**System Spezifikation**  
**(Pflichtenheft)**

**ShareList**

|  |  |
| --- | --- |
| Dokumentbezeichnung | T-3\_ShareList\_Pflichtenheft\_v01 |
| Version | 1.0 |
| Datum | 24.05.2022 |
| Status | Fertig |
| Vertraulichkeit | Projektmitglieder und Betreuer |
| Dokumentnummer | SYSSPEC-ShareList-0001 |
| Verteiler | Adrian Schauer, Fabian Lasser, Raymond Ermler, Mirac Fidanci, Luka Grgic, Peter Bierbamer und Robert Kofler |

HTL Spengergasse, Team 3

**Inhaltsverzeichnis**

[0 Versionshistorie 4](#_Toc104287807)

[1 Einleitung 5](#_Toc104287808)

[1.1 Zweck des Dokuments 5](#_Toc104287809)

[1.2 Umfang 5](#_Toc104287810)

[1.3 Definitionen, Begriffsklärungen, Abkürzungen 5](#_Toc104287811)

[1.4 Verweise 5](#_Toc104287812)

[1.5 Übersicht 6](#_Toc104287813)

[2 Systemübersicht 7](#_Toc104287814)

[2.1 Produktsicht 7](#_Toc104287815)

[2.2 Systemschnittstellen 7](#_Toc104287816)

[2.2.1 Benutzerschnittstellen 7](#_Toc104287817)

[2.2.2 Hardware-Schnittstellen 7](#_Toc104287818)

[2.2.4 Kommunikationsschnittstellen 7](#_Toc104287819)

[2.3 Speichereinschränkungen 7](#_Toc104287820)

[2.4 Betrieb 8](#_Toc104287821)

[2.5 Standortabhängige Anforderungen 8](#_Toc104287822)

[2.6 Benutzercharakteristik 8](#_Toc104287823)

[2.7 Zusammenfassung der Produktfunktionen 8](#_Toc104287824)

[2.8 Anforderungen an die Dokumentation 8](#_Toc104287825)

[2.9 Einschränkungen 8](#_Toc104287826)

[2.10 Lieferumfang 8](#_Toc104287827)

[2.11 Annahmen und Abhängigkeiten 8](#_Toc104287828)

[2.12 Anforderungsaufteilung 8](#_Toc104287829)

[3 Anforderungsdetails 9](#_Toc104287830)

[3.1 Externe Schnittstellen 9](#_Toc104287831)

[3.2 Detaillierte Funktionsbeschreibung 9](#_Toc104287832)

[3.2.1 Funktion /01/ 9](#_Toc104287833)

[3.2.2 Funktion /02/ 9](#_Toc104287834)

[3.2.3 Funktion /03/ 9](#_Toc104287835)

[3.2.4 Funktion /04/ 10](#_Toc104287836)

[3.2.5 Funktion /05/ 10](#_Toc104287837)

[3.2.6 Funktion /06/ 10](#_Toc104287838)

[3.2.7 Funktion /07/ 11](#_Toc104287839)

[3.2.8 Funktion /08/ 11](#_Toc104287840)

[3.2.9 Funktion /09/ 12](#_Toc104287841)

[3.3 Klassen und Module 14](#_Toc104287842)

[3.4 Datenbank Anforderungen 15](#_Toc104287843)

[3.4.1 Datenmodell und Dateien 15](#_Toc104287844)

[3.4.2 Entity-Relationship-Model 16](#_Toc104287845)

[3.4.3 Tabellendefinitionen 16](#_Toc104287846)

[3.5 Performanzanforderungen 17](#_Toc104287847)

[3.6 Entwurfseinschränkungen 17](#_Toc104287848)

[3.7 Qualitätsanforderungen, Systemcharakteristika 17](#_Toc104287849)

[4 Testfälle 18](#_Toc104287850)

[4.1 18](#_Toc104287851)

[4.2 18](#_Toc104287852)

[4.3 18](#_Toc104287853)

[4.4 19](#_Toc104287854)

[4.5 19](#_Toc104287855)

# 0 Versionshistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Version Nr. | Autor | Datum | Art und Grund der Änderung |
| 0.1 | Fabian Lasser, Adrian Schauer, Raymond Ermler, Mirac Fidanci, Luka Grgic | 2022-05-16 | Erstversion |
| 0.2 | Fabian Lasser, Adrian Schauer, Mirac Fidanci, Ermler Raymond, Luka Grgic | 2022-05-17 | Weitergearbeitet |
| 0.8 | Fabian Lasser, Adrian Schauer, Mirac Fidanci, Ermler Raymond, Luka Grgic | 2022-05-20 | Weitergearbeitet |
| 1.0 | Fabian Lasser, Adrian Schauer, Mirac Fidanci, Ermler Raymond, Luka Grgic | 2022-5-24 | Beendet |

**Arbeits-Logbuch (verpflichtend auszufüllen)**

|  |  |  |
| --- | --- | --- |
| **Name** (Zuname) | **Kapitel** | **Arbeitszeit** (h) |
| Adrian Schauer | 2.2.4, 3.3, 3.4.1, 3.4.2, 4.5 | 2,5h |
| Raymond Ermler | 1.1, 1.2, 1.5, 2.6, 2.11, 3.1, 3.2, 3.6, 4.2 | 4h |
| Fabian Lasser | 1.3, 2.1, 3.5, 3.7, 4.1 | 3,5h |
| Mirac Fidanci | 2.2, 2.3, 2.5, 2.7, 2.9, 2.10, 2.12, 4.3 | 4h |
| Luka Grgic | 2.4, 2.8, 3.2, 4.4 | 2h |
| Summe |  | 16h |

# 1 Einleitung

In diesem Kapitel werden die Projektmitglieder und das Projekt genauer vorgestellt. Nicht eindeutige Begriffe werden erklärt und eine konkrete Beschreibung des Projekts abgegeben.

## 1.1 Zweck des Dokuments

Dieses Dokument dient als System-Spezifikation (Pflichtenheft) für unser Projekt „ShareList“. Dieses Dokument richtet sich ausschließlich an die Teamglieder von T03 – ShareList und an KOFLER Robert, unserem Auftraggeber.

