Softwareprojekt

Projektarbeit - technische Spezifikation

MindCollector

Fachhochschule Bielefeld

Campus Minden

Studiengang Informatik

|  |  |
| --- | --- |
|  | Beteiligte Personen: |
|  |  |
| **Name** | **Matrikelnummer** |
| Fabian Wagenfeld | 1143629 |
|  |  |
| Marko Lemac | 1141203 |
| Leon Rudolph | 1225431 |

Inhaltsverzeichnis

|  |  |  |  |
| --- | --- | --- | --- |
| [1](#page3) | [Einführung](#page3) | | 3 |
|  | [1.1](#page3) | [Beschreibung](#page3) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3 |
|  | [1.2](#page3) | [Ziele](#page3) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3 |
| [2](#page4) | [Anforderungen](#page4) | | 4 |
|  | [2.1](#page4) | [Stakeholder](#page4) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
|  | [2.2](#page5) | [Funktionale Anforderungen](#page5) . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
|  | [2.3](#page8) | [Nicht-funktionale Anforderungen](#page8) . . . . . . . . . . . . . . . . . . . . . . . | 8 |
|  |  | [Qualitätsmerkmale](#page9) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |
|  | [2.4](#page10) | [Graphische Benutzerschnittstelle](#page10) . . . . . . . . . . . . . . . . . . . . . . . | 10 |
|  | [2.5](#page22) | [Anforderungen im Detail](#page22) . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 22 |
| [3](#page24) | [Technische Beschreibung](#page24) | | 24 |
|  | [3.1](#page24) | [Systemübersicht](#page24) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 24 |
|  | [3.2](#page25) | [Softwarearchitektur](#page25) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 25 |
|  |  | [VUE](#page25)....................................... | 25 |
|  |  | [VUEX](#page26)...................................... | 26 |
|  |  | [Axios](#page26) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 26 |
|  |  | [Django](#page26) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 26 |
|  |  | [EasyOCR](#page27) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |
|  |  | [SpeechRecognication](#page27) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |
|  |  | [MySQL](#page27) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |
|  | [3.3](#page27) | [Schnittstellen](#page27) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 27 |
|  |  | [Ereignisse](#page28) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 28 |
|  | [3.4](#page29) | [Datenmodell](#page29) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 29 |
|  | [3.5](#page33) | [Abläufe](#page33) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 33 |
|  | [3.6](#page34) | [Entwurf](#page34) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 34 |
| [4](#page35) | [Projektorganisation](#page35) | | 35 |
|  | [4.1](#page35) | [Annahmen](#page35) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
|  |  | [Verwendete Technologien](#page35) . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
|  |  | [Erweiterbarkeit](#page35) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
|  | [4.2](#page35) | [Verantwortlichkeiten](#page35) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
|  |  | [Softwarebausteine](#page35) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 35 |
|  | [4.3](#page36) | [Grober Projektplan](#page36) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 36 |
| [Literaturverzeichnis](#page37) | | | 37 |

2

1 Einführung

1.1 Beschreibung

Spontane Ideen, gefundene Links und Zitate, mündliche Mitteilungen gehen schnell verloren bzw. können schnell vergessen werden. Oder man trägt diese Dinge in eine App ein und stehen dann nicht auf allen Geräten zur Verfügung. Es soll eine responsive Webseite erstellt werden. Die Ideen werden als eine Art „Schnipsel“ angezeigt werden. Man soll ich Registrieren und Anmelden können. Jeder Benutzer hat Zugriﬀ auf seine eigenen Sammlungen. Kann diese mit anderem Benutzer teilen. Hinzufügen, löschen und editieren von den einzelnen Schnipseln soll möglich sein. Es gibt eine Übersicht über seine eigenen Ideen und den geteilten. Auch Bilder sollen gespeichert werden können.

1.2 Ziele

Mit MindCollector kann man seine Ideen, neuen Erfahrungen bei der Arbeit, nützliche Links erfassen und mit Hilfe eines Webbrowser auf Desktop und Smartphone abrufen. Auch soll es möglich sein, Sprachaufnahmen aufzunehmen. Diese Aufnahme wird dann analysiert und in Text umgewandelt. Macht man ein Bild von Text, soll dieser Text ebenfalls analysiert werden und in Text umgewandelt werden. Man kann vorher auch entscheiden, ob man das Bild umwandeln möchte.

Die Nutzung dieser App ist selbsterklärend, man brauch keinerlei Vorerfahrung.

3

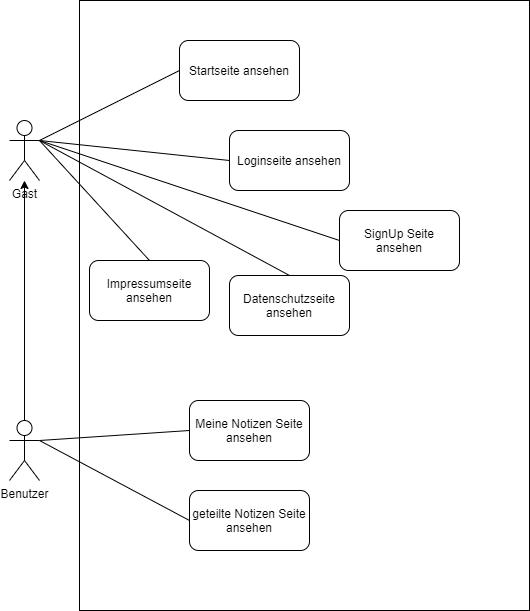
2 Anforderungen

2.1 Stakeholder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Funktion / Re-** | **Name** | **Kontakt / Ver-** | **Wissen** | **Interessen /** |
| **levanz** |  | **fügbarkeit** |  | **Ziele** |
| Backend Devel- | Fabian Wagen- | Per E-Mail, im- | Datenbank- | sichere Daten- |
| oper | feld | mer erreichbar, | Server, Webser- | speicherung |
|  |  | Verfügbarkeit | ver |  |
|  |  | 75% |  |  |
|  |  |  |  |  |
| Frontend Devel- | Marko Lemac | Per E-Mail, im- | Gestaltung Web- | Benutzerfreund- |
| oper |  | mer erreichbar, | seite für Desktop | lichkeit, einfache |
|  |  | Verfügbarkeit | und Mobile Gerä- | und moderne |
|  |  | 75% | te | Gestaltung |
| Fullstack Devel- | Leon Rudolph | Per E-Mail, im- | Audio und Bild | gute Bild- und |
| oper |  | mer erreichbar, | zu Text Analy- | Audioanalyse |
|  |  | Verfügbarkeit | se, Back- und |  |
|  |  | 75% | Frontend |  |

4

2.2 Funktionale Anforderungen

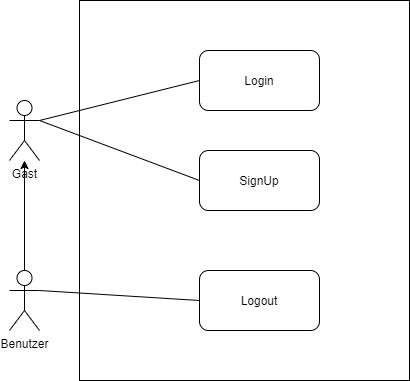


**Abbildung 1:** Ansicht UseCase

Als Gast kann ich mir die Start-, Login-, SignUp-, Datenschutz- und Impressumsseite ansehen.

Wenn ich mich eingeloggt habe, kann ich meine eigenen Notizen und die geteilten Notizen zusätzlich noch ansehen.

5

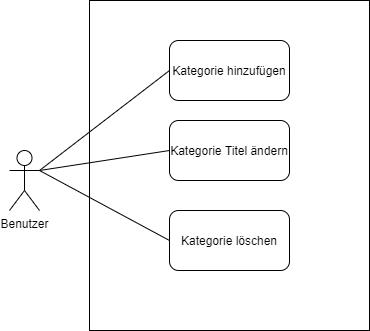


**Abbildung 2:** Authentifizierung UseCase

Ein Gast kann sich anmelden und registrieren.

Ein Benutzer kann sich zusätzlich noch ausloggen.

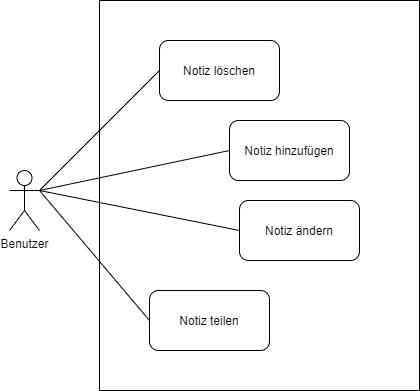
6



**Abbildung 3:** Kategorie UseCase

Der Benutzer kann Kategorien für die Notizen erstellen, ändern und wieder löschen.

7



**Abbildung 4:** Notizen UseCase

Der Benutzer kann eine Notiz hinzufügen, ändern, löschen und teilen.

Eine Notiz kann unterschiedlich sein. Entweder ein reiner Text oder ein Bild oder eine Audioaufnahme. Nur der reine Text kann geändert werden. Änderungen beim Bild oder bei der Audiodatei sollen nicht möglich sein.

2.3 Nicht-funktionale Anforderungen

Die Funktion zum teilen von Notizen wird in diesem Projekt nebensächlich sein und vorerst nicht implementiert.

