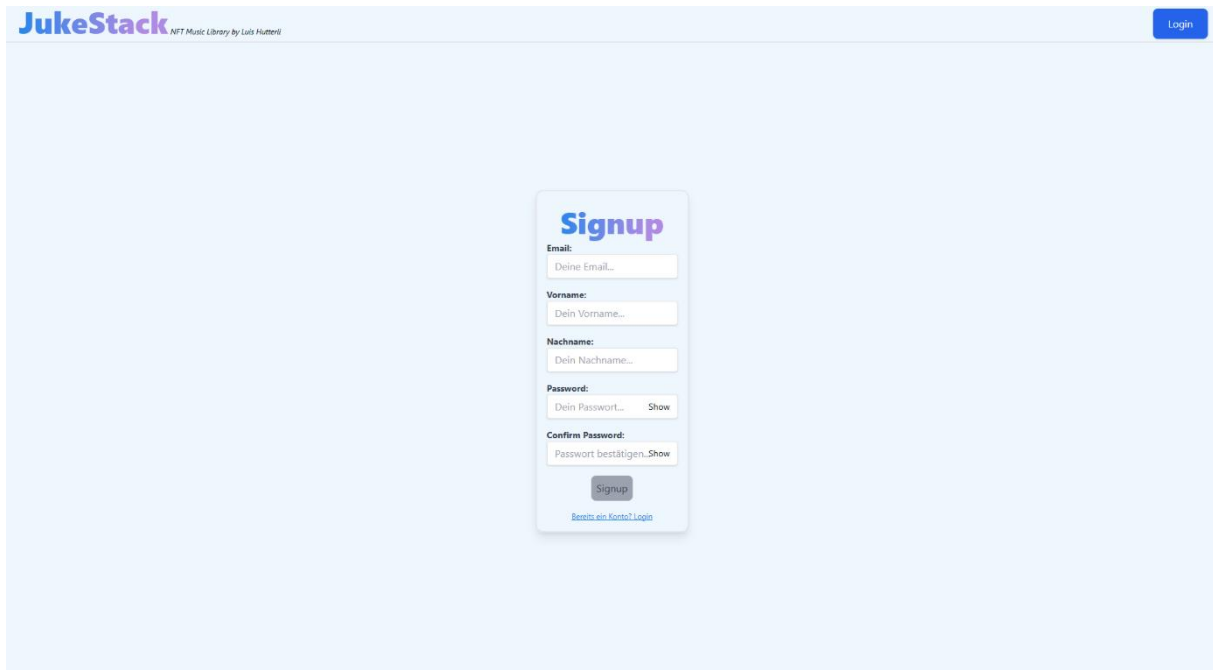
### 7.1 Landing



Das ist die Landing / Startseite wenn man nach unten scrollt. Man kann durch den Knopf «Jetzt Starten» oder durch den Knopf «Login» (im Header) auf die Login Seite wechseln.

