**Detailspezifikation**

**Forum TNT**

**für GIBS**

Dokumentname: Detailspezifikation.docx

Version: 1.0

Klassifizierung: vertraulich

Autor: Nico Schönbächler, Elias Tonini, Tim Schurtenberger

Letzte Änderung: 08.01.2019

Verteiler: Daniel Mosimann

**Änderungsübersicht:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Datum** | **Autor** | **Beschreibung der Änderung** | **Betroffene Kapitel** |
| 1.0 | 08.01.2019 | Nico Schönbächler | Erste Version | Alle |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Referenzierte Dokumente:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokument-bezeichnung** | **Dokumentname** | **Version** | **Kurzbeschreibung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Inhaltsverzeichnis**

[1. Einleitung 4](#_Toc535225739)

[1.1. Inhalt und Zweck des Dokuments 4](#_Toc535225740)

[1.2. Abkürzungen und Definitionen 4](#_Toc535225741)

[2. Fachliche Spezifikation 5](#_Toc535225742)

[2.1. Spezifische Benutzeranforderungen 5](#_Toc535225743)

[2.1.1. Anforderungen 5](#_Toc535225744)

[2.1.1. Log-In Use Case 5](#_Toc535225745)

[2.1.2. Sing-In Use Case 6](#_Toc535225746)

[2.1.3. Get All Blog Use Case 6](#_Toc535225747)

[2.1.4. Get Single Blog Use Case 7](#_Toc535225748)

[2.1.5. Create Blog Use Case 7](#_Toc535225749)

[2.1.6. Edit Blog Use Case 8](#_Toc535225750)

[2.1.7. Delete Blog Use Case 8](#_Toc535225751)

[2.2. Benutzerschnittstellen 9](#_Toc535225752)

[2.2.1. Gestaltung der Homeseite 9](#_Toc535225753)

[2.2.1. Gestaltung des Logins 9](#_Toc535225754)

[2.2.2. Gestaltung der Formulare 10](#_Toc535225755)

[2.3. Anforderungen an die Sicherheit 10](#_Toc535225756)

[3. Technische Spezifikation 11](#_Toc535225757)

[3.1. Strukturierung des Systems 11](#_Toc535225758)

[3.2. Physikalische Datenbeschreibung (Datenstrukturtypen) 11](#_Toc535225759)

[3.3. Sicherheit 12](#_Toc535225760)

[3.4. Anforderungen an die Entwicklungsumgebung 12](#_Toc535225761)

# Einleitung

## Inhalt und Zweck des Dokuments

Dieses Dokument beschreibt alle technische Spezifikation, des Projekts „TNT“, im Detail. Es beinhaltet Usecase-Diagramme, welche den Ablauf und die Funktionalitäten erklären, sowie Klassendiagramme, Datenbankdiagramm, Designs und Funktionsbeschreibungen.

## Abkürzungen und Definitionen

**Abkürzungen:**

|  |  |
| --- | --- |
| HW | Hardware |
| SW  DB | Software  Datenbank |
| HTTPS | Hypertext Transfer Protocol Secure |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Fachliche Spezifikation