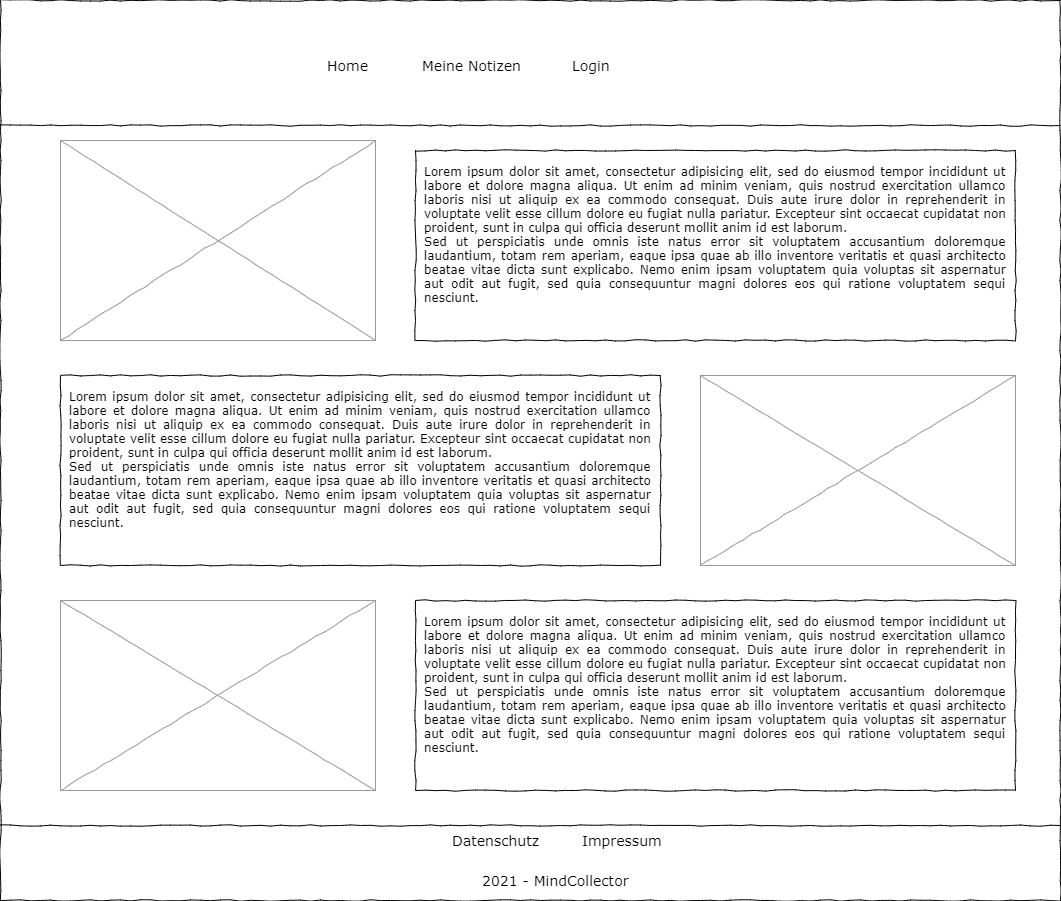
8

Qualitätsmerkmale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Qualitätsmerk-** | **sehr gut** | **gut** | **normal** | **nicht relevant** |
| **mal** |  |  |  |  |
| **Zuverlässigkeit** |  |  |  |  |
|  |  |  |  |  |
| Fehlertoleranz | X | - | - | - |
| Wiederherstellbar- | X | - |  | - |
| keit |  |  |  |  |
|  |  |  |  |  |
| Ordnungsmäßig- | X | - | - | - |
| keit |  |  |  |  |
| Richtigkeit | X | - | - | - |
|  |  |  |  |  |
| Konformität | - | X | - | - |
| **Benutzerfreund-** |  |  |  |  |
| **lichkeit** |  |  |  |  |
|  |  |  |  |  |
| Installierbarkeit | - | - | - | X |
| Verständlichkeit | X | - | - | - |
|  |  |  |  |  |
| Erlernbarkeit | - | X | - | - |
| Bedienbarkeit | - | X | - | - |
|  |  |  |  |  |
| **Performance** |  |  |  |  |
| Zeitverhalten | - | X | - | - |
|  |  |  |  |  |
| Eﬃzienz | - | - | X | - |
| **Sicherheit** |  |  |  |  |
|  |  |  |  |  |
| Analysierbarkeit | - | - | - | X |
| Modifizierbarkeit | - | - | - | X |
|  |  |  |  |  |
| Stabilität | - | X | - | - |
| Prüfbarkeit | - | X |  | - |