## 1.2 Umfang

Das Projekt ShareList dient der Entwicklung einer digitalen ToDo-Liste welches folgende wesentlichen Leistungsmerkmale aufweist:

* ShareList bietet die Möglichkeit mehrere Benutzer gleichzeitig an den Listen arbeiten zu lassen
* Der Ersteller der Liste kann anderen Personen Rollen zuweisen, welche Ihnen bestimmte Rechte geben
* Man kann nicht nur eine Liste haben, sondern mehrere, was bei der Vielfalt der bestimmten Dinge nützlich ist, die man haben, will

## 1.3 Definitionen, Begriffsklärungen, Abkürzungen

|  |  |
| --- | --- |
| **Begriff** | **Beschreibung** |
| Share | teilen |
| API | Application Programming Interface – Schnittstelle für entfernten Zugriff |
| IT | Informationstechnologie |
| Performance | Mögliche Leistung der Software |
| Account | Konto |
| ToDo | “Noch zu tun” |
| Webapplikation | App im die im Internet aufrufbar ist |
| MySql | Eine Datenbank |
| Https | Protokoll für verschlüsselte Kommunikation zwischen Server und Nutzer |
| SMTP | Protokoll für eine sichere Kommunikation mit dem Mailserver |
| TCP/IP | Protokolle für sicheres arbeiten mit der Datenbank. |
| RAM | Random Access Memory |
| Provider | Anbieter |
| Libraries | Programmbibliothek |
| Endpoint | Endpunkt |
| UI (User Interface) | Eine Benutzeroberfläche |

## 1.4 Verweise

E.N. Twickler. „Entwurfsmuster für Datenbankprojekte“, 1. Auflage, ABC-Publishing Group TIML-DEDB-20140002

## 1.5 Übersicht

Im Kapitel “Systemübersicht” wird das System unseres Projektes detailliert beschrieben wie zum Beispiel diverse vorkommende Schnittstellen. Hierbei werden nur Projekt interne Elemente genauer erläutert.

Im Kapitel „Anforderungsdetails“ werden die Anforderung an dem Projekt selbst beschrieben. Außerdem werden hier auch die Funktionen des Projektes erläutert.

Im Kapitel „Test-Fälle“ werden die unterschiedlichen Tests, welche erfolgreich absolviert wurden dokumentiert.

Im Kapitel „Anhänge“ werden die im Pflichtenheft vorkommenden Fremdwörter aufgezeichnet und anschließend erklärt.

# 2 Systemübersicht

Unter diesem Kapitel wird eine Übersicht des Systems abgeben, z.B. die Software die benutzt wird und Einschränkungen welche im System vorhanden sind.

## 2.1 Produktsicht

ShareList ist ein eigenständiges Produkt und hat keinen Bestandteil eines größeren Systems.

Graphische Visualisierung:



Abbildung 1.0 Produktsicht

## 2.2 Systemschnittstellen

### 2.2.1 Benutzerschnittstellen

Der User kann seine Liste bearbeiten oder auch löschen.

### 2.2.2 Hardware-Schnittstellen

Das Projekt benötigt keine spezielle Hardware und ist eine Webapplikation und ist daher frei zugänglich im Internet.

2.2.3 Software-Schnittstellen

Im Projekt wird eine MySql Datenbank genutzt, um die Benutzerdaten zu speichern.

### 2.2.4 Kommunikationsschnittstellen

Folgende Kommunikationsschnittstellen werden hauptsächlich benutzt: Https Protokoll, um verschlüsselte Kommunikation zwischen Server und Nutzer bieten zu können. SMTP Protokoll, um sicheres kommunizieren mit dem Mail Server zu gewährleisten. TCP/IP Protokolle, für sicheres arbeiten mit der Datenbank.

## 2.3 Speichereinschränkungen

Die Datenbank kann eventuell volllaufen mit Daten.

## 2.4 Betrieb

Ein Benutzer der Seite muss registriert sein mit einem Passwort und einer gültigen E-Mail-Adresse. ShareList ist eine simple Seite und so gut wie jeder PC kann sie zum Laufen bringen, daher gibt es keine Beschränkungen in Hinsicht auf RAM oder sonstigem Speicher. Allerdings wird eine Internetverbindung benötigt.

## 2.5 Standortabhängige Anforderungen

Ein erstelltes Konto wird benötigt, um auf die Software zugreifen zu können.

## 2.6 Benutzercharakteristik

Als Zielgruppe dieses Projekts sind Personen, jeglichen Alters, Bildungsgrades, Erfahrungslevels oder technischen Wissens. Die Eigenschaften sind nicht relevant für die Festlegung bestimmter Systemanforderungen.

## 2.7 Zusammenfassung der Produktfunktionen

Benutzer können Listen erstellen, bearbeiten und löschen, außerdem können andere Nutzer bei Listen mitwirken.

## 2.8 Anforderungen an die Dokumentation

Für die Dokumentation des Projekts haben wir mehrere Dokumente wie z.B. Das Lastenheft, welches eine PDF-Datei ist für die Zusammenstellung der Forderungen, welche benötigt werden für die korrekte Vollendung unseres Projekts. Auch der Statusbericht trägt zu einer gründlichen Dokumentation bei. Der Statusbericht ist ebenfalls eine PDF-Datei. Ein Benutzerhandbuch haben wir nicht, da wir eine Website haben mit einer simplen Benutzung und kaum Komplexitäten, weil wir die Seite so gestaltet haben, dass die Verwendung klar und offensichtlich ist.

## 2.9 Einschränkungen

Die Website ist auf Englisch und andere Nutzer können nicht ohne Erlaubnis auf andere Listen zugreifen.

## 2.10 Lieferumfang

Da das Projekt eine Website ist benötigen wir einen Provider und um einen Nutzer zu erstellen wird eine E-Mail versendet mit dem Verifzierungsschlüssel.

## 2.11 Annahmen und Abhängigkeiten

Die Benutzung anderer Libraries könnte zur Verkomplizierung der Systemfunktionen führen, da diese womöglich mit den bestechenden Libraries nicht zusammenarbeiten können. Auch wäre die Erlernbarkeit dieser neuen Technologien ein großes Problem, da wir schon mit anderen Technologien vertraut sind, welche anders funktionieren und es dadurch zu Verwechslungen kommen könnte.

## 2.12 Anforderungsaufteilung

Falls der Arbeitgeber weiter Wünsche nach der Vollendung der Seite hat, werden diese dementsprechend impliziert und in eine neuere Version eingebunden.

# 3 Anforderungsdetails

Im Kapitel “Anforderungsdetails” werden die Details welche in Kapitel 2 angesprochen. wurden genauer beschrieben.

## 3.1 Externe Schnittstellen

REST Schnittstelle:  
Port 127.0.0.1:8000  
  
Webschnittstelle:  
Port 127.0.0.1:8000  
Aufbereitetes Web UI für den Benutzer der Seite

## 3.2 Detaillierte Funktionsbeschreibung

### 3.2.1 Funktion /01/

Navbar im Frontend, welche dem Benutzer ermöglicht, leicht die Webapp zu navigieren. Mit folgenden links: Logo (/), Home (/), About Us (/about), Register (/register).