## Spezifische Benutzeranforderungen

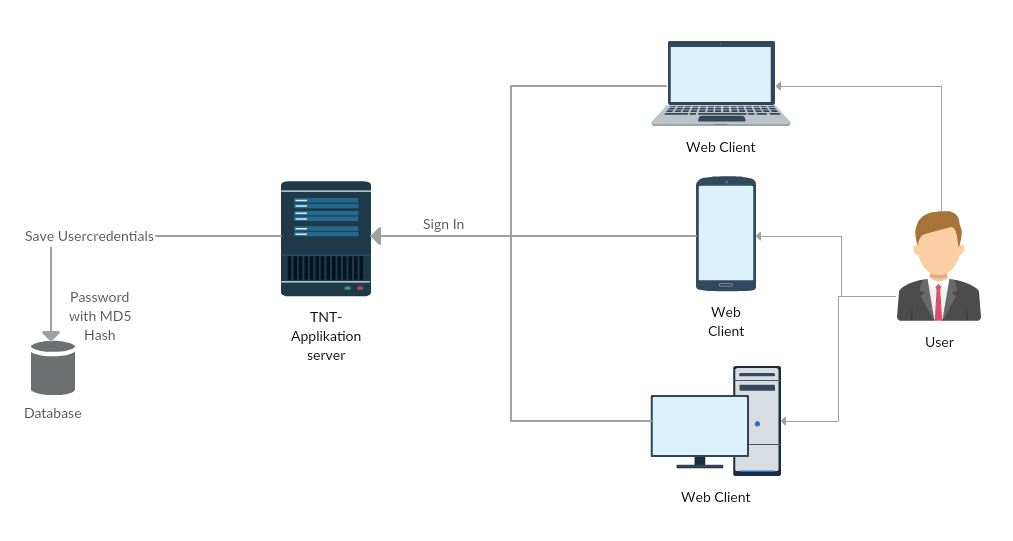
### Anforderungen

Benutzer können nur von den Administratoren erstellt und verwaltet werden.   
Um einen Benutzer erstellen, müssen folgende Informationen gegeben sein:

* Name
* Vorname
* Benutzername
* Passwort

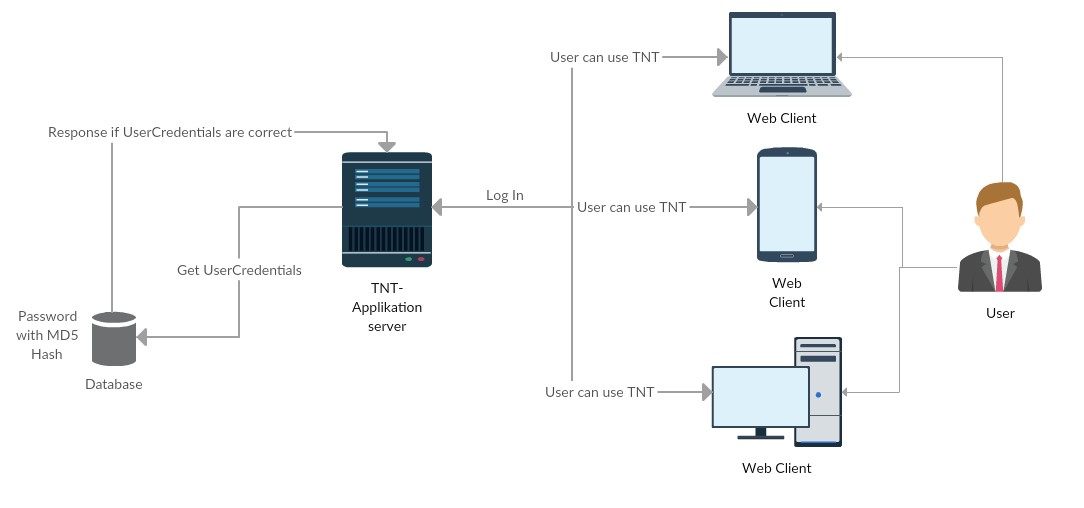
Damit der Benutzer sich anmelden kann, muss er sein Benutzername und sein Passwort kennen. Ist er einmal angemeldet, kann er zu ihm zugewiesenen Tasks, Zeiten erfassen.

### Log-In Use Case



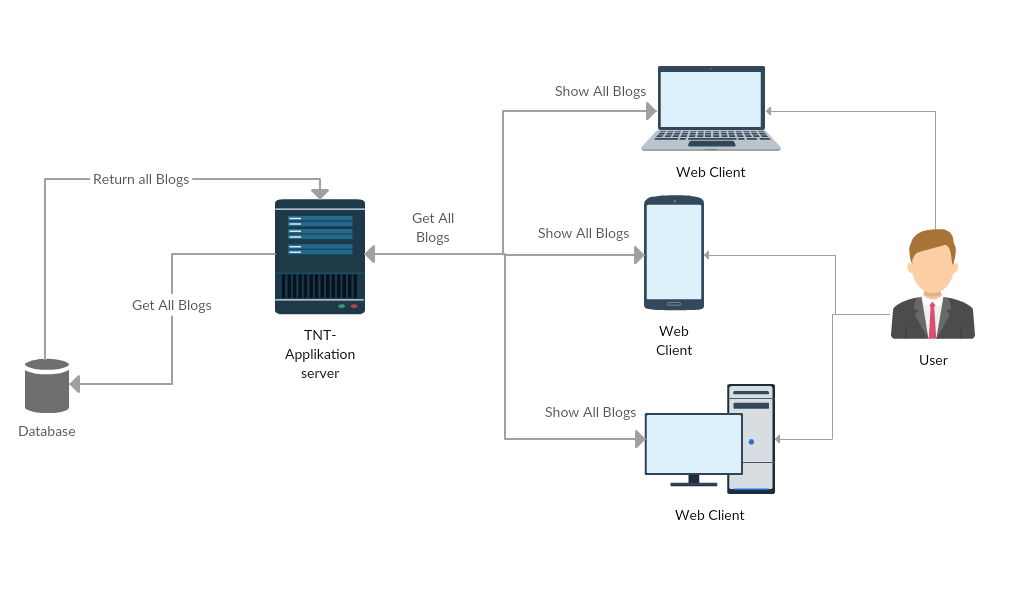
Der User greift mit einem Browser auf die Webapplikation zu. Der TNT Webapplikationsserver fordert anschliessend der User auf, sich zu registrieren. Falls die Eingabedaten korrekt sind, werden die Anmeldedaten verschlüsselt in der Datebank hinterlegt, und der User ist angemeldet.