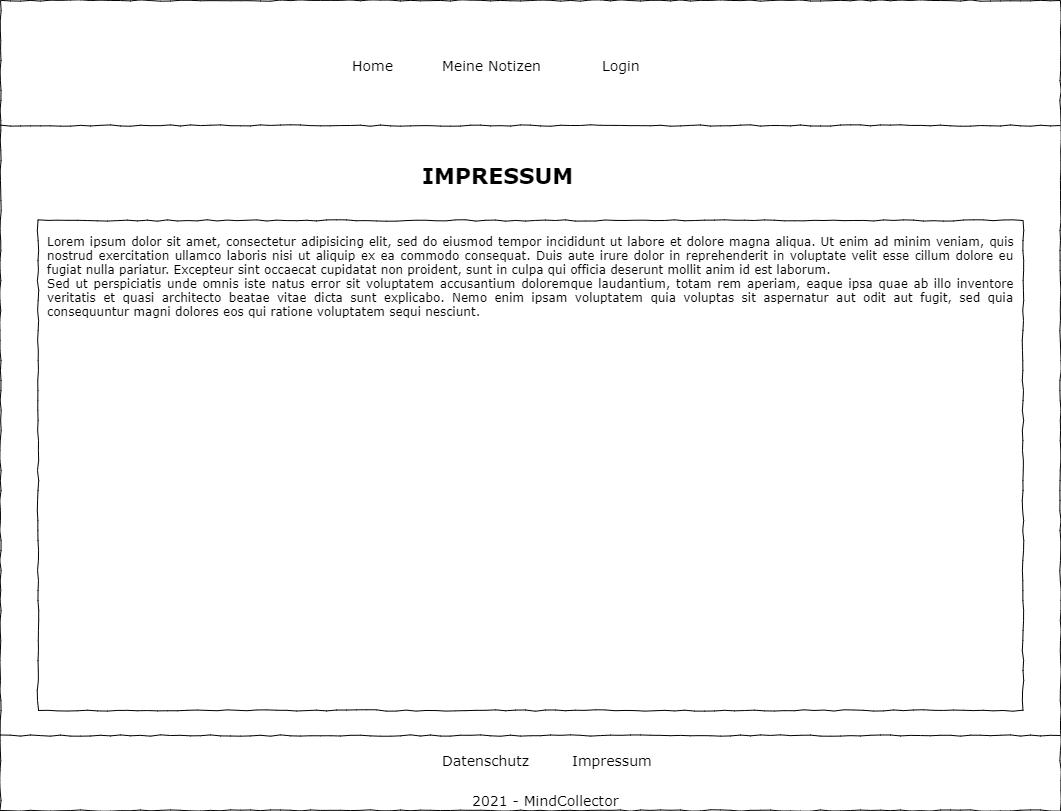
9

2.4 Graphische Benutzerschnittstelle



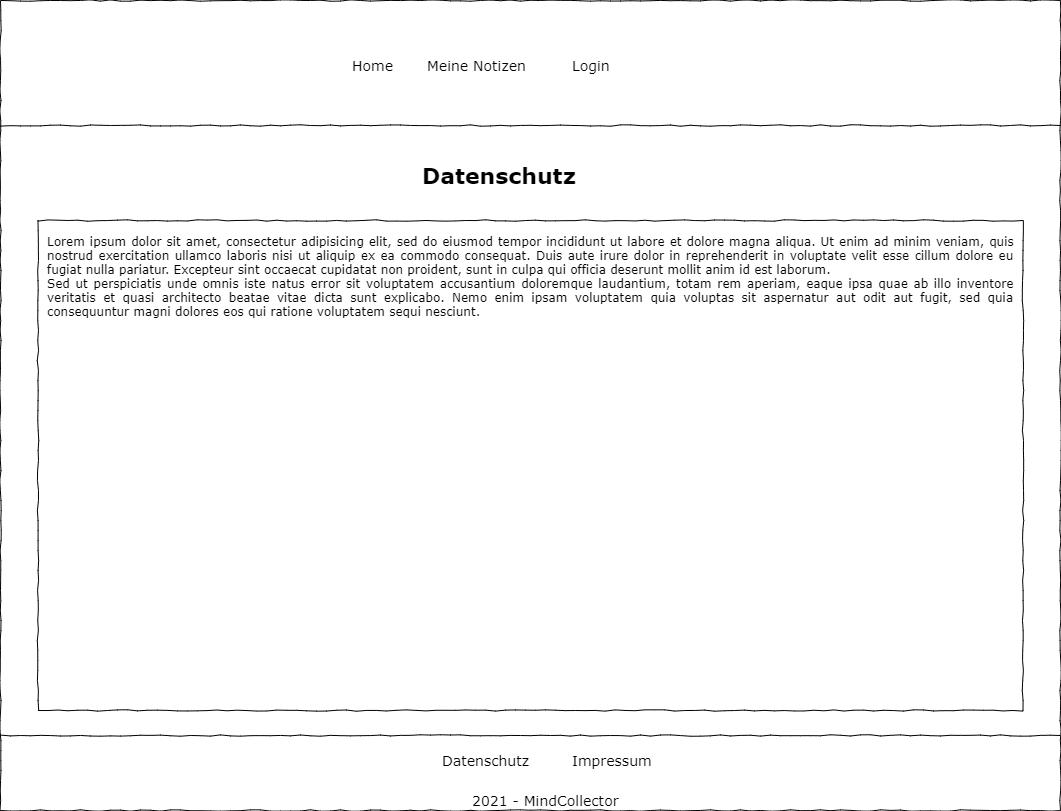
**Abbildung 5:** Startseite

10



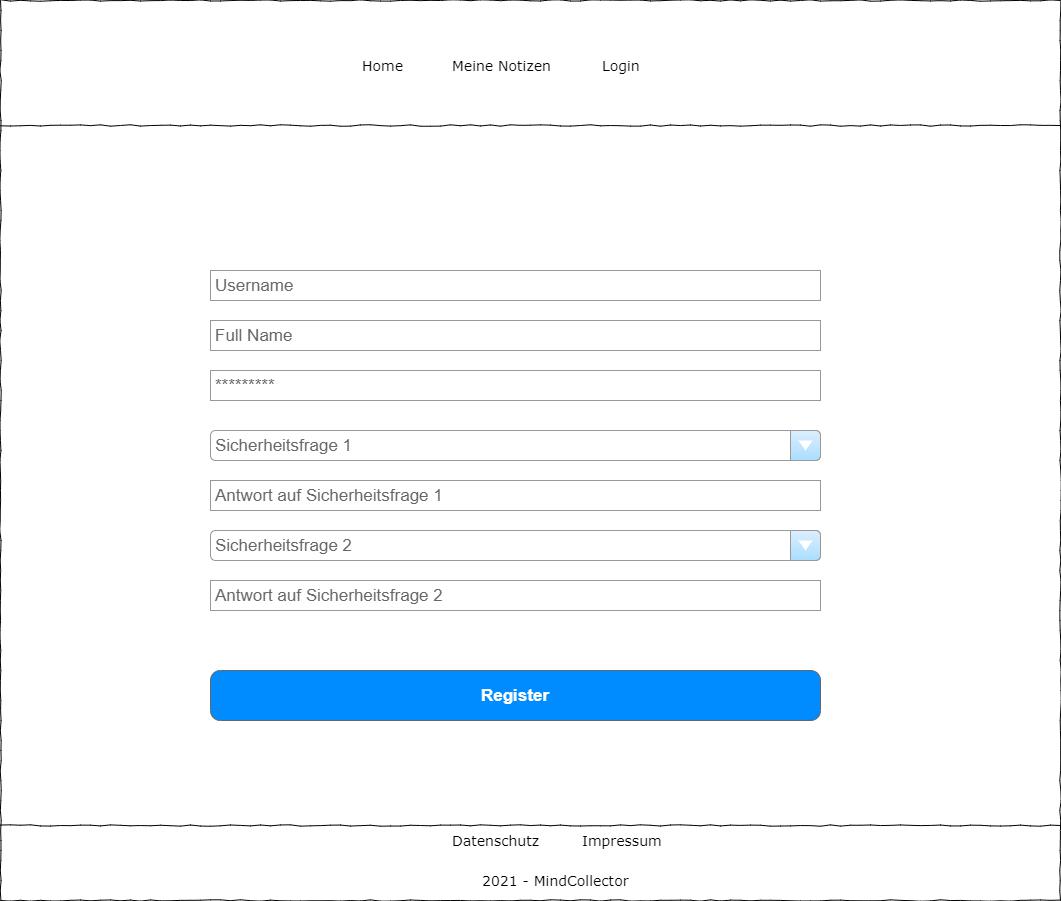
**Abbildung 6:** Impressum

11



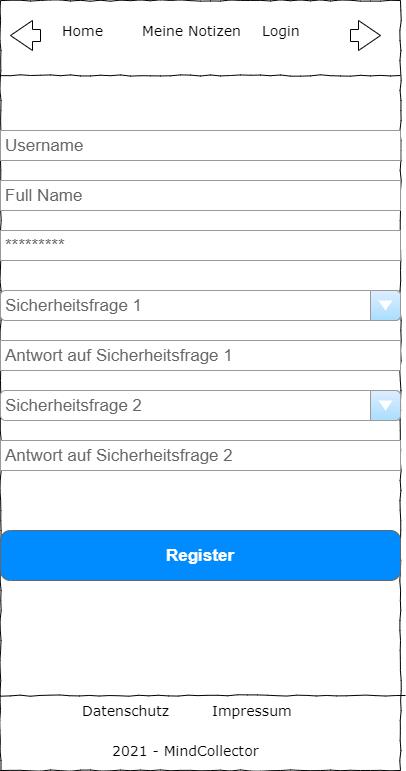
**Abbildung 7:** Datenschutz

12



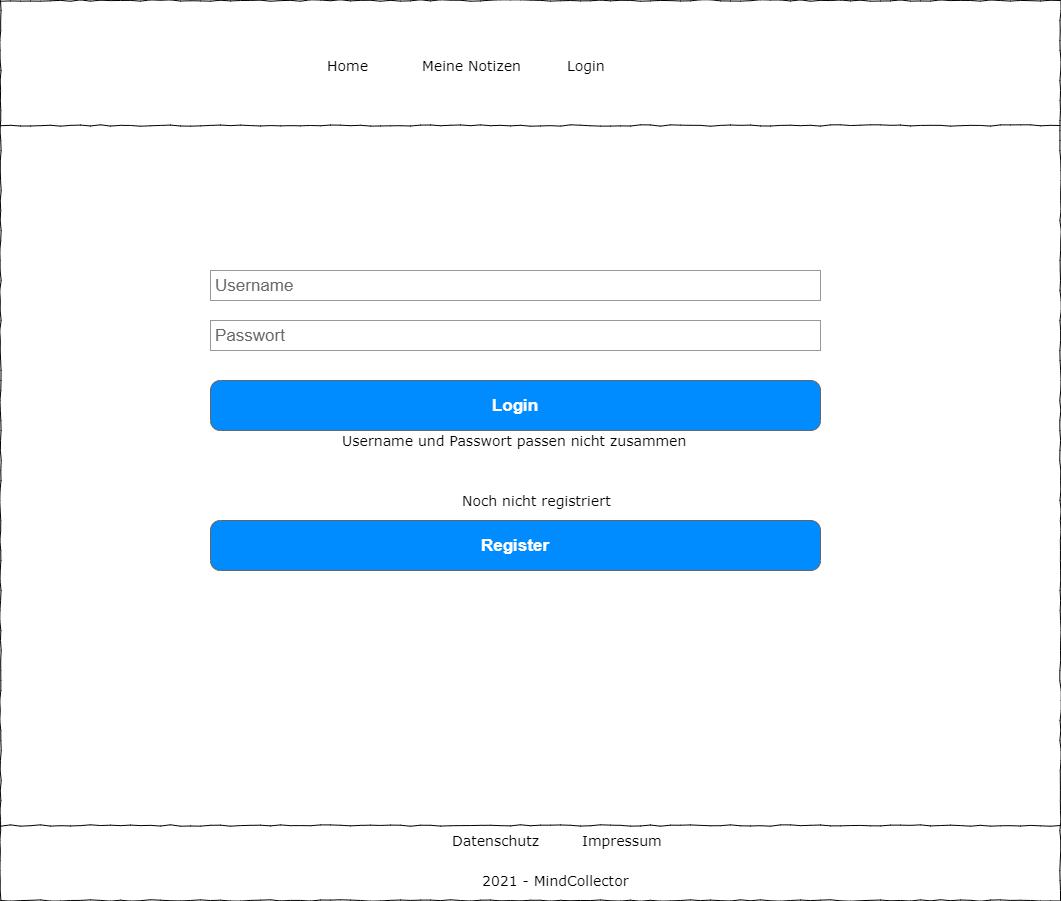
**Abbildung 8:** SignUp

13



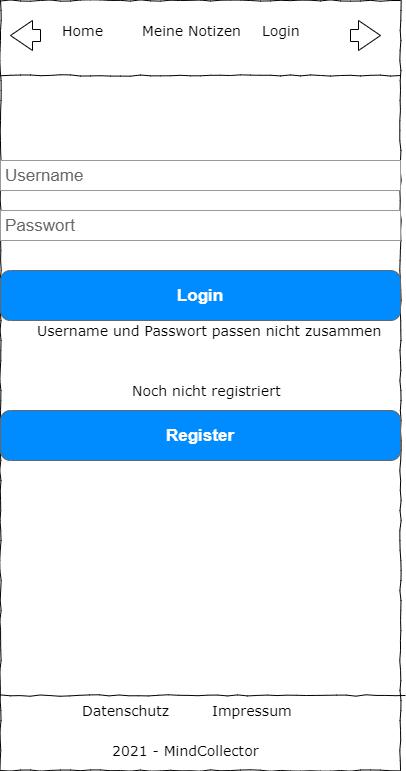