Abbildung 2.0 Navbar

### 3.2.2 Funktion /02/

(/)

Auf dieser Seite befindet sich die Homepage und beinhaltet Informationen wieso man diese Webapp benutzen sollte und auch wie.



Abbildung 3.0 Homepage Informationen

### 3.2.3 Funktion /03/

Footer im Frontend, welche dem Benutzer ermöglicht, leicht die Webapp zu navigieren. Mit folgenden links: About Us (/about) und Contact, welche einem zulässt eine Mail an unseren Benutzerservice zu schicken.



Abbildung 4.0 Footer

### 3.2.4 Funktion /04/

Bei Register kann man sich ein Account auf unserer Webseite erstellen. Dafür muss man eine E-Mail und ein Passwort eingeben. Auch kann man von hier aus zu Login (/login) gelangen, wenn man bereits ein Account auf der Webseite besitzt.

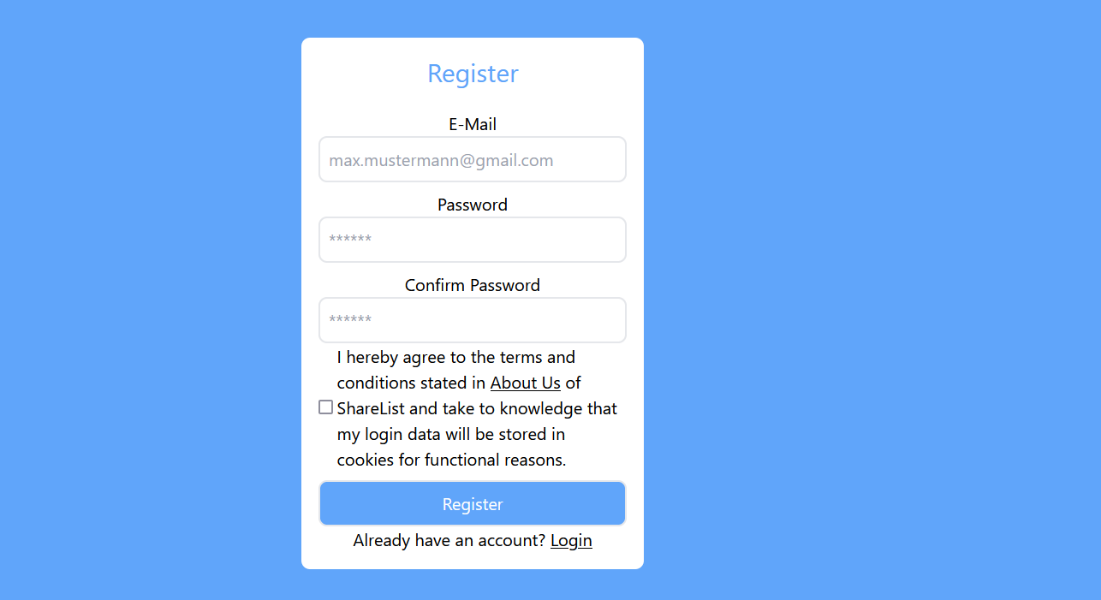


Abbildung 5.0 Register

### 3.2.5 Funktion /05/

Bei Login kann man sich in einen bereits erstellten Account sich einloggen. Auch kann man von hier aus zu Register (/register) kommen.

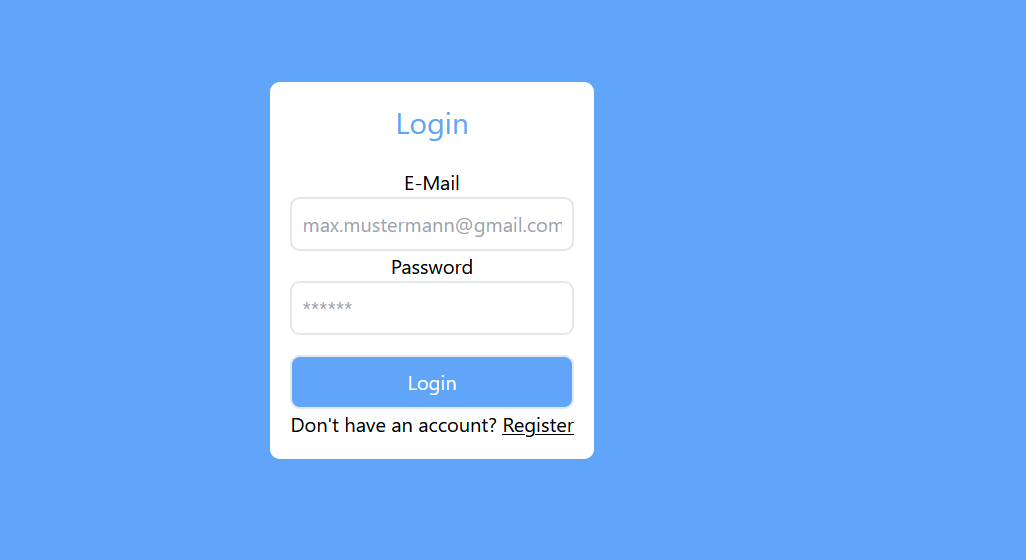


Abbildung 6.0 Login

### 3.2.6 Funktion /06/

Bei About Us befindet sich unser Impressum und andere wichtige Informationen für den Benutzer. Außerdem kommt man von hier aus zu unserer Datenschutzerklärung (/privacypolicy).