### Sing-In Use Case



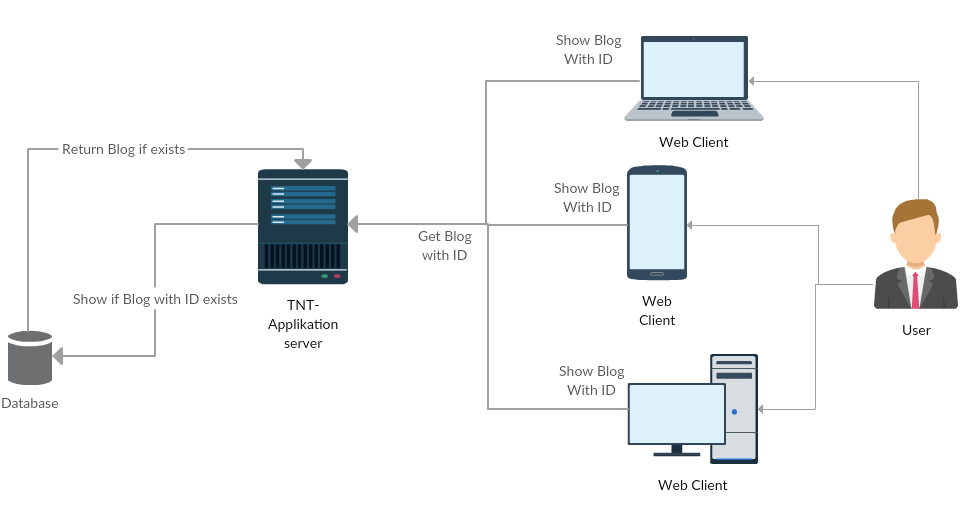
Wenn der User sich schon mal, bei der Applikation registriert hat, kann dieser mit seinen Anmeldedaten sich bei der Webapplikation anmelden.

### Get All Blog Use Case



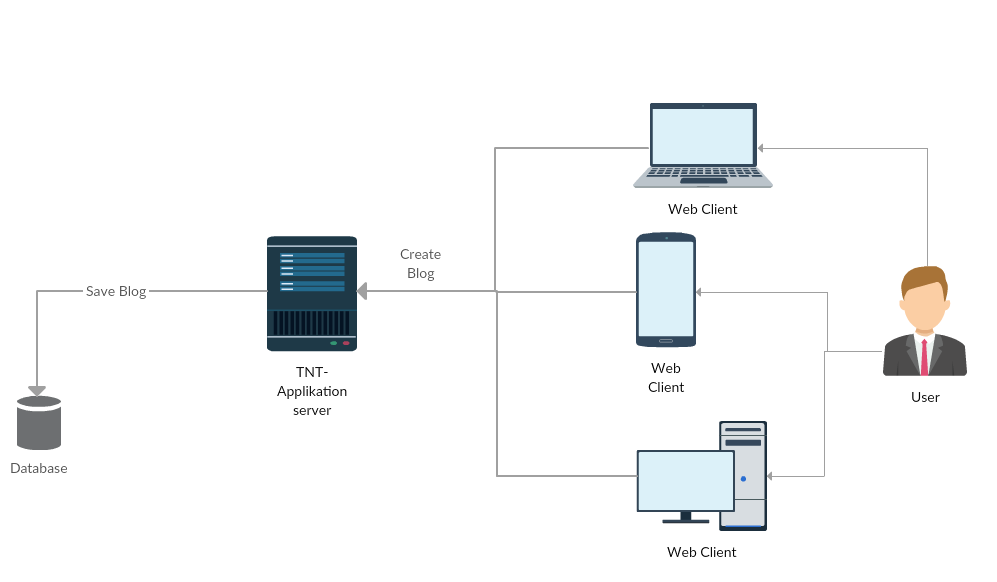
Wenn der User alle Blogs anschauen will, kann er eine Anfrage an den TNT-Webapplikationsserver starten. Falls der User angemeldet ist, holt der Applikationsserver die alle Blogs und schickt diese zurück.