**Abbildung 9:** Signup-App

14



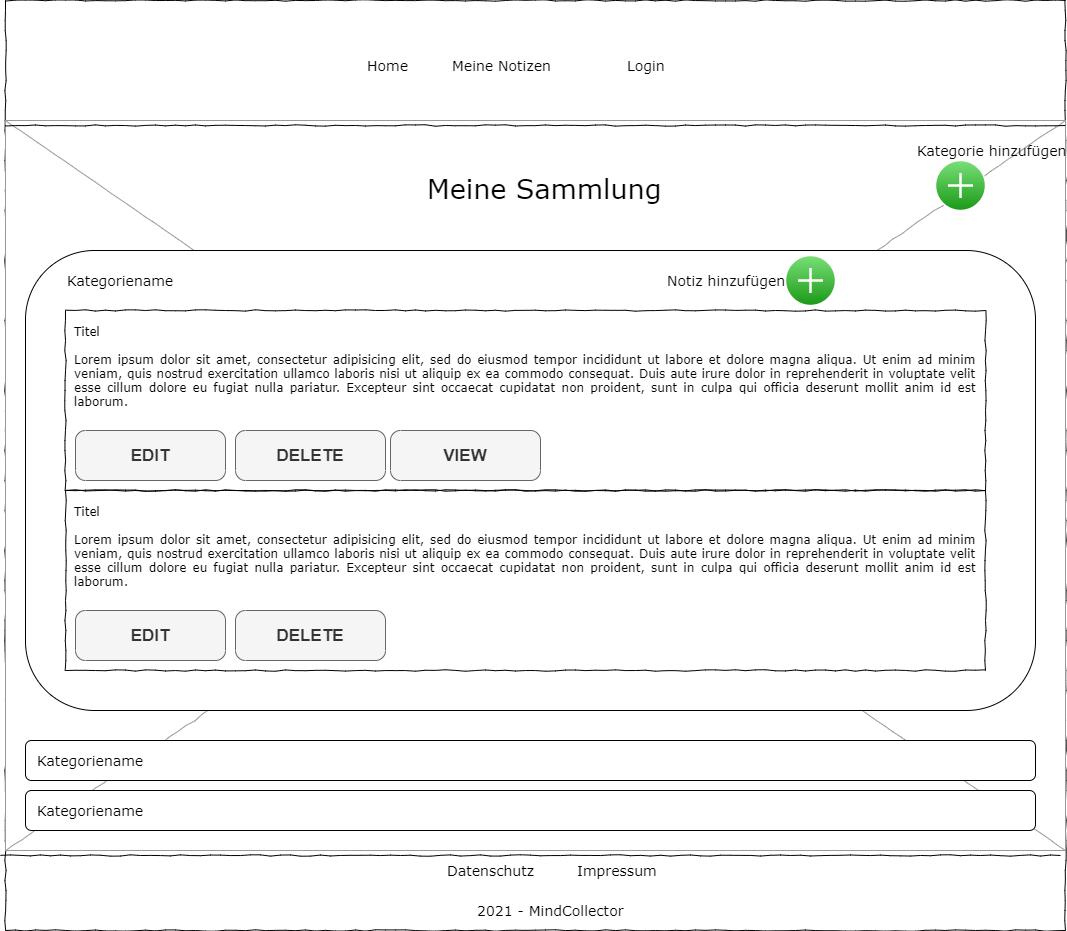
**Abbildung 10:** Login

15



**Abbildung 11:** Login-App

16



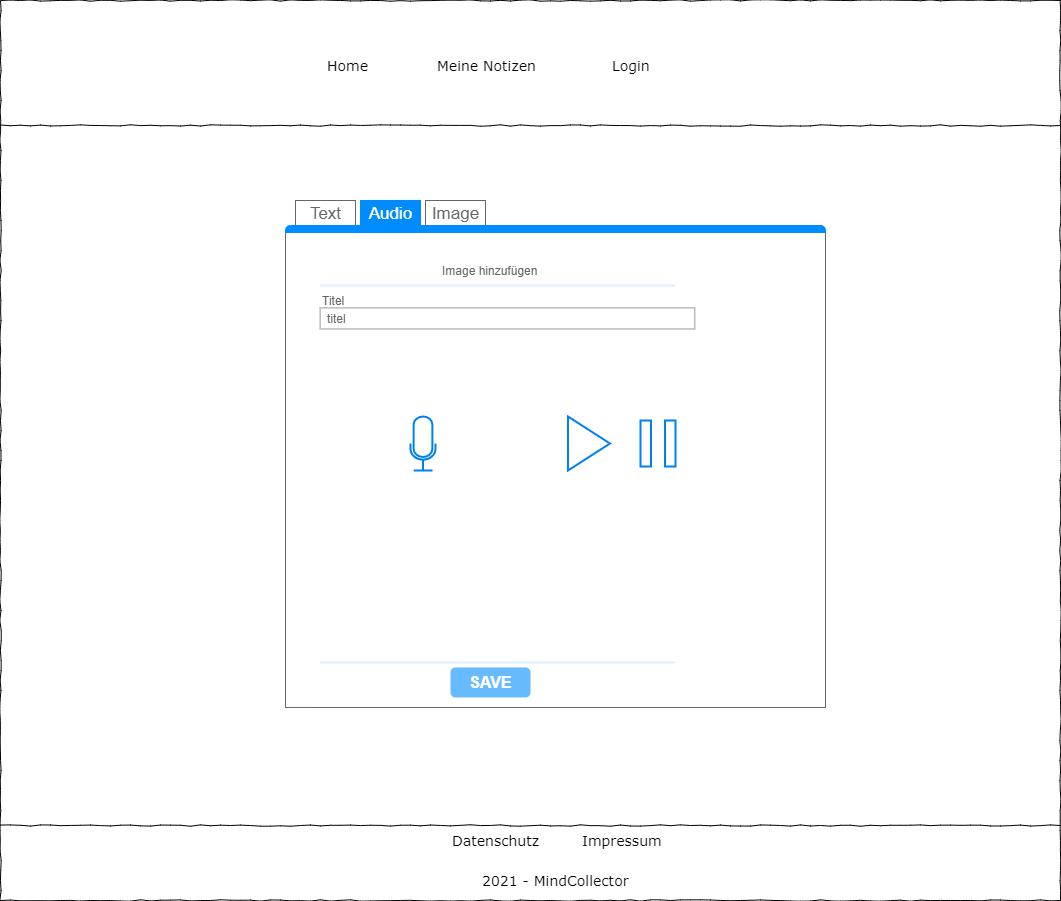
**Abbildung 12:** Meine Notizen

17



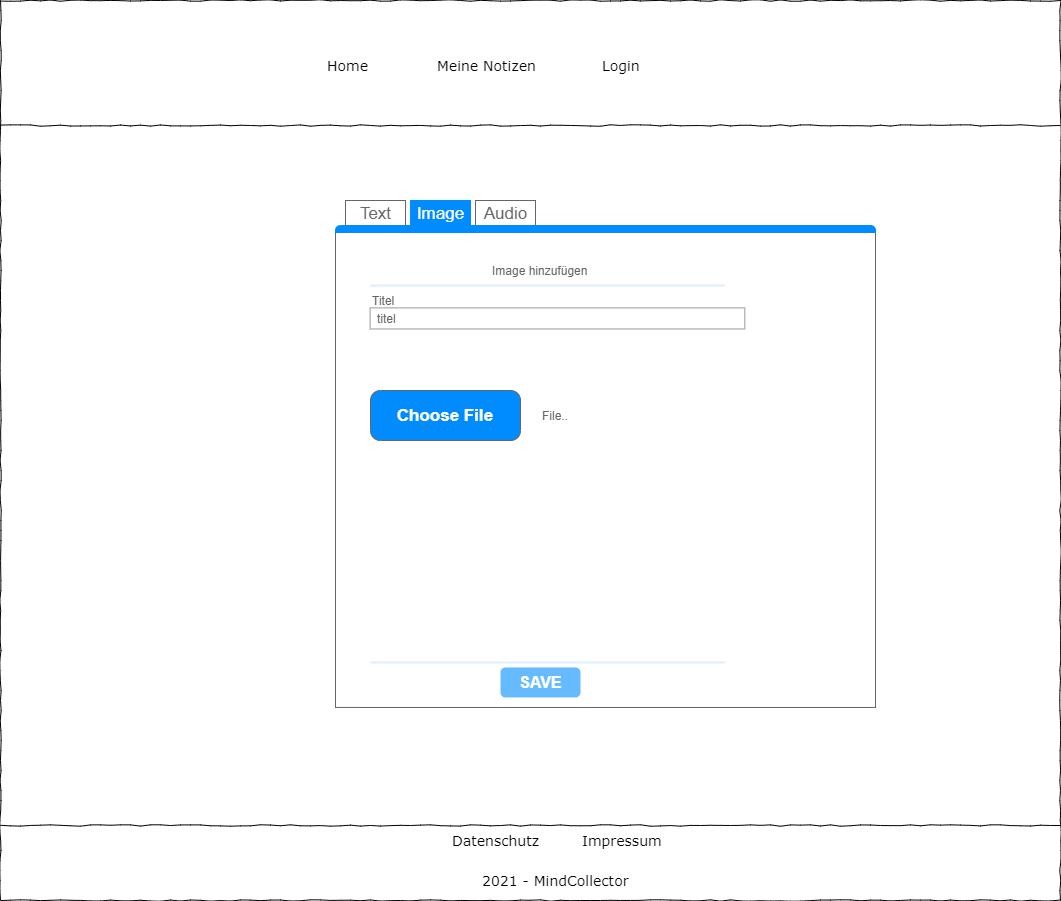
**Abbildung 13:** Meine Notizen-app

18



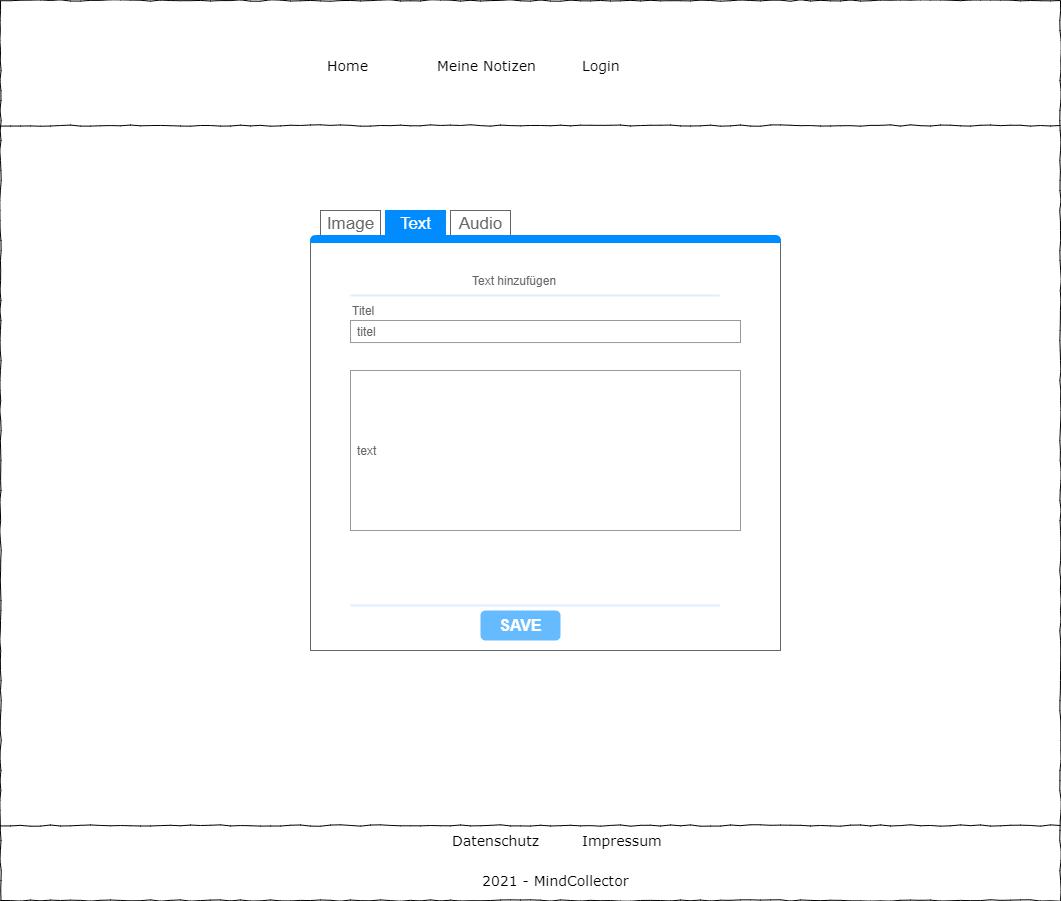
**Abbildung 14:** Notiz Audio hinzufügen