## 7.2 Account

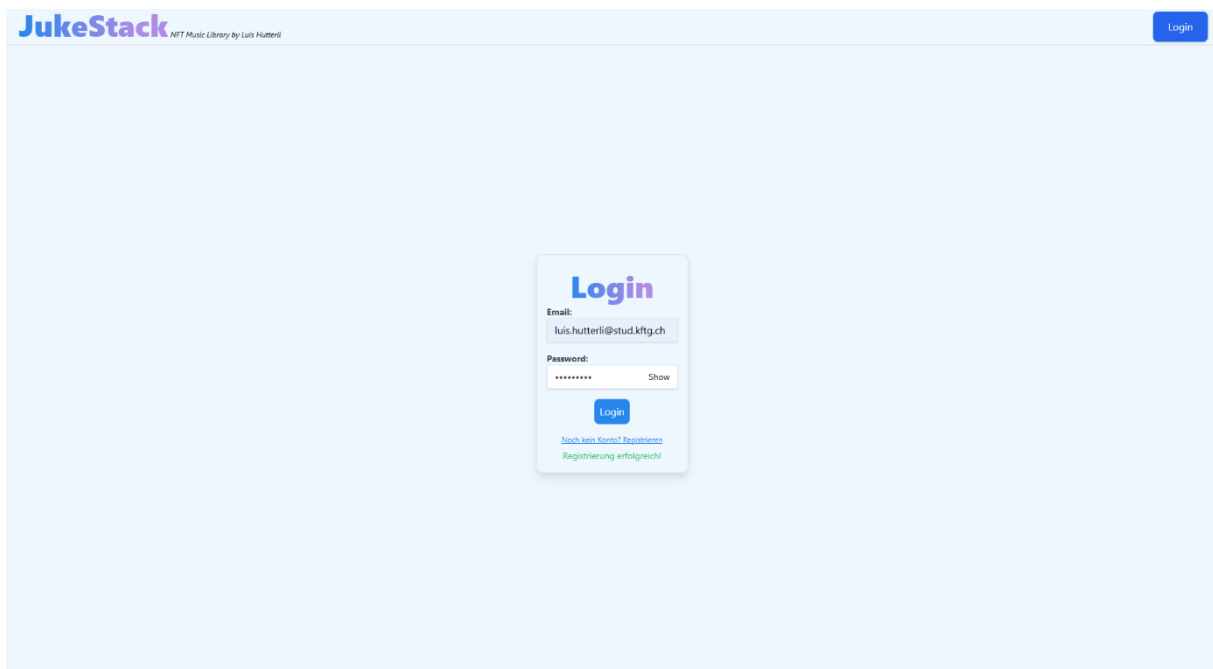
### 7.2.1 Erstellen



The screenshot shows the 'Signup' form on the JukeStack website. The header includes the 'JukeStack' logo and the tagline 'NFT Music Library by Luis Hutterli', along with a 'Login' button. The form itself is titled 'Signup' and contains the following fields: 'Email:' with a placeholder 'Deine Email...', 'Vorname:' with a placeholder 'Dein Vorname...', 'Nachname:' with a placeholder 'Dein Nachname...', 'Password:' with a placeholder 'Dein Passwort...' and a 'Show' link, and 'Confirm Password:' with a placeholder 'Passwort bestätigen' and a 'Show' link. Below the fields is a 'Signup' button and a link that says 'Bereits ein Konto? Login'.

Hier kann der Benutzer ein neues Konto unter Angabe seiner E-Mail, seines Namens und eines Passwortes erstellen. Achtung die E-Mail-Adresse muss gültig sein.

### 7.2.2 Anmelden

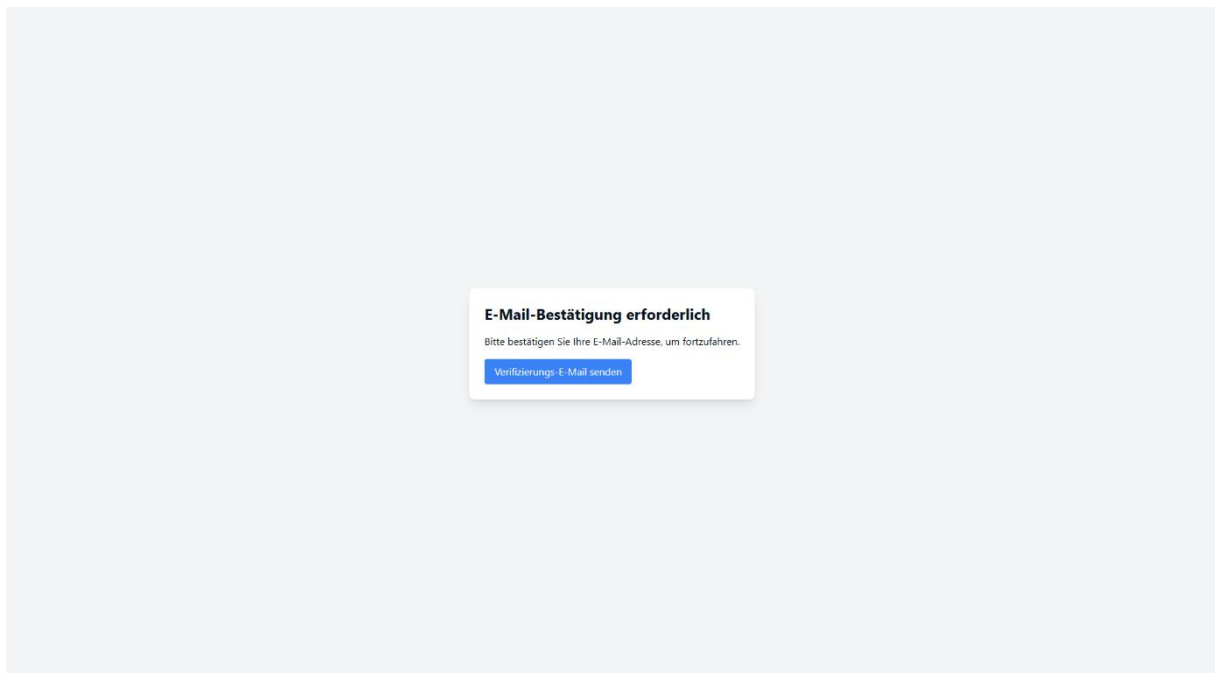


The screenshot shows the 'Login' form on the JukeStack website. The header is identical to the previous page. The form is titled 'Login' and contains two fields: 'Email:' with the value 'luis.hutterli@stud.kftg.ch' and 'Password:' with masked characters '\*\*\*\*\*' and a 'Show' link. Below the fields is a 'Login' button. At the bottom of the form, there is a link 'Noch kein Konto? Registrierung' and a green message 'Registrierung erfolgreich!'.