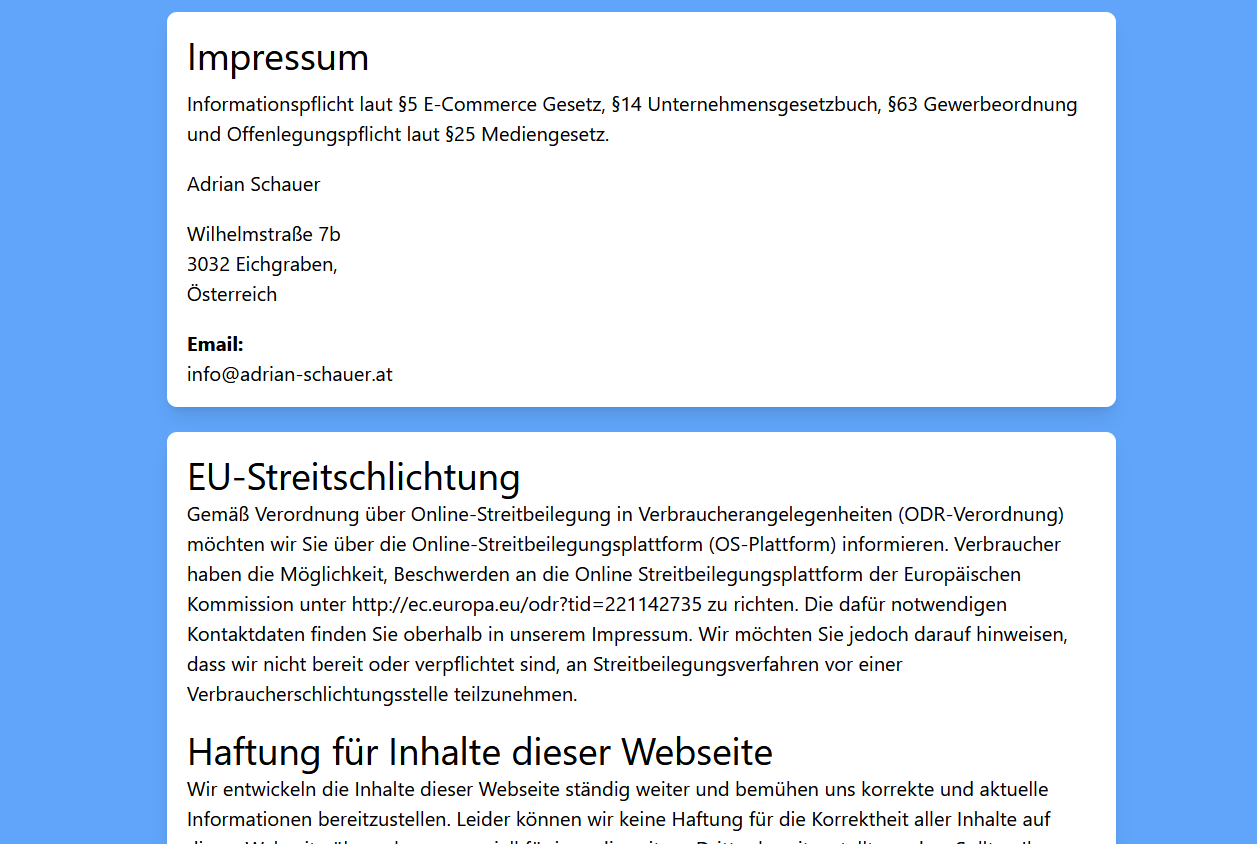


Abbildung 7.0 About Us

### 3.2.7 Funktion /07/

Auf der Seite Datenschutz befindet sich unsere AGB.

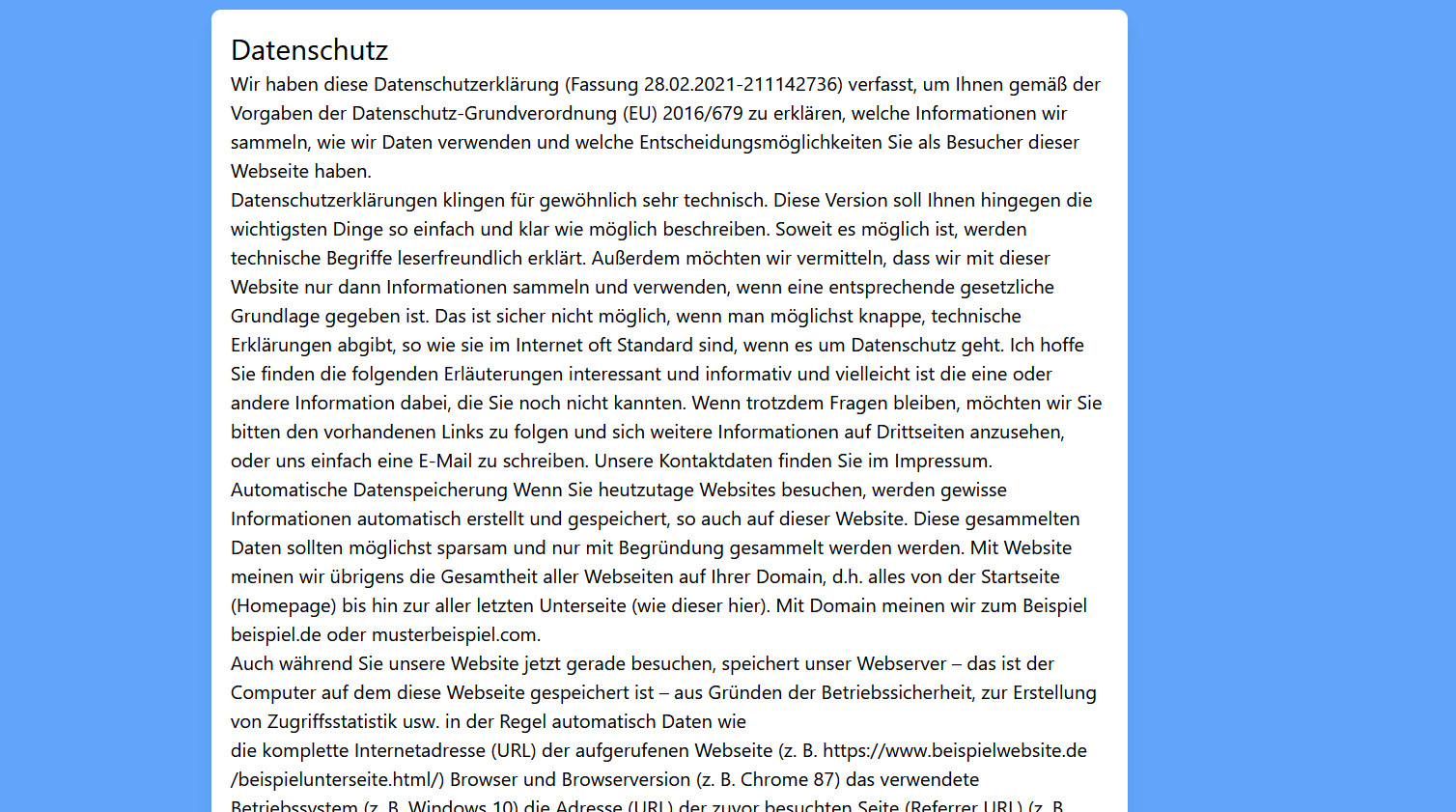


Abbildung 8.0 Datenschutz

### 3.2.8 Funktion /08/

Auf der Seite Lists befinden sich alle Listen auf die, der Benutzer Zugriff hat. Außerdem kann man hier auch neue Listen erstellen.