19



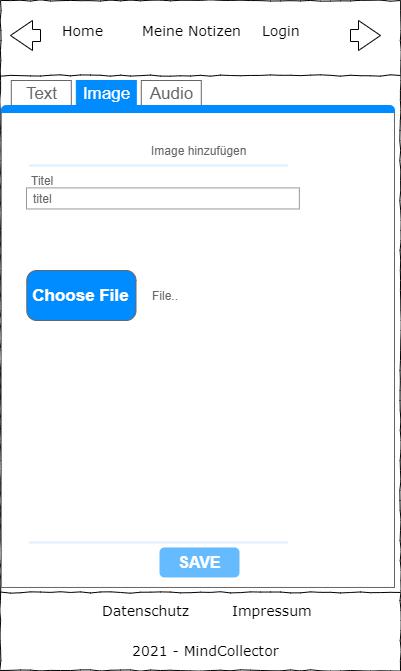
**Abbildung 15:** Notiz Bild hinzufügen

20



**Abbildung 16:** Notiz Text hinzufügen

21



**Abbildung 17:** Notiz Bild hinzufügen App

2.5 Anforderungen im Detail

* “Als Nutzer möchte ich mich an- und abmelden können, um meine Notizen sichern zu können.“

**–** Akzeptanz: an- abmelden ermöglichen.

* “Als Nutzer benötige ich eine Seite, um mich registrieren zu können.“

**–** Akzeptanz: Registrierseite vorhanden.

* “Als Nutzer möchte ich spontane Ideen oder Zitate abspeichern können, damit ich diese später nachlesen kann.“

**–** Akzeptanz: neue Ideen/Zitate können hinzugefügt werden.

22

* “Als Nutzer möchte ich Notizen, die ich hinzugefügt habe, löschen, falls ich diese nicht mehr benötige.“

**–** Akzeptanz: Notizen können gelöscht werden.

* “Als Nutzer möchte ich Notizen, die ich hinzugefügt habe, bearbeiten, so dass ich mögliche Fehler beheben kann.“

**–** Akzeptanz: Notizen können bearbeitet werden.

* “Als Nutzer möchte ich alle meine Notizen in Kategorien einteilen können, damit ich diese sortieren kann und einen Überblick über diese habe.“

**–** Akzeptanz: Notizen können in Kategorien eingeteilt werden.

* “Als Nutzer möchte ich Bilder von Text hochladen, so dass diese in Text umgewandelt werden können und zu meinen anderen Ideen hinzugefügt werden können.“

**–** Akzeptanz: Bild kann hochgeladen werden und wird zu Text umgewandelt.

* “Als Nutzer möchte ich Ideen als Sprachaufnahmen hochladen können, damit diese in Text umgewandelt werden und ich diese später nachlesen kann, ohne sie per Hand abtippen zu müssen.“

**–** Akzeptanz: Sprachaufnahmen können hochgeladen werden und werden in Textumgewandelt.

* “Als Gast möchte ich eine Seite sehen können, um mir einen Überblick zu verschaﬀen, was auf dieser Seite angeboten wird.“

**–** Akzeptanz: Startseite mit Informationen über Webanwendung kann eingesehenwerden.

* “Als Gast möchte ich die Möglichkeit haben mir einen Account anzulegen, damit ich die Funktionen der Seite nutzen kann.“

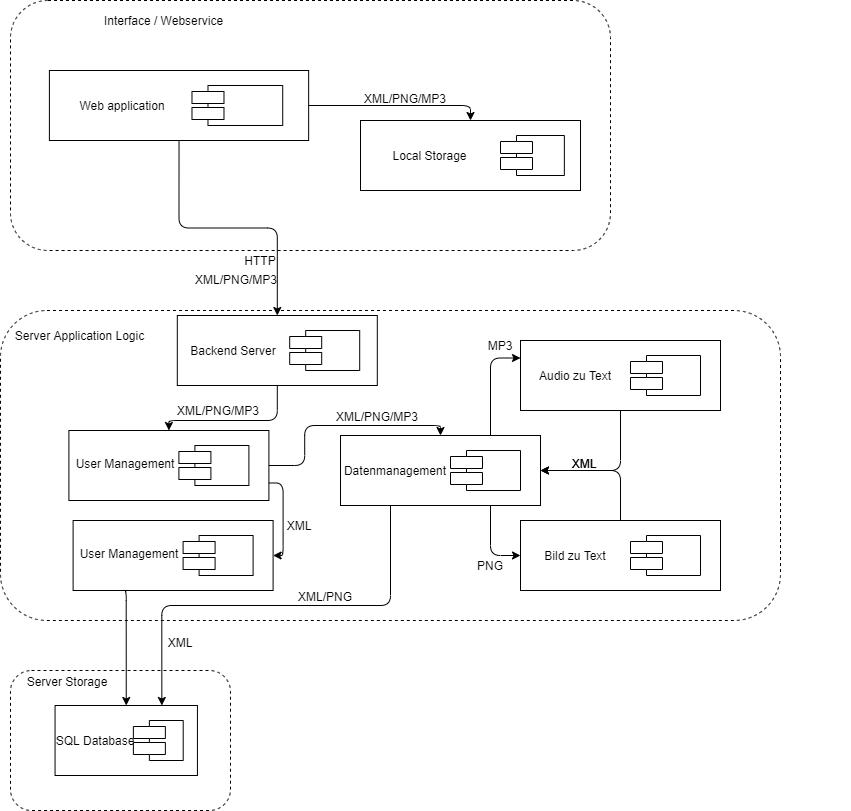
**–** Akzeptanz: SignUp-Seite ist vorhanden, bei der Nutzer sich selber anmeldenkönnen.

23

3 Technische Beschreibung

3.1 Systemübersicht

In der Abbildung [18](#page24) wird die systemische Übersicht der Anwendung, des Backends und der Datenbank hergestellt. Die Anwendung selbst hat einen speicher auf dem Gerät und Sachen ”zwischenzuspeichern”. Der Server beschäftigt sich mit der Verarbeitung des Audio-und Videosmaterials, so wie der Authentifizierung und Verwaltung der Daten in Form von Text, Bild und Audio.



**Abbildung 18:** Systemarchitekturdiagramm

24

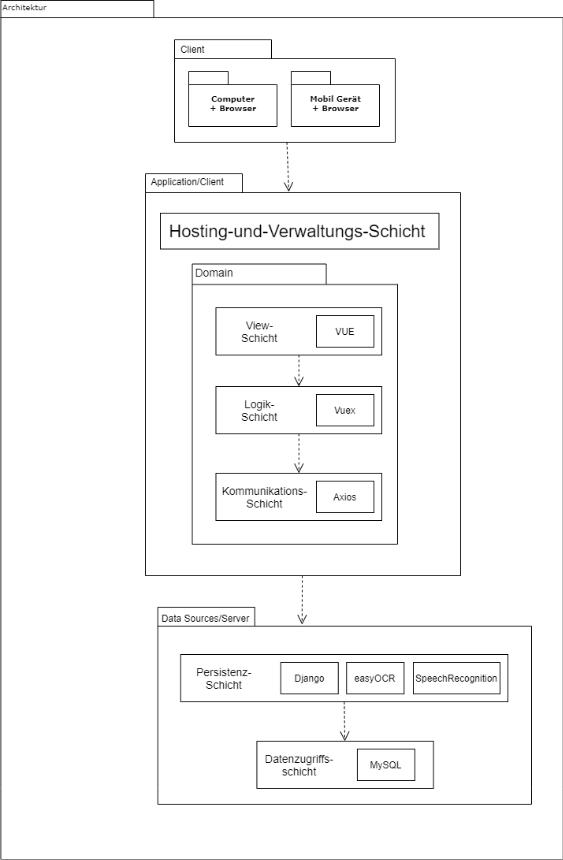
3.2 Softwarearchitektur

Die Schichten werden in diesem Fall durch Applikation und Server unterteilt. Die möglichen Endgeräte sind ein Computer mit einem installierten Browser und ein mobiles Gerät ebenfalls mit einem vorinstallierten Browser. Auf dem Client wird die Domain geladen. Diese beinhält die View-Schicht, da dort mit Hilfe des «VUE »Frameworks das Interface geschaﬀen wird, was der User benutzt. Der Client selbst besitzt eine Logik-Schicht, die sowohl die Daten der Webanwendung, als auch die Speicherung lokal vornimmt mit der Komponente «Axios »

Die Kommunikationsschicht wird auf dem Client durch «VUEX »übernommen, der sowohl

Lokal die Speicherung vornimmt und über die Verbindung mit dem Backend auch die

Kommunikation verrichtet.



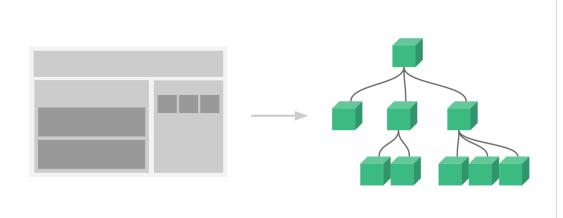
**Abbildung 19:** Schichtenmodell

VUE

Wie in allen gängigen Frameworks, wird auch in Vue komponentenorientiert gearbeitet. Eine Anwendung wird dabei in kleinere Bestandteile (Bausteine, Komponenten) zerlegt. Dadurch werden einfach wiederverwendbare und pflegbare Einheiten erstellt, die in eine hierarchische Struktur gebracht werden, um somit eine größere Anwendung Stück für Stück zusammenzusetzen. Komponenten bilden somit die fundamentalen Bausteine einer

25

Anwendung. In folgender Grafik wird veranschaulicht, wie eine Webseite in Komponenten zerlegt und in Form einer Baumstruktur in Beziehung zueinander gebracht werden kann:[1](#page26)



**Abbildung 20:** Baumstruktur VUE[1]

VUEX

Vuex ist ein State Management Pattern + Bibliothek für Vue.js Anwendungen. Es dient als zentraler Speicher für alle Komponenten in einer Anwendung, mit Regeln, die sicherstellen, dass der Zustand nur in einer vorhersehbaren Weise verändert werden kann. Es ist auch mit Vue’s oﬃzieller devtools Erweiterung integriert, um erweiterte Funktionen wie Zero-Config Time-Travel Debugging und State Snapshot Export / Import zu bieten.[2](#page26)

Axios

Axios ist eine Komponente, die zur sicheren Erfassung der Daten über einen Server funktioniert. Sie kommuniziert mit HTTP.