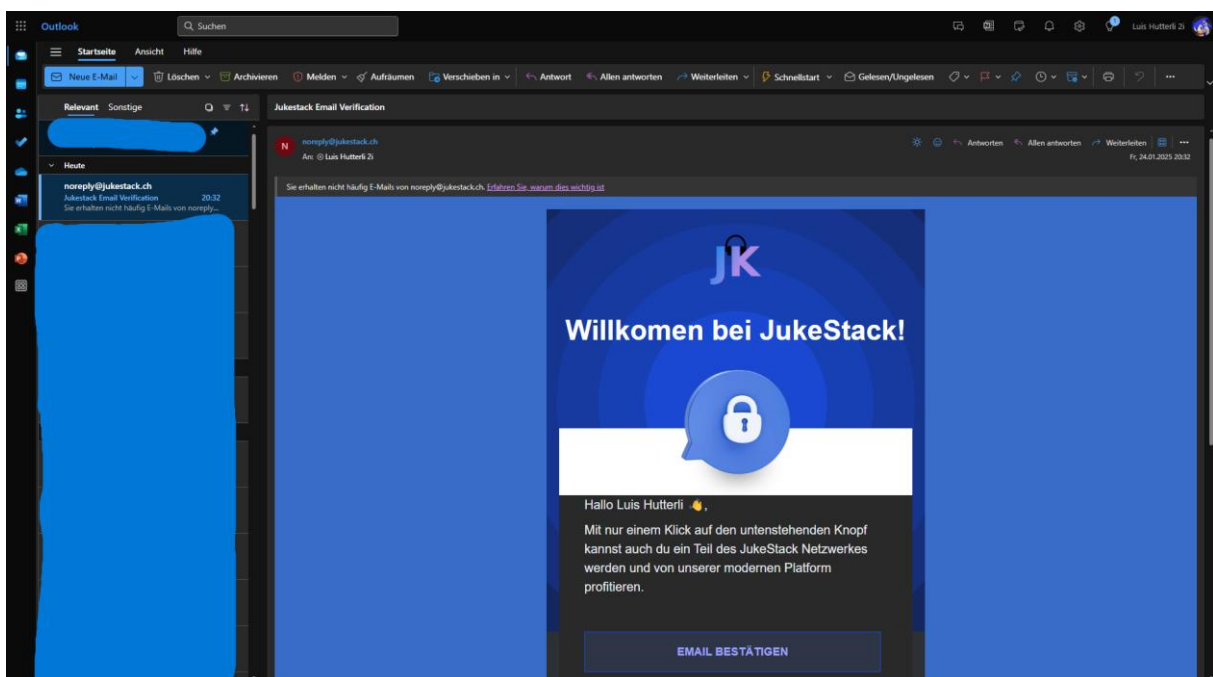
Hier kann der Benutzer sich mit seinem eben erstellten Konto anmelden.



### 7.2.3 E-Mail-Verifizieren



Durch Klicken auf den Knopf kann der Benutzer sich eine Verifizierungs-E-Mail schicken lassen.



Der Benutzer erhält eine ähnliche E-Mail und kann sich durch Drücken auf den Knopf verifizieren und wird automatisch auf die Hauptseite weitergeleitet.

## 7.2.4 Einstellungen

Durch Klicken auf die E-Mail-Adresse in sämtlichen Header kann der Benutzer auf seine Einstellungen zugreifen.

**JukeStack** NFT Music Library by Luis Hutterli luis.hutterli@stud.kftg.ch

**Einstellungen**

**Email ändern**

Neue Email  
luis.hutterli@stud.kftg.ch

Email speichern

**Name ändern**

Vorname  
Luis

Nachname  
Hutterli

Name speichern

**Passwort ändern**

Neues Passwort

Passwort bestätigen

Passwort speichern

Hier kann der Benutzer die Einstellungen von seinem Konto einsehen und bearbeiten. Achtung bei einer Änderung der E-Mail-Adresse muss der Benutzer diese auch Verifizieren.

## 7.3 Hauptseite – Desktop

### 7.3.1 Song Ausleihen

**JukeStack** NFT Music Library by Luis Hutterli luis.hutterli@stud.kftg.ch

**JK Bibliothek**

- 10 Freaky Girls (with 21 Savage) - 3:30 by Metro Boomin, 21 Savage **Ausleihen**
- 500lbs - 2:33 by Lil Tecca **Ausleihen**
- All Eyes On Me - 3:00 by VALORANT, JeanDesux, Whipped Cream **Ausleihen**
- Au Revoir - 2:42 by Milano, Eddin **Ausleihen**
- Back One Day (Outro Song) - 3:54 by NEFFEX, TheFatRat **Ausleihen**
- B@cane - A COLORS SHOW - 3:05 by Yamé **Ausleihen**
- 90210 - 5:40 by Travis Scott, Kacy Hill **Ausleihen**
- All Eyes On Me - 5:07 by 2Pac, Big Syke **Ausleihen**
- HIGHEST IN THE ROOM - 2:57 by Travis Scott **Ausleihen**
- Timeless - 4:16 by Playboi Carti, the Weeknd **Ausleihen**