### Get Single Blog Use Case



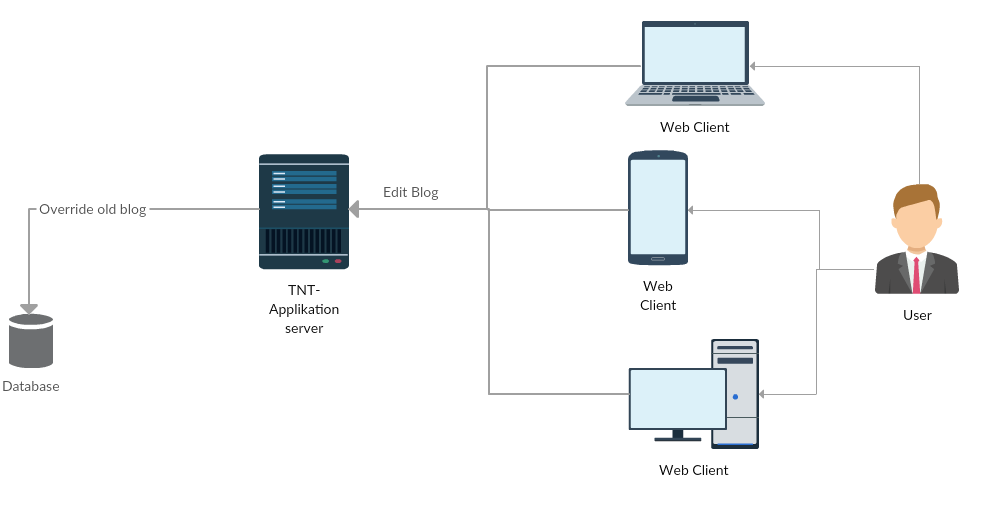
Der angemeldete User schickt eine Anfrage mit einer ID über das Webinterface der TNT Applikation. Mit der ID kann der TNT-Applikationsserver den passenden Blog auf der Datenbank finden, und schickt diesen, wenn er existiert zurück. Andernfalls schickt das Back-End eine 404 Fehlermeldung an den User.

### Create Blog Use Case



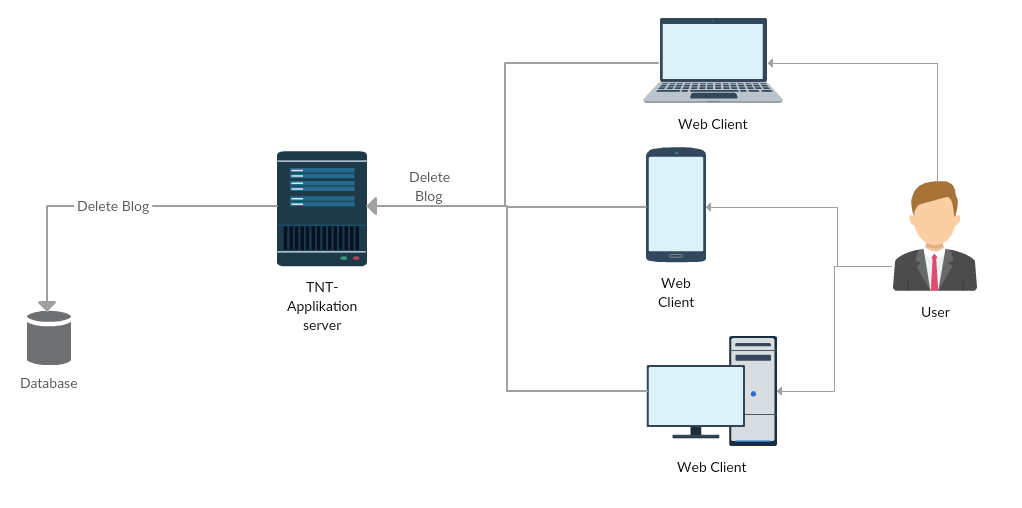
Der User schickt korrekte Blog-Daten an den TNT-Webapplikationsserver. Falls diese korrekt sind, speichert dieser die Daten auf der Datenbank. Andernfalls wird eine 400 Fehlermeldung an den User zurückgeschickt.