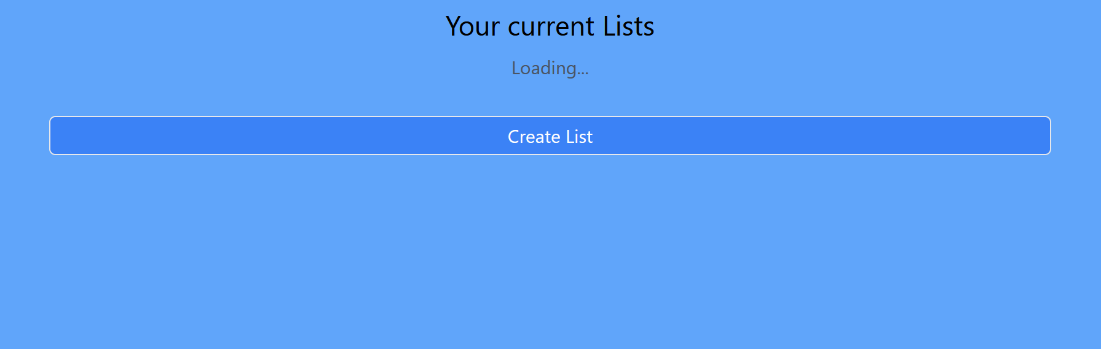


Abbildung 9.0 Listen

### 3.2.9 Funktion /09/

Auf der Seite Profile (/profile) können die Account Details geändert werden. Auch kann der Account gelöscht werden nachdem man im Account eingeloggt ist.

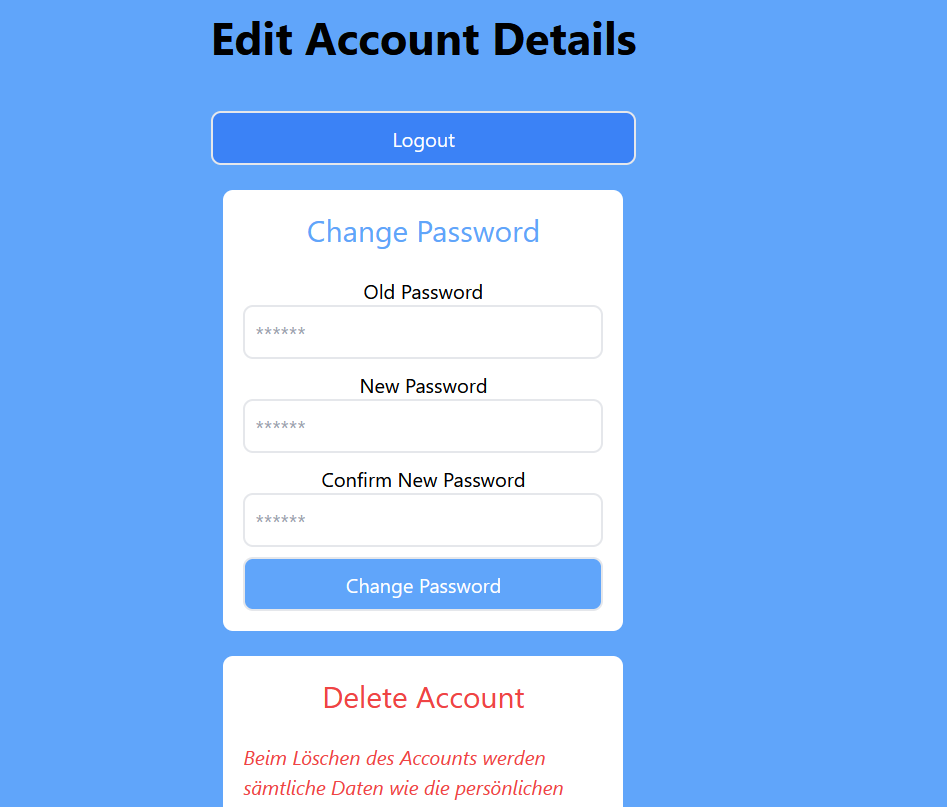


Abbildung 10.0 Profile

**3.2.10 Funktion /**

Auf der Seite “Invite” kann man andere Benutzer zu seiner Liste hinzufügen.

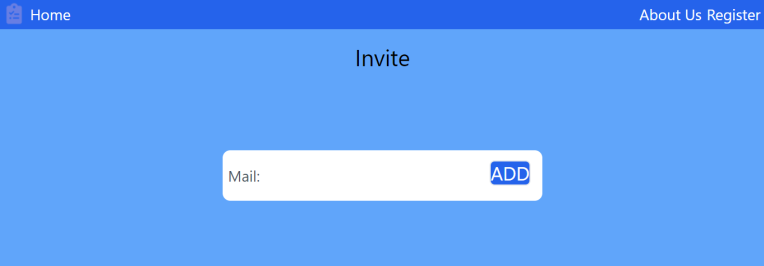


Abbildung 11.0 Invite

**3.2.11 Funktion /**

Bei “List Edit” kann man seiner Liste einen Namen und eine Beschreibung geben. Auch gibt es eine “TransferList”, wo der Benutzer der Liste geändert wird. Um die Liste zu löschen muss man sein Passwort eingeben und schließlich den “Delete” Knopf drücken.

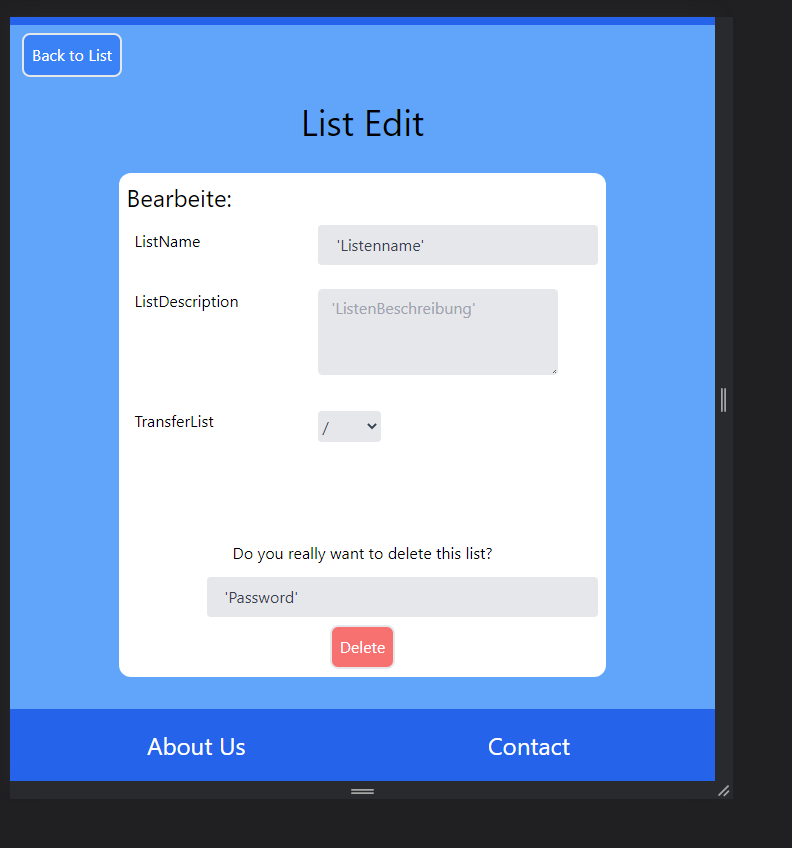


Abbildung 12.0 Liste editieren

**3.2.12 Funktion /**

Wir haben ebenso eine Seite für die jetzigen Listen welche benutzt werden. Auf dieser Seite kann man eine Liste mit dem “Delete” Knopf löschen und dem “Show“ Knopf kann man sich eine Liste anschauen