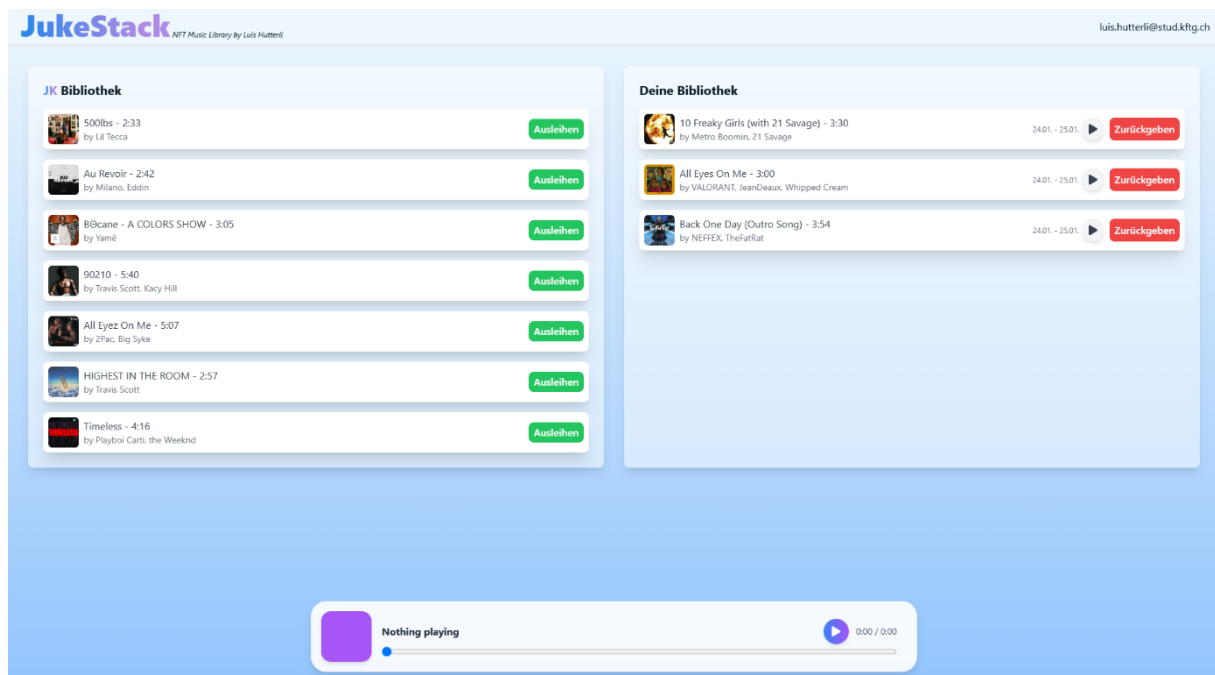
**Deine Bibliothek**

Keine Einträge

Nothing playing 0:00 / 0:00

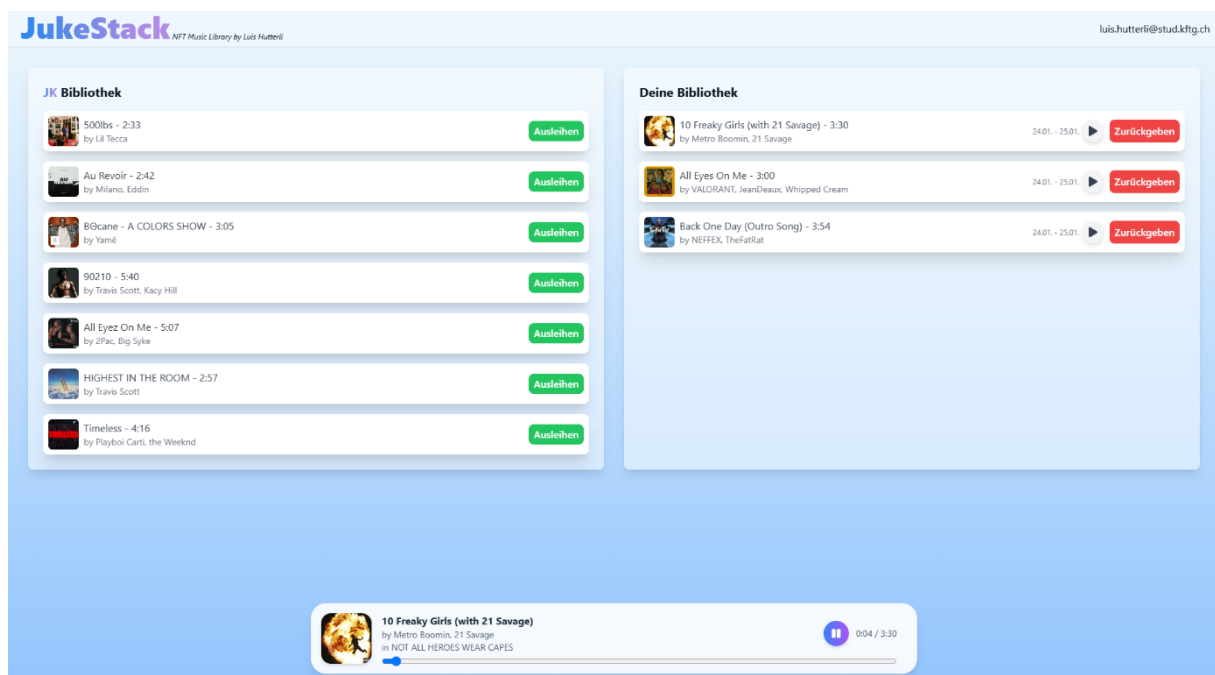
Durch Klicken auf den Knopf «Ausleihen» kann der Benutzer den gewünschten Song ausleihen und in seine Bibliothek hinzufügen.