### Edit Blog Use Case



Der User schickt die neuen Daten an den TNT-Webapplikationsserver. Dieser überprüft zuerst ob der Blog schon existiert, bzw. ob der User Berechtigung darauf hat. Wenn ja, überschreibt er die Daten des Blogs und schickt dem User eine Meldung, dass der Blog updated wurde. Wenn nicht schickt er dem User eine 404 Fehlermeldung bezüglich des Blogs zurück. Wenn der User keine Berechtigung auf den Blog hat, wird eine 401 Fehlermeldung zurückgeschickt.

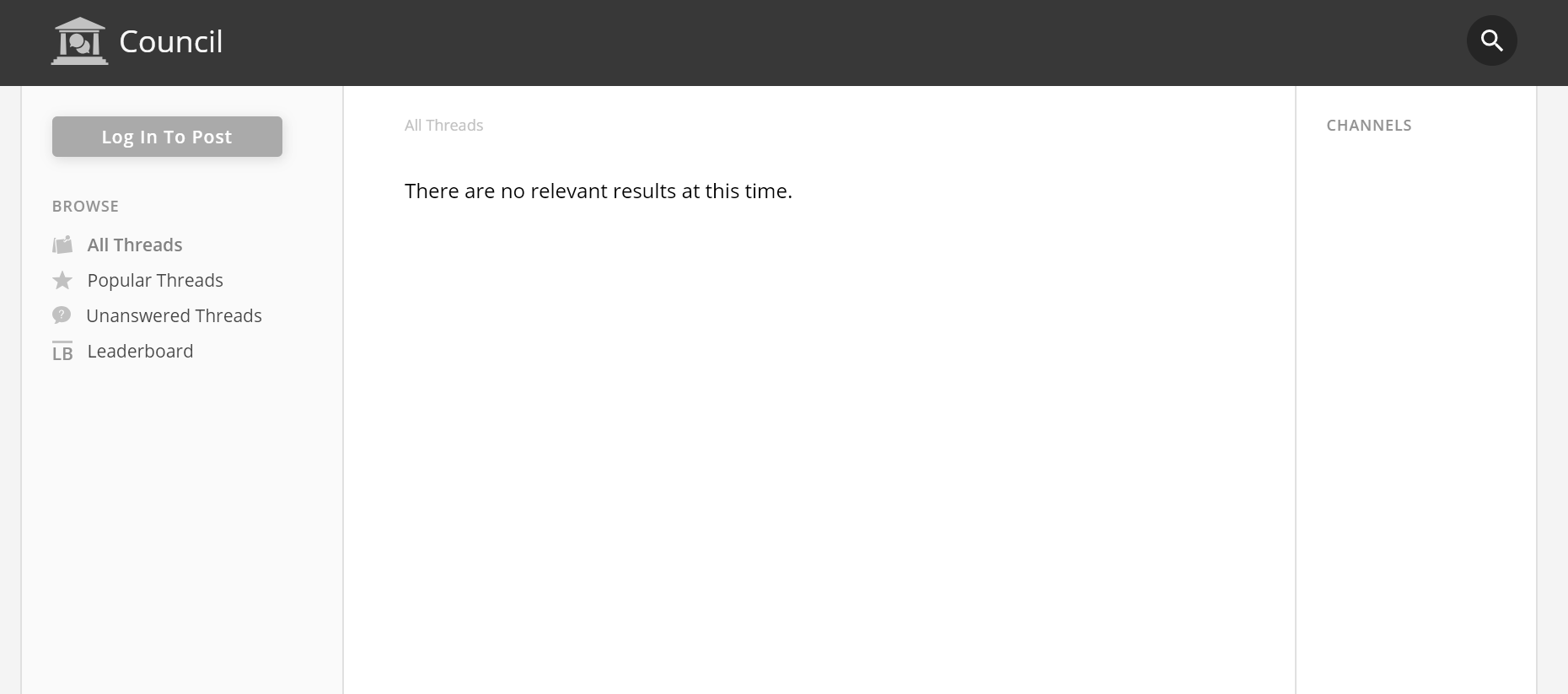
### Delete Blog Use Case



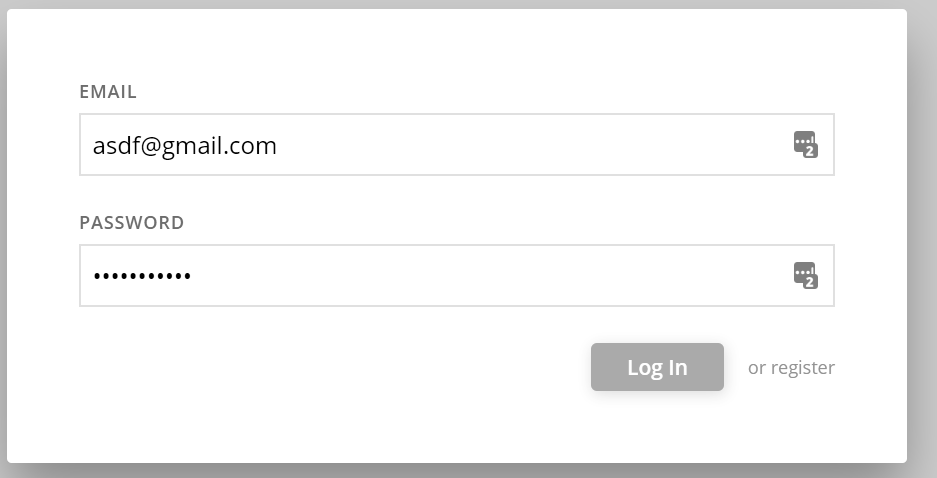
Der User schickt eine Id mit dem Löschbefehl an den TNT Webapplikationsserver. Dieser überprüft zuerst ob der