* XMLHttpRequests vom Browser aus erstellen
* Erstellen von http-Anfragen aus node.js
* Unterstützt die Promise-API
* Abfangen von Anfrage und Antwort
* Transformieren von Anfrage- und Antwortdaten
* Abbrechen von Anfragen
* Automatische Transformationen für JSON-Daten
* Client-seitige Unterstützung für den Schutz vor XSRF[3](#page26)

Django

Django ist ein freies, quelloﬀenes Web-Anwendungs-Framework, geschrieben in Python. Ein Web-(Anwendungs-)Framework ist eine Art Baukastensystem, das dir mit vielen vorgefertigten Teilen die Entwicklung von Web-Anwendungen stark erleichtert.

Wenn du eine Website entwickelst, brauchst du immer wieder sehr ähnliche Elemente:

* Auszug der Seite vuejs.de[1]

2 Auszug der Seite rathes.me[2]

3 Die Features von Axios wurden von npmjs.com[3] bezogen

26

Einen Weg, Benutzer zu verwalten (Registrierung, Anmeldung, Abmeldung etc.), einen Administrationsbereich, Formulare, Upload von Dateien usw.

Glücklicherweise wurde schon vor einiger Zeit erkannt, dass Web-Entwickler immer wieder die gleichen Probleme zu lösen haben. Gemeinsam entstanden so verschiedene Frameworks (Django ist so eines), welche die Web-Entwicklung durch vorgefertigte Elemente erleichtern. Frameworks sind dazu da, damit du das Rad nicht neu erfinden musst. Du kannst dich auf die konkret zu erfüllenden Anforderungen der Webseite kümmern. Die grundlegende Basis der Webseite stellt dir das Framework zur Verfügung.[4](#page27)

EasyOCR

Das EasyOCR-Paket wird von Jaided AI erstellt und gepflegt, einem Unternehmen, das sich auf Dienstleistungen im Bereich der optischen Zeichenerkennung spezialisiert hat. EasyOCR ist mit Python und der PyTorch-Bibliothek implementiert. Wenn Sie eine CUDA-fähige GPU haben, kann die zugrundeliegende PyTorch Deep-Learning-Bibliothek die Texterkennung und OCR-Geschwindigkeit enorm beschleunigen.

Zum jetzigen Zeitpunkt kann EasyOCR OCR-Text in 58 Sprachen erkennen, darunter Englisch, Deutsch, Hindi, Russisch und mehr! Die EasyOCR-Macher planen, in Zukunft weitere Sprachen hinzuzufügen. Die vollständige Liste der Sprachen, die EasyOCR unter-stützt, finden Sie auf der folgenden Seite.

Derzeit unterstützt EasyOCR nur OCR von getipptem Text. Später im Jahr 2020 ist geplant, auch ein Modell zur Handschrifterkennung herauszubringen![5](#page27)

SpeechRecognication

Ist eine Spracherkennung, die aus Audiodatein Text erkennt und diese abspeichern [kann[6]](#page37)

MySQL

MySQL verwaltet, zeigt, speichert und ändert Daten in Tabellen – die klassische Aufgabe eines Datenbank-Managementsystems (DBS). Es funktioniert dabei als Client-Server-System: Die jeweilige Datenbank ist der Server. Die Software auf der Client-Seite schickt Befehle an die Datenbank. Die Datenbank übersetzt die Befehle in ausführbaren Code, führt die Befehle aus und sendet die Informationen darüber an den Client.[6](#page27)

3.3 Schnittstellen

Die Schnittstelle zwischen dem Client und dem Backend findet durch«VUEX »auf der Seite des Clients und «Django »auf der Seite des Backends statt. Die Daten werden aufbereitet und je nach Format an den passenden Teil des Servers geschickt oder gerufen.

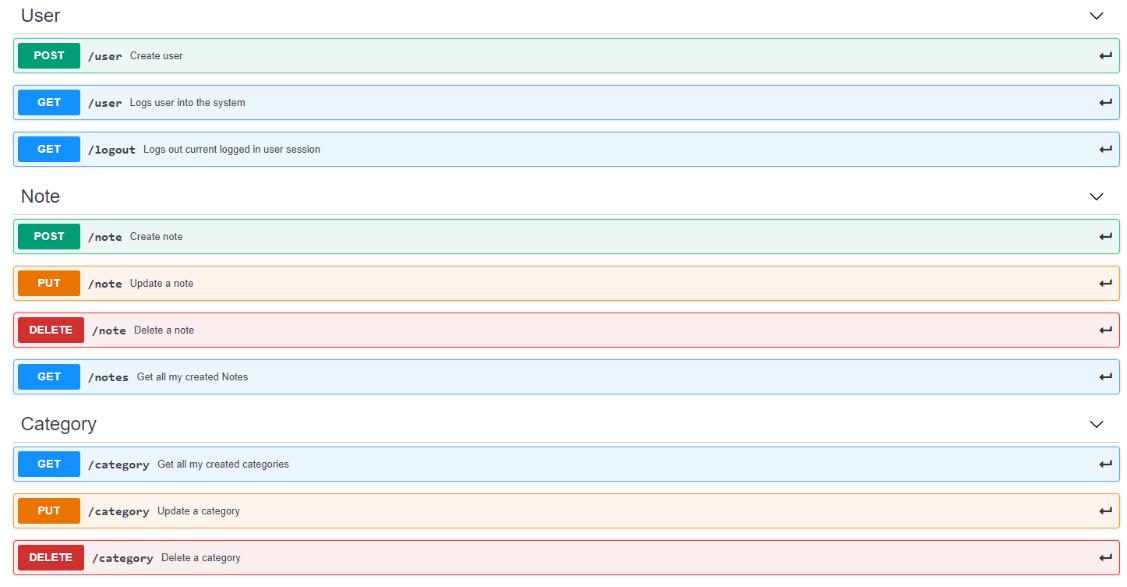
Da SQL auch Bitmaps oder Bilder speichern kann, werden nur für die Speicherung der Audiodatein ein Fileserver gebraucht.

* Die Beschreibung wurde von djangogirls.org[4] bezogen

5 Erläuterung bezogen von pyimagesearch.com[5]

6 Zusammenfassung von checkdomain.de[7]

27



**Abbildung 21:** Schnittstellenbeschreibung

Ereignisse

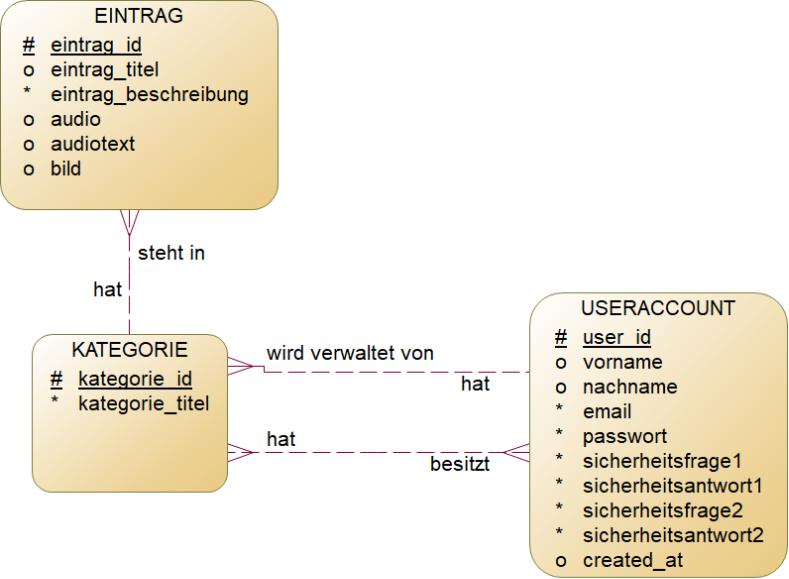
Die resultierenden Ergebnisse der Anwendung sind eine strukturierte Auflistung von Ideen. So kann Klarheit geschaﬀen werden, wenn man mehrere Eindrücke verwalten muss oder diese detailliert beschreiben will.

Durch die Möglichkeit, der Spracherkennung, soll eine schnelle ”Memo” gemacht werden können. Dieses Attribut kann eine überzeugende Fähigkeit der Anwendung darstellen, welches die Dokumentierung erleichtert.

28

3.4 Datenmodell

Das ER-Modell stellt alle notwendigen Entitäten der Anwendung dar. Da zur Verifizierung des Users zwei Sicherheitsfragen benötigt werden, fällt diese Entität besonders groß aus. Um die Ideen mit ihren einzelnen Beiträgen näher später definieren zu können, wird diese, so wie später auf der Domain, als einzelne Entität abgebildet.



**Abbildung 22:** ER-Modell

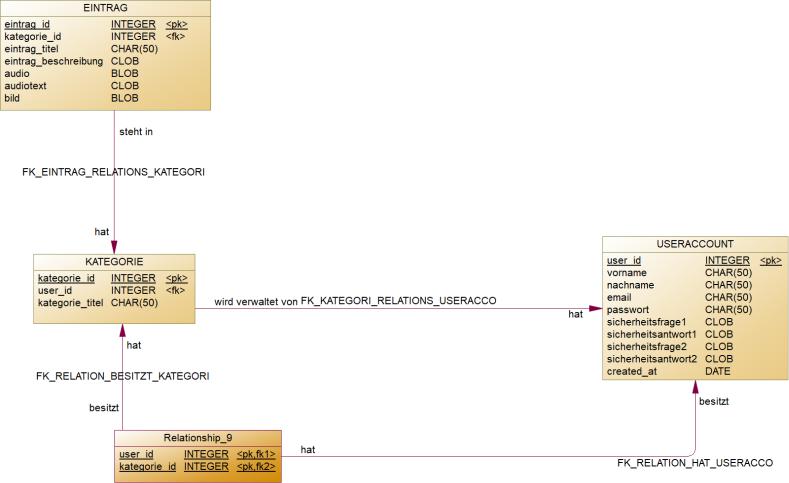
In dem physikalischen Datenmodell werden die zugehörigen Datentypen und Verhältnisse gegenüber dem ER-Modell klar dargestellt. So kann man Beispielsweise bei der Entität ”Eintrag” bei dem Attribut ”bild” erkennen, dass es ein Bild oder «BLOB »ist und die Beziehung zwischen ”Kategorie” und ”Useraccount” so geprägt ist, dass mehrere Kategorien möglich sind, aber in dem Fall des Teilens einer Kategorie auch mehrere Benutzer der Kategorie möglich wären.