### 7.3.2 Song Zurückgeben



Durch Klicken auf den Roten Knopf «Zurückgeben» kann der Benutzer eine beliebige Ausleihe wieder zurückgeben.

### 7.3.3 Song Abspielen / Steuern

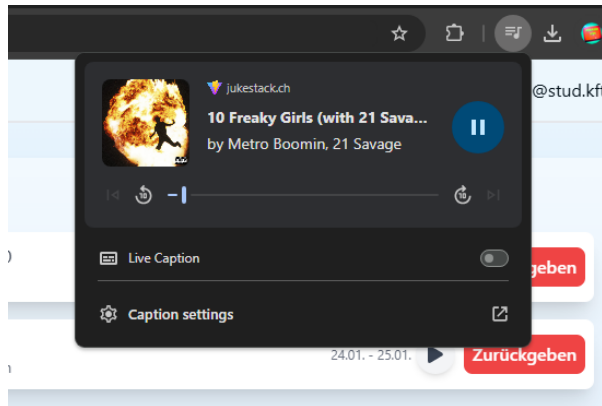


Durch das Klicken auf den Abspiel Knopf (Dreieck neben dem Zurückgeben Knopf) kann der Benutzer den gewünschten Song abspielen und er wird in den unteren Musik-Player eingefügt.

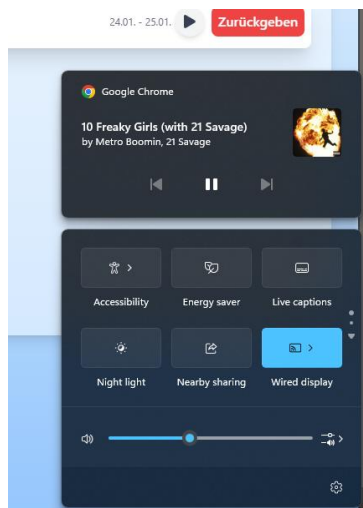
Im Musik Player kann der Song beliebig pausiert und wieder abgespielt werden und durch das Benutzen des unteren Schiebereglers kann auch gespult werden.

### 7.3.4 Song Steuern über OS-Integration

Wenn der Benutzer wie oben erklärt einen Song anwählt und abspielt, dann wird dieser Song auch einer «mediaSession» beigefügt, wodurch die Steuerung auch durch Betriebssystem oder Browser Komponenten möglich ist.



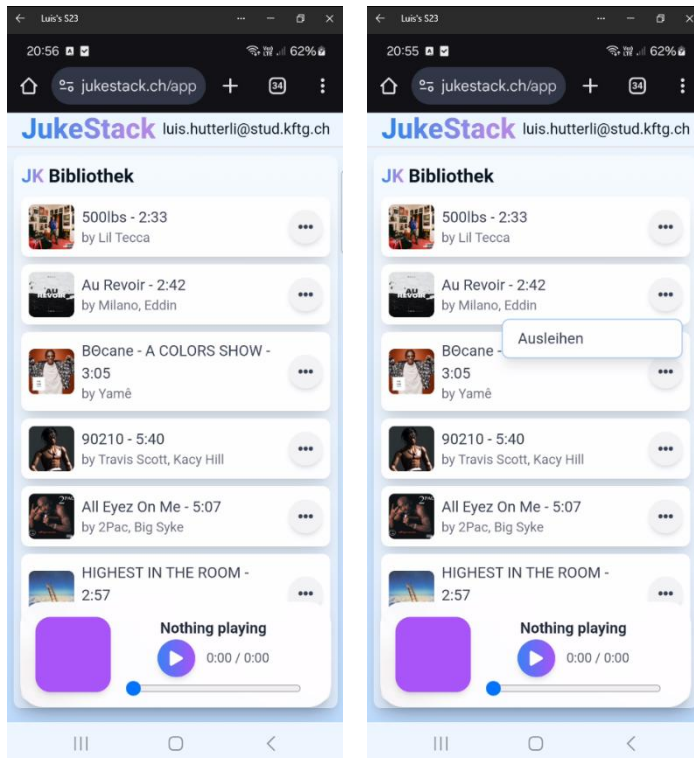
Hier ein Beispiel der Chrome Komponente.



Hier ein Beispiel der Windows Komponente.