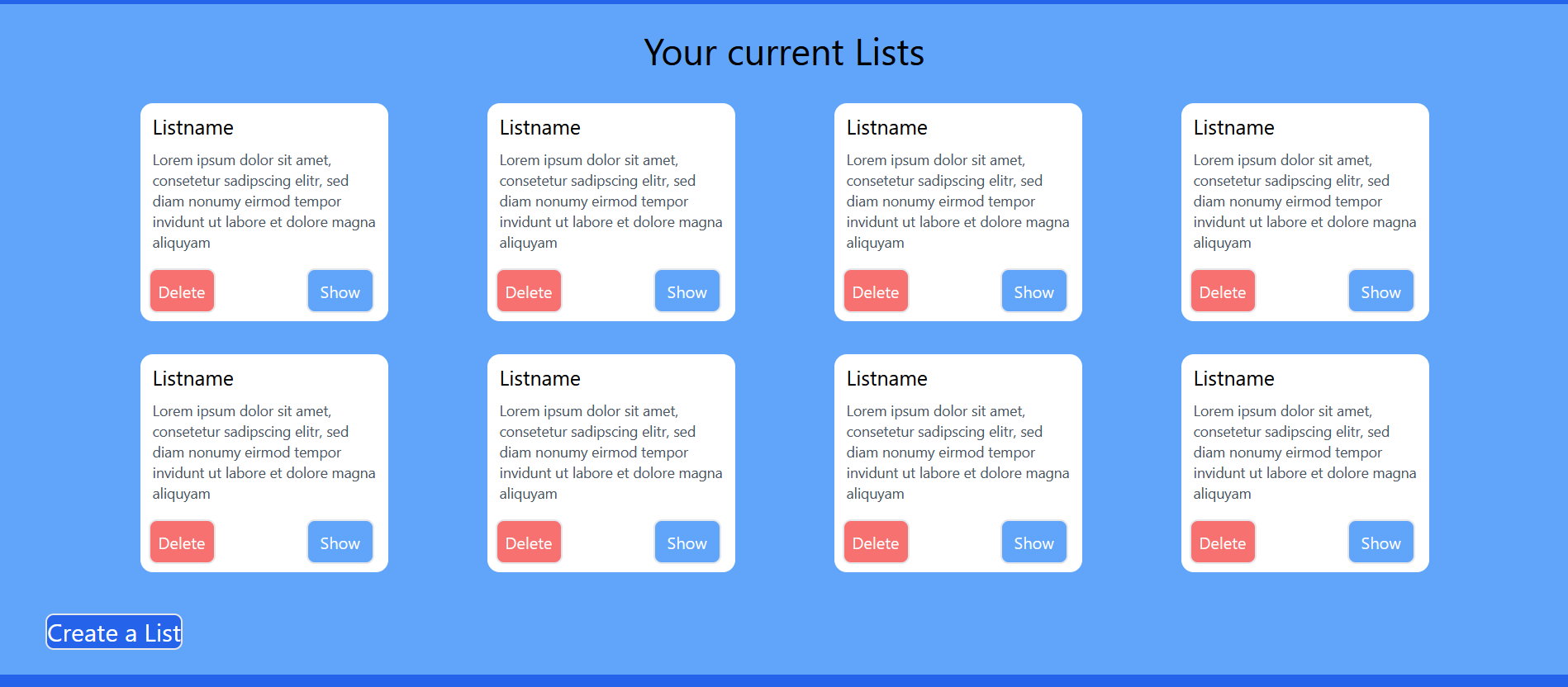


Abbildung 13.0 “Your current Lists”

## 3.3 Klassen und Module

+--------+----------+---------------------------------------+------+------------------------------------------------------------+------------+

| Domain | Method | URI | Name | Action | Middleware |

+--------+----------+---------------------------------------+------+------------------------------------------------------------+------------+

Dieses Modul gibt die gefragte Liste zurück.

| | POST | api/list | | Closure | api |

Dieses Modul erstellt eine Liste

| | POST | api/list/create | | Closure | api |

Dieses Modul löscht eine Liste

| | POST | api/list/delete | | Closure | api |

Dieses Modul schickt ein Invite weg

| | POST | api/list/invite | | Closure | api |

Dieses Modul akzeptiert ein bestimmen Invite

| | POST | api/list/invite/accept | | Closure | api |

Dieses Modul löscht ein bestimmtes Invite

| | POST | api/list/invite/delete | | Closure | api |

Dieses Modul gibt alle offenen Invites zurück

| | POST | api/list/invites | | Closure | api |

Dieses Modul fügt ein Item einer Liste hinzu

| | POST | api/list/item/add | | Closure | api |

Dieses Modul setzt ein Item auf checked

| | POST | api/list/item/check | | Closure | api |

Dieses Modul löscht ein bestimmtes Item

| | POST | api/list/item/delete | | Closure | api |

Dieses Modul updated ein bestimmtes Item

| | POST | api/list/item/update | | Closure | api |

Dieses Modul entfernt ein Mitglied von einer Liste

| | POST | api/list/member/remove | | Closure | api |

Dieses Modul gibt/entfernt einem Mitglied write access

| | POST | api/list/member/write | | Closure | api |

Dieses Modul gibt alle Mitglieder zurück

| | POST | api/list/members | | Closure | api |

Dieses Modul benennt eine bestimmte Liste um

| | POST | api/list/rename | | Closure | api |

Dieses Modul transferiert eine Liste zu einem anderen Mitglied

| | POST | api/list/transfer | | Closure | api |

Dieses Modul gibt alle Listen zurück

| | POST | api/lists | | Closure | api |

Dieses Modul ändert das Passwort eines Nutzers

| | POST | api/user/changepwd | | Closure | api |

Dieses Modul gibt zurück ob der Nutzer bereits verifiziert ist

| | POST | api/user/check | | Closure | api |

Dieses Modul überprüft ob die E-Mail bereits benutzt wird

| | POST | api/user/checkemail | | Closure | api |

Dieses Modul löscht einen Nutzer

| | POST | api/user/delete | | Closure | api |

Dieses Modul logged einen Nutzer ein und gibt einen Token für den Aufenthalt zurück