Auch wenn die geplante Funktion eine Audio-Datei in Text zu konvertieren eine größere

Rolle spielen soll, ist in der Datenbank auch geplant, für spätere Verwendung, dass die

Audio-Datein gespeichert werden könnte.

29



**Abbildung 23:** Physikalisches Datenmodell

Angehängt zu dem physikalischen Datenmodell ist ein Implementierungsskript, welches nochmal detailliert die Verbindungen darstellt. Die einzelnen Verbindungen haben in dem Skript keine spezifischen Namen erhalten, weil sie so nicht wieder aufgerufen werden müssen.

* /∗==============================================================∗/

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | /∗ | DBMS name : | ORACLE V e r s i o n | 19c | ∗/ |
| 3 | /∗ | Created on : | 04.05.2021 12: | 39:36 | ∗/ |

* /∗==============================================================∗/

5

6

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | a l t e r | t a b l e EINTRAG | |
| 8 | drop c o n s t r a i n t | | FK\_EINTRAG\_RELATIONS\_KATEGORI; |
| 9 |  |  |  |
| 10 | a l t e r | t a b l e KATEGORIE | |
| 11 | drop c o n s t r a i n t | | FK\_KATEGORI\_RELATIONS\_USERACCO; |
| 12 |  |  |  |
| 13 | a l t e r | t a b l e RELATIONSHIP\_9 | |
| 14 | drop c o n s t r a i n t | | FK\_RELATION\_BESITZT\_KATEGORI; |
| 15 |  |  |  |
| 16 | a l t e r | t a b l e RELATIONSHIP\_9 | |
| 17 | drop c o n s t r a i n t | | FK\_RELATION\_HAT\_USERACCO; |
| 18 |  |  |  |
| 19 | drop | t a b l e EINTRAG | c a s c a d e c o n s t r a i n t s ; |
| 20 |  |  |  |

1. drop i n d e x RELATIONSHIP\_12\_FK ;

|  |  |
| --- | --- |
| 23 | drop t a b l e KATEGORIE c a s c a d e c o n s t r a i n t s ; |
| 24 |  |

1. drop i n d e x BESITZT\_FK ;
2. drop i n d e x HAT\_FK;

|  |  |  |  |
| --- | --- | --- | --- |
| 29 | drop | t a b l e | RELATIONSHIP\_9 c a s c a d e c o n s t r a i n t s ; |
| 30 |  |  |  |
| 31 | drop | t a b l e | USERACCOUNT c a s c a d e c o n s t r a i n t s ; |
| 32 |  |  |  |

30

1. /∗==============================================================∗/

34 /∗ Table : EINTRAG ∗/

1. /∗==============================================================∗/

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 36 | c r e a t e t a b l e EINTRAG | ( |  |  |
| 37 | EINTRAG\_ID | INTEGER | not | n u l l , |
| 38 | KATEGORIE\_ID | INTEGER | not | n u l l , |
| 39 | EINTRAG\_TITEL | CHAR(50) | not | n u l l , |
| 40 | EINTRAG\_BESCHREIBUNG CLOB | | not | n u l l , |
| 41 | AUDIO | BLOB, |  |  |
| 42 | AUDIOTEXT | CLOB, |  |  |
| 43 | BILD | BLOB, |  |  |
| 44 | c o n s t r a i n t PK\_EINTRAG primary key | | (EINTRAG\_ID) |  |

1. ) ;
2. /∗==============================================================∗/

48 /∗ Table : KATEGORIE ∗/

1. /∗==============================================================∗/

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 50 | c r e a t e t a b l e | KATEGORIE | ( |  |  |
| 51 | KATEGORIE\_ID | | INTEGER | not | n u l l , |
| 52 | USER\_ID |  | INTEGER | not | n u l l , |
| 53 | KATEGORIE\_TITEL | | CHAR(50) | not | n u l l , |
| 54 | c o n s t r a i n t | PK\_KATEGORIE primary key | | (KATEGORIE\_ID) | |

1. ) ;
2. /∗==============================================================∗/

58 /∗ Index : RELATIONSHIP\_12\_FK ∗/

1. /∗==============================================================∗/

60 c r e a t e i n d e x RELATIONSHIP\_12\_FK on KATEGORIE (

1. USER\_ID ASC
2. ) ;

63

1. /∗==============================================================∗/

65 /∗ Table : RELATIONSHIP\_9 ∗/

1. /∗==============================================================∗/

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 67 | c r e a t e t a b l e | RELATIONSHIP\_9 ( | |  |  |  |
| 68 | USER\_ID |  | INTEGER |  | not | n u l l , |
| 69 | KATEGORIE\_ID | | INTEGER |  | not | n u l l , |
| 70 | c o n s t r a i n t | PK\_RELATIONSHIP\_9 | | primary key | (USER\_ID, KATEGORIE\_ID) | |

1. ) ;
2. /∗==============================================================∗/

74 /∗ Index : HAT\_FK ∗/

1. /∗==============================================================∗/

76 c r e a t e i n d e x HAT\_FK on RELATIONSHIP\_9 (

1. USER\_ID ASC
2. ) ;

79

1. /∗==============================================================∗/

81 /∗ Index : BESITZT\_FK ∗/

1. /∗==============================================================∗/

83 c r e a t e i n d e x BESITZT\_FK on RELATIONSHIP\_9 (

1. KATEGORIE\_ID ASC
2. ) ;

86

1. /∗==============================================================∗/

88 /∗ Table : USERACCOUNT ∗/

31

1. /∗==============================================================∗/

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 90 | c r e a t e t a b l e USERACCOUNT ( | |  |  |
| 91 | USER\_ID | INTEGER | not | n u l l , |
| 92 | VORNAME | CHAR(50) | not | n u l l , |
| 93 | NACHNAME | CHAR(50) | not | n u l l , |
| 94 | EMAIL | CHAR(50) | not | n u l l , |
| 95 | PASSWORT | CHAR(50) | not | n u l l , |
| 96 | SICHERHEITSFRAGE1 | CLOB | not | n u l l , |
| 97 | SICHERHEITSANTWORT1 | CLOB | not | n u l l , |
| 98 | SICHERHEITSFRAGE2 | CLOB | not | n u l l , |
| 99 | SICHERHEITSANTWORT2 | CLOB | not | n u l l , |
| 100 | CREATED\_AT | DATE, |  |  |
| 101 | c o n s t r a i n t PK\_USERACCOUNT primary key | | (USER\_ID) |  |

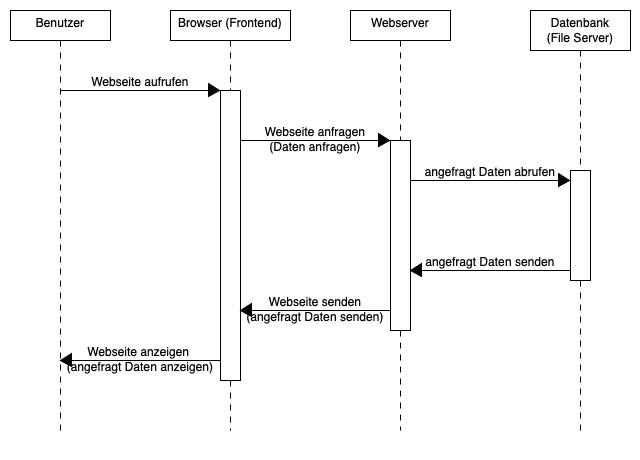
1. ) ;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 104 | a l t e r | t a b l e EINTRAG | |  |  |  |
| 105 | add | c o n s t r a i n t | FK\_EINTRAG\_RELATIONS\_KATEGORI | f o r e i g n | key | (KATEGORIE\_ID) |
| 106 |  | r e f e r e n c e s | KATEGORIE (KATEGORIE\_ID) ; |  |  |  |
| 107 |  |  |  |  |  |  |
| 108 | a l t e r | t a b l e KATEGORIE | |  |  |  |
| 109 | add | c o n s t r a i n t | FK\_KATEGORI\_RELATIONS\_USERACCO f o r e i g n key | | | (USER\_ID) |
| 110 |  | r e f e r e n c e s | USERACCOUNT (USER\_ID) ; |  |  |  |
| 111 |  |  |  |  |  |  |
| 112 | a l t e r | t a b l e RELATIONSHIP\_9 | |  |  |  |
| 113 | add | c o n s t r a i n t | FK\_RELATION\_BESITZT\_KATEGORI | f o r e i g n | key (KATEGORIE\_ID) | |
| 114 |  | r e f e r e n c e s | KATEGORIE (KATEGORIE\_ID) ; |  |  |  |
| 115 |  |  |  |  |  |  |
| 116 | a l t e r | t a b l e RELATIONSHIP\_9 | |  |  |  |
| 117 | add | c o n s t r a i n t | FK\_RELATION\_HAT\_USERACCO f o r e i g n key | | (USER\_ID) | |
| 118 |  | r e f e r e n c e s | USERACCOUNT (USER\_ID) ; |  |  |  |

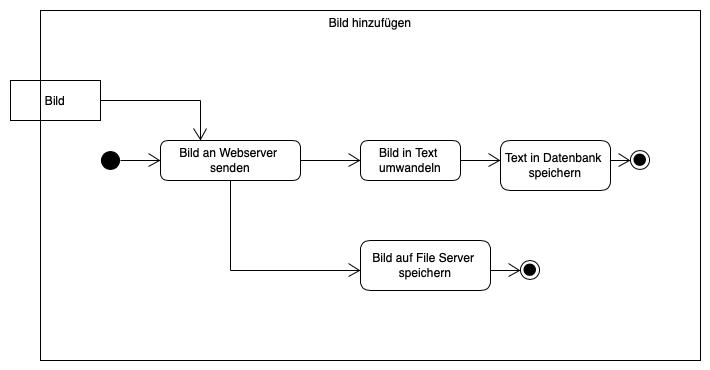
**Listing 1:** Implementierungsskript

32

3.5 Abläufe

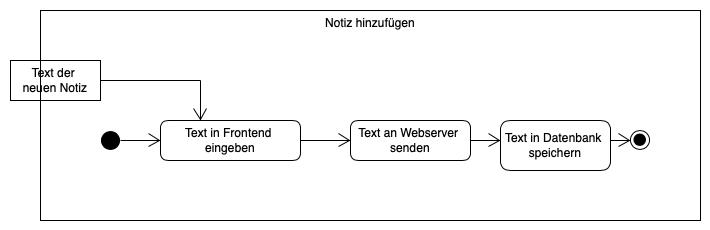


**Abbildung 24:** Kommunikation Client-Server



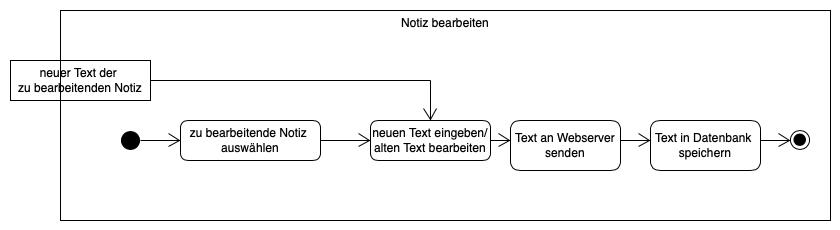
**Abbildung 25:** Ablauf Bild hinzufügen

Der Ablauf eine Audio-Datei hochzuladen soll so ablaufen, wie oben gezeigt. Also genauso, oder ähnlich, wie bei einem Bild.