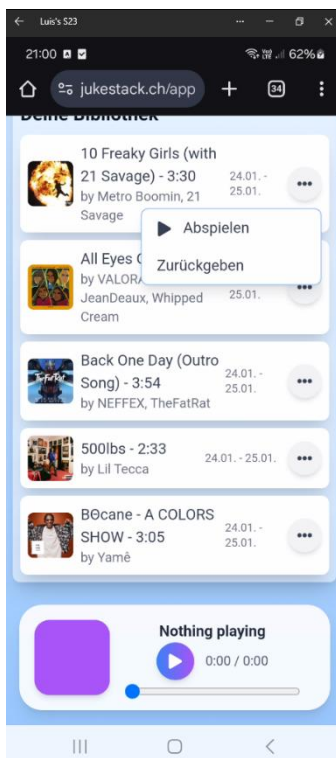
## 7.4 Hauptseite – Mobile

### 7.4.1 Song Ausleihen



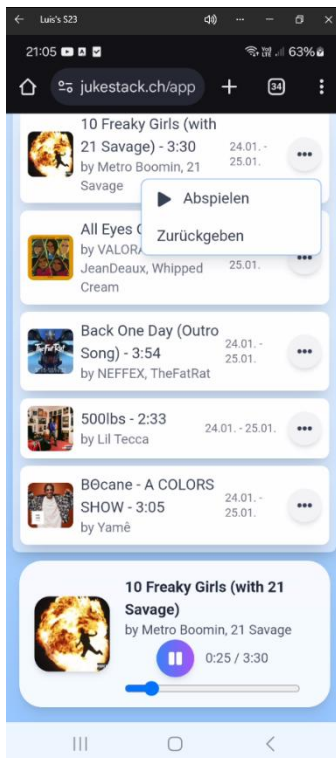
Durch das Klicken auf das Menu-Knopf (linkes Bild) kann der Benutzer das Menu des Songs öffnen und durch den Unterpunkt «Ausleihen» den Song ausleihen und zu seiner Bibliothek hinzufügen.

### 7.4.2 Song Zurückgeben



Ein beliebiger Song aus der eigenen Bibliothek kann durch das Öffnen des Song-Menüs (oben beschrieben) und durch Anwählen des Unterpunkts «Zurückgeben» zurückgegeben werden.

### 7.4.3 Song Abspielen / Steuern

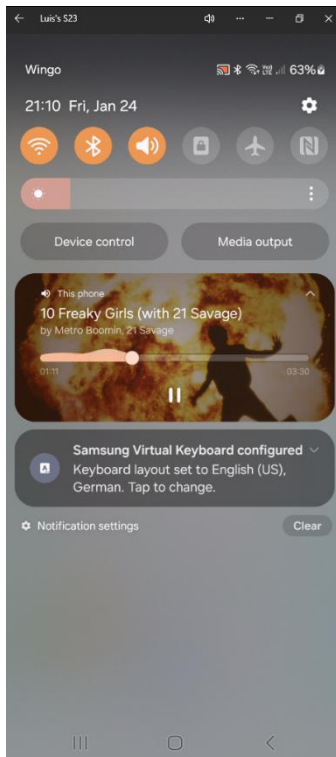


Ein beliebiger Song aus der eigenen Bibliothek kann durch das Öffnen des Song-Menüs (oben beschrieben) und durch Anwählen des Unterpunkts «Abspielen» abgespielt werden und wird in den untenstehenden Musik-Player eingefügt.

Im Musik Player kann der Song beliebig pausiert und wieder abgespielt werden und durch das Benutzen des unteren Schiebereglers kann auch gespult werden.

### 7.4.4 Song Steuern über OS-Integration

Wenn der Benutzer wie oben erklärt einen Song anwählt und abspielt, dann wird dieser Song auch einer «mediaSession» beigefügt, wodurch die Steuerung auch durch Betriebssystem oder Browser Komponenten möglich ist.

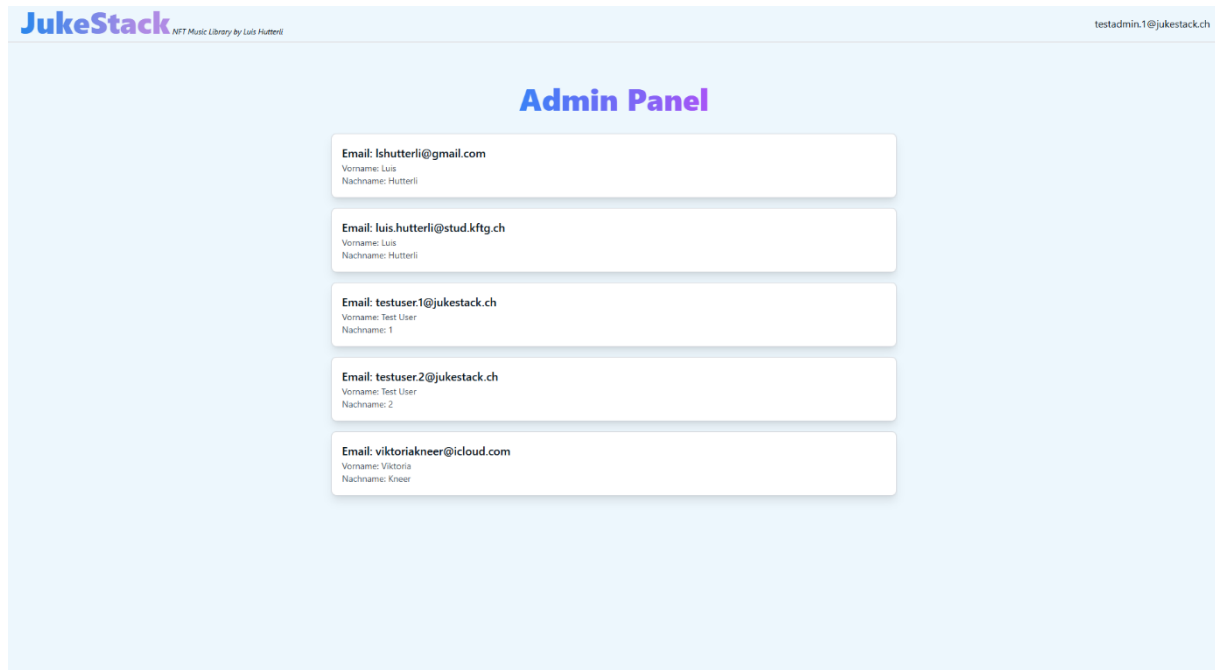


Hier ein Beispiel des Samsung Musik-Players (OneUI 6.1)