| | POST | api/user/login | | Closure | api |

Dieses Modul registriert einen neuen Nutzer

| | POST | api/user/register | | Closure | api |

Dieses Modul verifiziert einen Nutzer

| | POST | api/user/verify | | Closure | api |

Dieses Modul gibt das angefragte Bild zurück

| | GET|HEAD | images/{filename} | | Closure | web |

Diese Module geben alle die Single Page Application zurück

| | GET|HEAD | {route?} | | Closure | web |

| | GET|HEAD | {route?}/{secondRoute?} | | Closure | web |

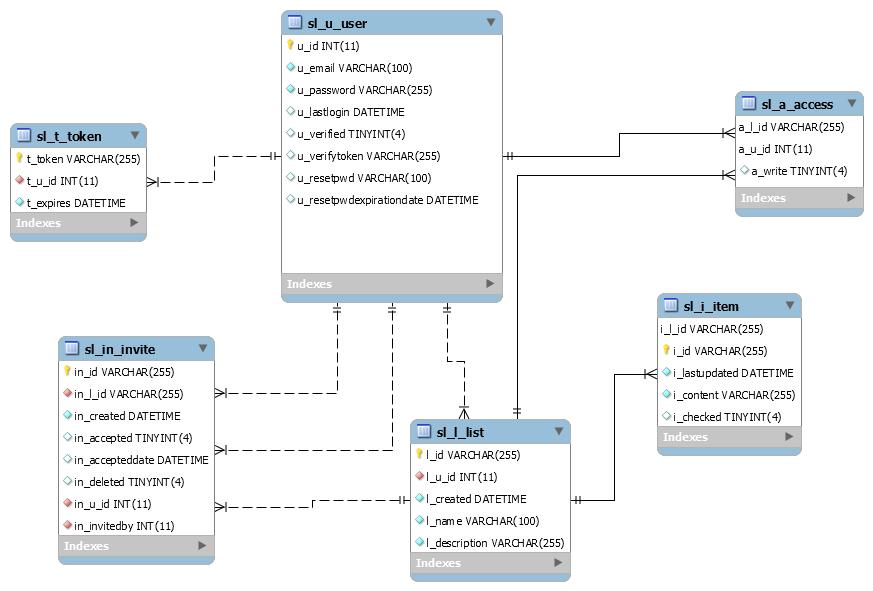
| | GET|HEAD | {route?}/{secondRoute?}/{thirdRoute?} | | Closure | web |

+--------+----------+---------------------------------------+------+------------------------------------------------------------+------------+

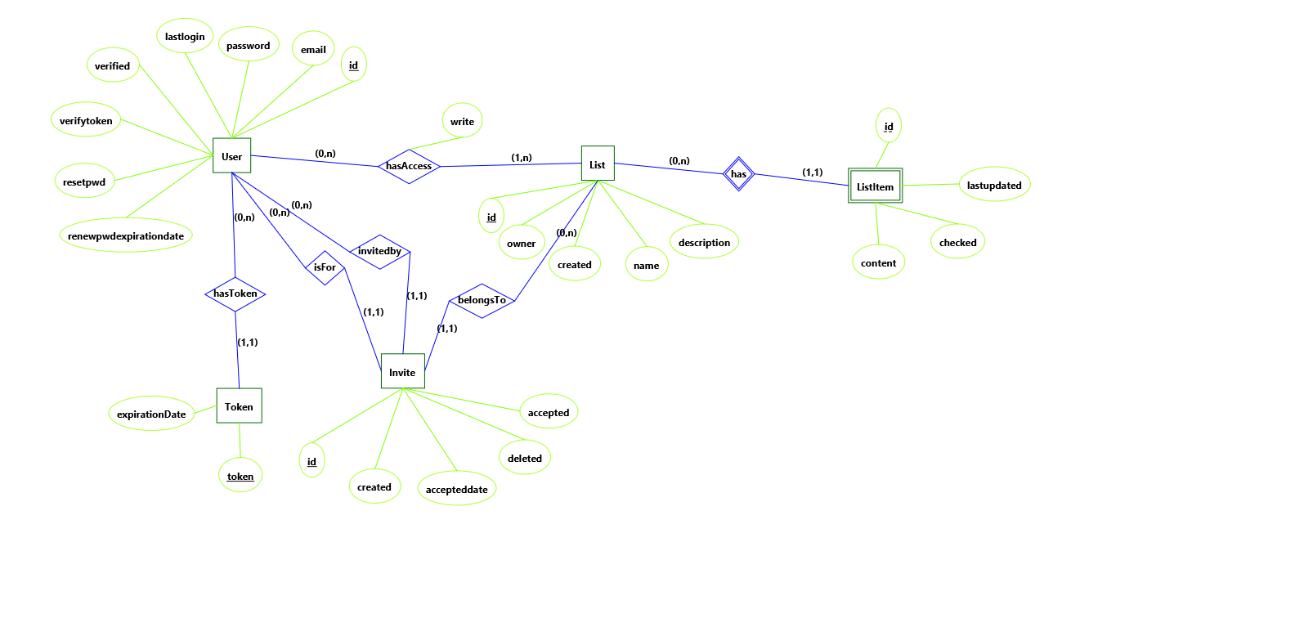
## 3.4 Datenbank Anforderungen

### 3.4.1 Datenmodell und Dateien

Für das Backend liegen keine Konfigurationsdateien vor. Das Backend ist ausschließlich über Umgebungsvariablen zu konfigurieren. Als Datenbank wird nur MySQL verwendet.