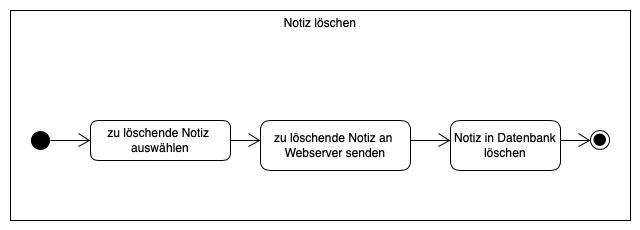


**Abbildung 26:** Ablauf Notiz hinzufügen

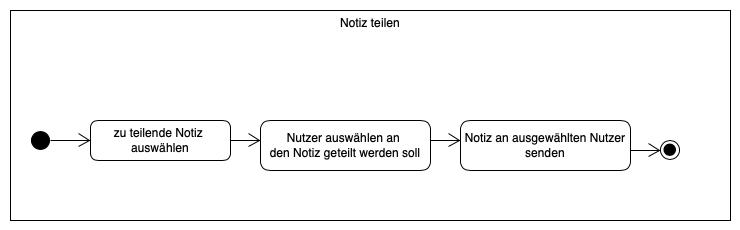
33



**Abbildung 27:** Ablauf Notiz bearbeiten

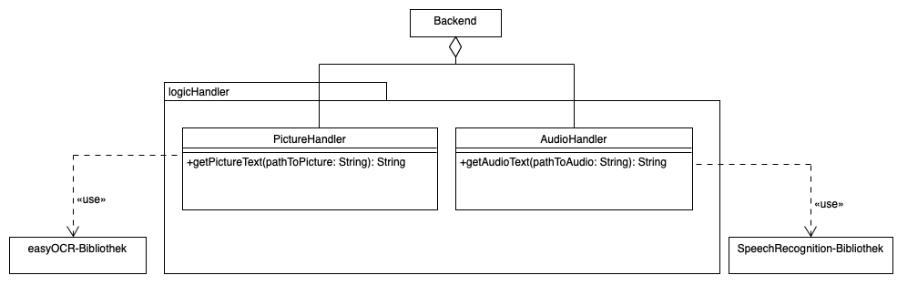


**Abbildung 28:** Ablauf Notiz löschen



**Abbildung 29:** Ablauf Notiz teilen

3.6 Entwurf



**Abbildung 30:** UML für Schnittstelle zwischen erweiterter Logik und dem Backend

34

4 Projektorganisation

4.1 Annahmen

Verwendete Technologien

* Frontend: VUE Framework, Axios (http), Vuex (Central Store)
* Backend: Django Framework, MySql, Ice Datenbank, phpMyAdmin
* Erkennung von Text in Bildern: easyOCR-Bibliothek
* Audio in Text Umwandlung: SpeechRecognition-Bibliothek
* Programmiersprachen: Python, JavaScript
* Mackup-Sprache: HTML
* Stylesheet-Sprache: CSS

Erweiterbarkeit

Das Softwareprodukt soll später erweiterbar sein, indem zum Beispiel weitere Funktionen eingefügt werden.

4.2 Verantwortlichkeiten

|  |  |
| --- | --- |
| **Softwarebaustein** | **Person** |
| Frontend | Marko Lemac |
|  |  |
| Backend | Fabian Wagenfeld |
| Erweiterte Logik | Leon Rudolph |

Softwarebausteine

**Frontend:** Graphische Oberfläche, die Interaktion mit dem Benutzer ermöglicht.

**Backend:** Webserver, mit integrierter Datenbank.

**Erweiterte Logik:** Funktionen, welche das Erkennen von Text auf Bildern er-möglicht, oder auch Sprachaufnahmen in Text umwandeln kann.

35

4.3 Grober Projektplan

* KW 18 (04.05)

**–** Abgabe Softwarespezifikation

**–** Recherche Themengebiete

* KW 19 (11.05)

**–** Einarbeitung in Themengebiete

**–** Implementierung (1)

* KW 20 (18.05)

**–** Implementierung (2)

**–** Integration (1)

* KW 21 (25.05)

**–** Präsentation des MVPs

* KW 22-23 (08.06)

**–** Implementierung (3)

* KW 24 (15.06)

**–** Integration (2)

* KW 25 (22.06)

**–** Testen

* KW 26/27 (29.06/06.07)

**–** Präsentation der Software

36

Literaturverzeichnis

1. vuejs.de. (2020). “Die Vue DevTools,” Adresse: [https : / / vuejs . de / artikel /](https://vuejs.de/artikel/vuejs-tutorial-deutsch-anfaenger/#:~:text=Vue%20(oder%20Vue.,oder%20Angular%20(Google)%20unterst%C3%BCtzt.) [vuejs- tutorial- deutsch- anfaenger/#:~:text=Vue%20(oder%20Vue.,oder%](https://vuejs.de/artikel/vuejs-tutorial-deutsch-anfaenger/#:~:text=Vue%20(oder%20Vue.,oder%20Angular%20(Google)%20unterst%C3%BCtzt.) [20Angular%20(Google)%20unterst%C3%BCtzt.](https://vuejs.de/artikel/vuejs-tutorial-deutsch-anfaenger/#:~:text=Vue%20(oder%20Vue.,oder%20Angular%20(Google)%20unterst%C3%BCtzt.) (besucht am 04. 05. 2021).
2. rathes.me. (2017). “Was ist Vuex?” Adresse: <https://rathes.me/blog/de/vuex-einstieg/>(besucht am 04. 05. 2021).
3. npmjs.com. (2020). “Features from Axios,” Adresse: [https : / / www . npmjs . com /](https://www.npmjs.com/package/axios#features) [package/axios#features](https://www.npmjs.com/package/axios#features) (besucht am 04. 05. 2021).
4. djangogirls.org. (2020). “Django - Was ist das?” Adresse: [https : / / tutorial](https://tutorial.djangogirls.org/de/django/) . [djangogirls.org/de/django/](https://tutorial.djangogirls.org/de/django/) (besucht am 04. 05. 2021).
5. pyimagesearch.com. (2020). “What is the EasyOCR package?” Adresse: [https](https://www.pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/) : [//www.pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/](https://www.pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/) (besucht am 04. 05. 2021).
6. pypi.org. (2020). “Project description,” Adresse: [https : / / pypi . org / project /](https://pypi.org/project/SpeechRecognition/) [SpeechRecognition/](https://pypi.org/project/SpeechRecognition/) (besucht am 04. 05. 2021).
7. checkdomain.de. (2021). “Wie funktioniert MySQL?” Adresse: [https : / / www](https://www.checkdomain.de/hosting/lexikon/mysql/#:~:text=MySQL%20ist%20ein%20quelloffenes%20SQL,die%20Grundlage%20f%C3%BCr%20dynamische%20Webseiten.&text=SQL%20steht%20f%C3%BCr%20%E2%80%9AStructured%20Query,f%C3%BCr%20den%20Zugriff%20auf%20Datenbanken.) .

[checkdomain . de / hosting / lexikon / mysql / # : ~ : text = MySQL % 20ist % 20ein %](https://www.checkdomain.de/hosting/lexikon/mysql/#:~:text=MySQL%20ist%20ein%20quelloffenes%20SQL,die%20Grundlage%20f%C3%BCr%20dynamische%20Webseiten.&text=SQL%20steht%20f%C3%BCr%20%E2%80%9AStructured%20Query,f%C3%BCr%20den%20Zugriff%20auf%20Datenbanken.) [20quelloffenes % 20SQL , die % 20Grundlage % 20f % C3 % BCr % 20dynamische %](https://www.checkdomain.de/hosting/lexikon/mysql/#:~:text=MySQL%20ist%20ein%20quelloffenes%20SQL,die%20Grundlage%20f%C3%BCr%20dynamische%20Webseiten.&text=SQL%20steht%20f%C3%BCr%20%E2%80%9AStructured%20Query,f%C3%BCr%20den%20Zugriff%20auf%20Datenbanken.) [20Webseiten . &text = SQL % 20steht % 20f % C3 % BCr % 20 % E2 % 80 % 9AStructured %](https://www.checkdomain.de/hosting/lexikon/mysql/#:~:text=MySQL%20ist%20ein%20quelloffenes%20SQL,die%20Grundlage%20f%C3%BCr%20dynamische%20Webseiten.&text=SQL%20steht%20f%C3%BCr%20%E2%80%9AStructured%20Query,f%C3%BCr%20den%20Zugriff%20auf%20Datenbanken.) [20Query , f % C3 % BCr % 20den % 20Zugriff % 20auf % 20Datenbanken.](https://www.checkdomain.de/hosting/lexikon/mysql/#:~:text=MySQL%20ist%20ein%20quelloffenes%20SQL,die%20Grundlage%20f%C3%BCr%20dynamische%20Webseiten.&text=SQL%20steht%20f%C3%BCr%20%E2%80%9AStructured%20Query,f%C3%BCr%20den%20Zugriff%20auf%20Datenbanken.) (besucht am 04. 05. 2021).

37