Hinweis: So kann ein Song auch abgespielt und gesteuert werden, wenn das Handy gesperrt oder im Standby ist.

## 7.5 Admin

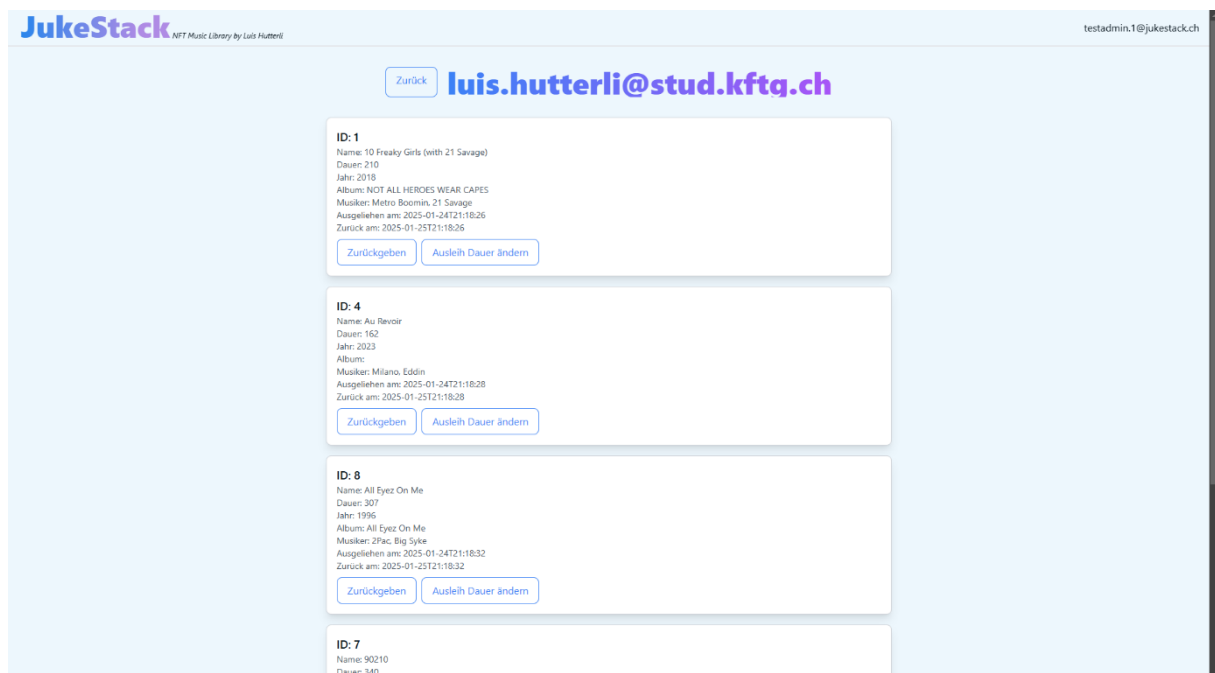
### 7.5.1 Startseite – Liste aller Benutzer



Wenn sich ein Admin anmeldet, wird er automatisch auf dieses Admin-Panel weitergeleitet. Hier kann der Admin eine Liste von allen Benutzern und einiger ihrer wichtigen Eckdaten sehen.