Abbildung 14.0 Relationales-Model

### 3.4.2 Entity-Relationship-Model

Abbildung 15.0 ERM-Model

### 3.4.3 Tabellendefinitionen

sl\_a\_access(

a\_l\_id varchar(255) not null,

a\_u\_id int not null,

a\_write tinyint,

primary key(a\_l\_id,a\_u\_id)

);

sl\_l\_list(

l\_id varchar(255) not null,

l\_created datetime not null,

l\_name varchar(100) not null,

l\_description varchar(255) not null default '',

l\_u\_id int not null,

primary key(l\_id)

);

sl\_i\_item(

i\_id varchar(255) not null,

i\_l\_id varchar(255) not null,

i\_lastupdated datetime not null,

i\_content varchar(255) not null,

i\_checked tinyint default false,

primary key(i\_id,i\_l\_id)

);

sl\_t\_token(

t\_u\_id int not null,

t\_token varchar(255) not null,

t\_expires datetime not null,

primary key (t\_token)

);

sl\_u\_user(

u\_id int primary key AUTO\_INCREMENT,

u\_email varchar(100) not null,

u\_password varchar(255) not null,

u\_lastlogin datetime,

u\_verified tinyint default 0,

u\_verifytoken varchar(255),

u\_resetpwd varchar(100),

u\_resetpwdexpirationdate datetime

);

sl\_in\_invite(

in\_id varchar(255) not null,

in\_l\_id varchar(255) not null,

in\_u\_id int not null,

in\_invitedby int not null,

in\_created datetime not null,

in\_accepted tinyint default 0,

in\_accepteddate datetime,

in\_deleted tinyint default 0,

primary key(in\_id)

);

## 3.5 Performanzanforderungen

Bis zu 1000 Personen können gleichzeitig unsere Website benutzen, um Listen zu erstellen und bearbeiten. Wenn mehr Personen gleichzeitig die Seite benutzen würden, würde es zu Performanceeinbrüchen kommen (Längere Ladezeiten, Serverabsturz, …,).

## 3.6 Entwurfseinschränkungen

Es gibt keine speziellen Hardwareeinschränkungen, wie in 2.2.2 zuvor spezifiziert wurde. Auch braucht man kein spezielles Betriebssystem, um auf unsere Webseite zugreifen zu können. Für unser Projekt gab es keine strengen Namenskonventionen Variablen- und Dateinamen.

## 3.7 Qualitätsanforderungen, Systemcharakteristika

Die meisten HTTP Endpoints benötigen eine Nutzer Authentifizierung mit Nutzername oder E-Mail, und Password. Dies ist nötig, um die Sicherheit des Servers – insbesondere die der Datenbank – zu gewährleisten.

# 4 Testfälle

## 4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID / Bezeichnung | T01 Erstellen einer Liste | | | |
| Beschreibung | Auf der Website den Knopf “Create List” um eine Liste zu erstellen, die er später dann befüllen kann. | | | |
| Testvoraussetzung | User muss einen bei Account ShareList besitzen. | | | |
| Input | User muss den “Create List” Knopf drücken. | | | |
| Vorgehensweise / Testschritte | User geht auf die Lists Seite und drückt dort den “Create List” Knopf. Danach kann er Daten in die Liste eingeben. | | | |
| Erwartetes Ergebnis | Die Liste ist auf der Website einsehbar. | | | |
| Beobachtetes Ergebnis | Die Liste wird angezeigt und in der Datenbank hinzugefügt. | | | |
| Tester / Datum / Fehlerklasse | Fabian Lasser | 24.05.2022 | Keine Fehler |

## 4.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID / Bezeichnung | T02 Löschen einer Liste | | | |
| Beschreibung | User löscht eine Liste, welche er/sie erstellt hat. | | | |
| Testvoraussetzung | Man muss eingeloggt sein und die Liste erstellt haben. | | | |
| Input | Man muss auf den Knopf, auf dem Löschen steht, drücken. | | | |
| Vorgehensweise / Testschritte | Man loggt sich zuerst ein und navigiert zur erstellten Liste, welche gelöscht werden soll. Dann drückt man auf “Löschen” und die Liste wird entfernt. | | | |
| Erwartetes Ergebnis | Die Liste wird für alle User, welche Zugriff auf diese hatten, gelöscht und die Liste wird aus der zentralen Datenbank entfernt mit allen enthaltenen Daten. | | | |
| Beobachtetes Ergebnis | Liste wurde erfolgreich gelöscht. | | | |
| Tester / Datum / Fehlerklasse | Raymond Ermler | 24.05.2022 | Keine Fehler |

## 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID / Bezeichnung | T03 Passwort ändern | | | |
| Beschreibung | Der User ändert sein Passwort | | | |
| Testvoraussetzung | Muss eingeloggt sein und altes Passwort muss bekannt sein | | | |
| Input | E-Mail und Passwort | | | |
| Vorgehensweise / Testschritte | Der User ruft seinen Account auf und drückt den Knopf mit der Aufschrift Passwort ändern. Das alte Passwort muss eingeben werden und dann das Neue | | | |
| Erwartetes Ergebnis | Geändertes Passwort | | | |
| Beobachtetes Ergebnis | User konnte sein Passwort ändern | | | |
| Tester / Datum / Fehlerklasse | Mirac Fidanci | 23.05.2022 | Keine Fehler |

## 4.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID / Bezeichnung | T04 Löschen eines Accounts | | | |
| Beschreibung | Der Nutzer löscht einen Account. | | | |
| Testvoraussetzung | Der Account der gelöscht werden will muss existieren | | | |
| Input | E-Mail und Passwort | | | |
| Vorgehensweise / Testschritte | Auf der “Edit Account” Seite unter “Delete Account” den “Delete” Knopf drücken. Allerdings müssen E-Mail und Passwort bestätigt werden. | | | |
| Erwartetes Ergebnis | Der Account wird gelöscht. | | | |
| Beobachtetes Ergebnis | Accounts können problemlos gelöscht werden. | | | |
| Tester / Datum / Fehlerklasse | Luka Grgic | 24.05.2022 | Keine Fehler |

## 4.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID / Bezeichnung | T05 Account erstellen | | | |
| Beschreibung | Nutzer erstellt ich einen Account auf der Website | | | |
| Testvoraussetzung | Die E-Mail zur Account Erstellung ist noch nicht in Benutzung und ist nicht angemeldet. | | | |
| Input | E-Mail und Password | | | |
| Vorgehensweise / Testschritte | Der Nutzer navigiert zu Registrierung und gibt seine Daten an. Diese werden darauf validiert, wenn er auf “Registrieren” drückt. E-Mail darf noch nicht benutzt sein und das Passwort muss zwischen 6 und 20 Zeichen liegen. Nutzer kriegt eine Mail mit einem Bestätigungslink. Diesen muss der Nutzer öffnen und daraufhin wird er automatisch verifiziert. Das Fenster mit der Registrierung wird automatisch angemeldet und auf die “Lists” Seite weitergeleitet. | | | |
| Erwartetes Ergebnis | Nutzer wird einen Account erstellt und wird automatisch angemeldet. | | | |
| Beobachtetes Ergebnis | Account wurde erstellt und Nutzer wird automatisch angemeldet. | | | |
| Tester / Datum / Fehlerklasse | Adrian Schauer | 24.05.2022 | Kein Fehler |