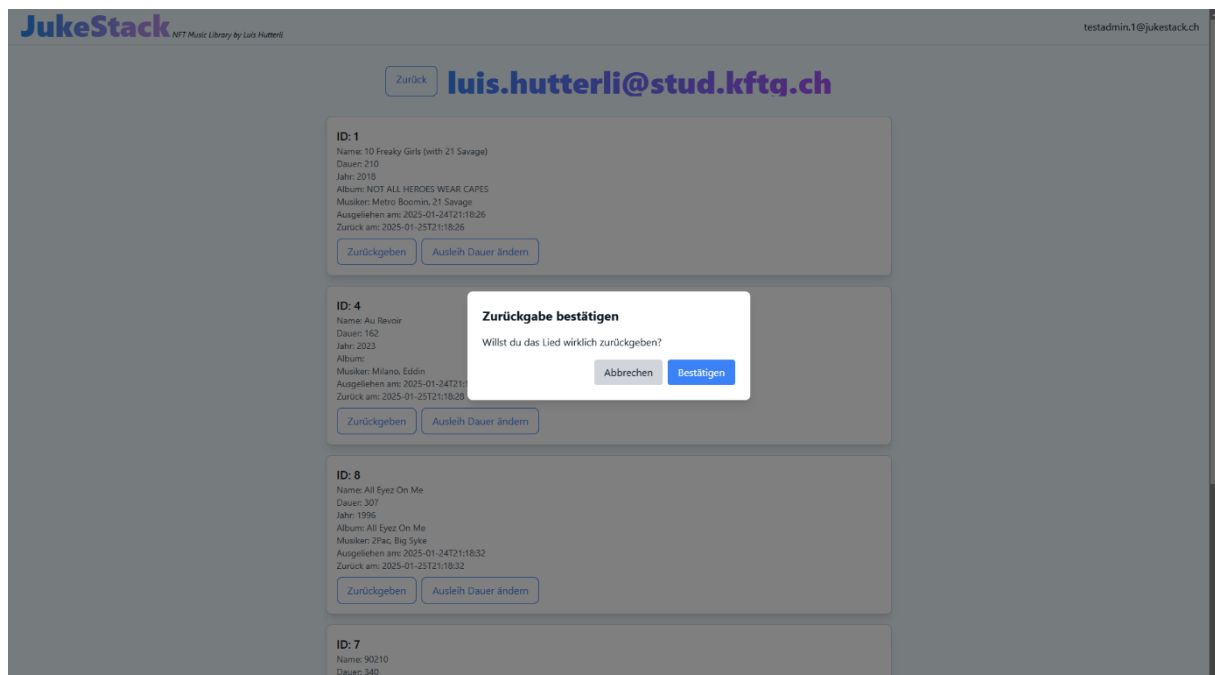
Durch Klicken auf einen Benutzer wird der Admin auf die Einzelansicht des Benutzers weitergeleitet.

### 7.5.2 Benutzer – Liste der Ausleihen



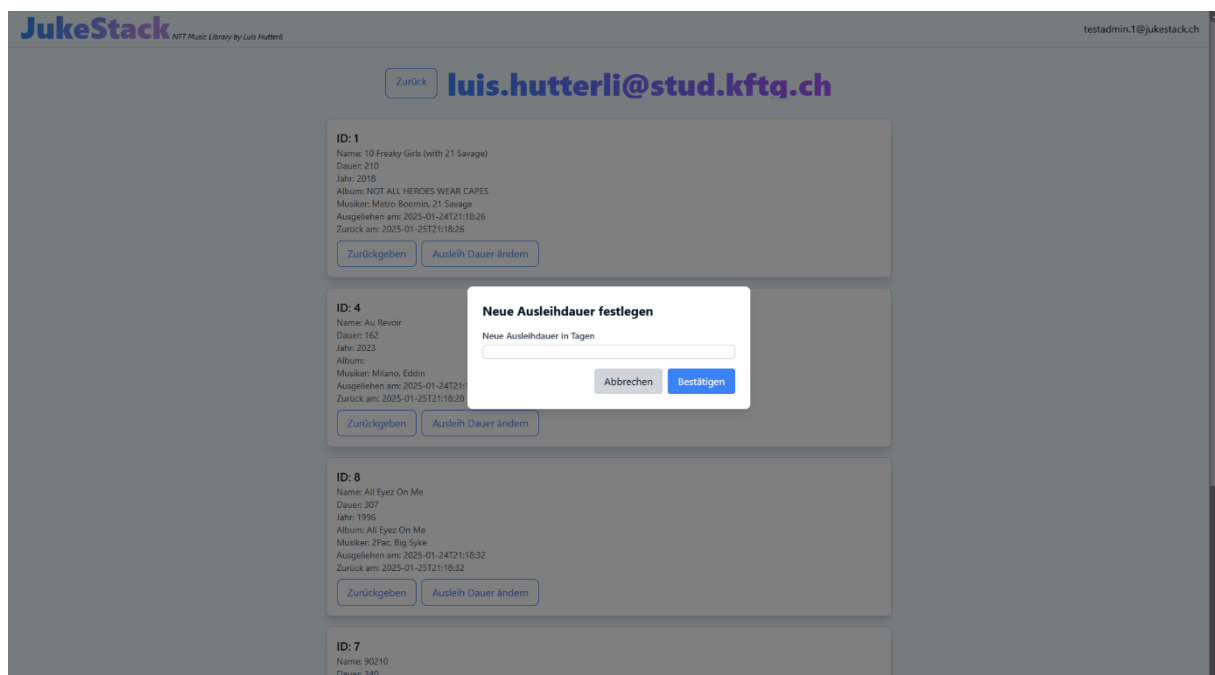
In dieser Einzel Ansicht kann der Admin die aktuellen Ausleihen vom Benutzer einsehen und bearbeiten.

### 7.5.3 Ausleihe – Zurückgeben



Durch Klicken auf «Zurückgeben» in der Liste aller Ausleihen wird dieser Dialog angezeigt, wo der Admin seine Entscheidung bestätigen muss und so dann den Song zurückgeben kann.

### 7.5.4 Ausleihe – Dauer ändern



Durch Klicken auf «Zurückgeben» in der Liste aller Ausleihen wird dieser Dialog angezeigt, wo der Admin die Ausleihdauer in Tagen festlegen kann.