Blog existiert. Wenn der Blog nicht existiert wird dem User eine 404 Fehlermeldung bezüglich des Blogs zurückgesendet. Wenn der User keine Berechtigung auf den Blog hat, bekommt der User eine 401 Fehlermeldung. Wenn jedoch beides zutrifft und der User den vorhandenen Blog löschen darf, wird der Blog gelöscht und dem User eine Meldung nach Beendung des Prozesses geschickt.

## Benutzerschnittstellen

### Gestaltung der Homeseite

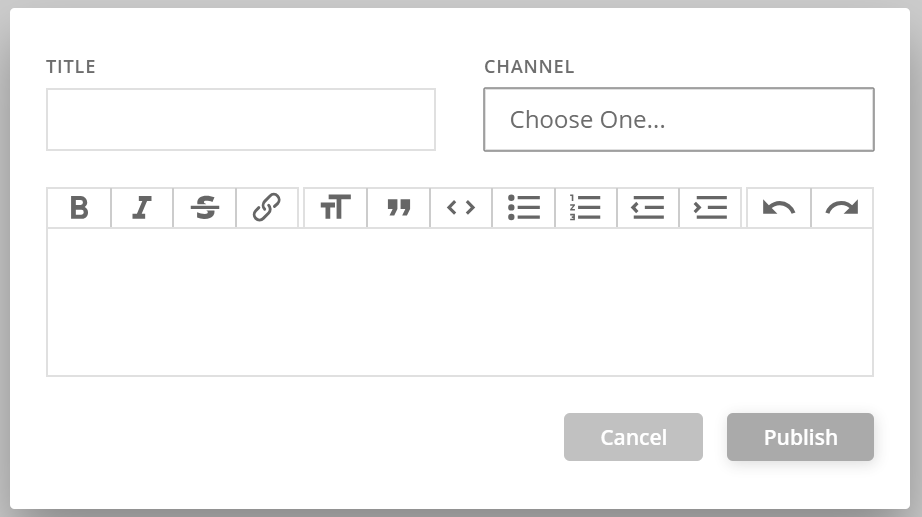


### Gestaltung des Logins



### Gestaltung der Formulare

Über diese Formulare können alle Daten erfasst und bearbeitet werden, sofern die nötigen Berechtigungen vorhanden sind.



## Anforderungen an die Sicherheit

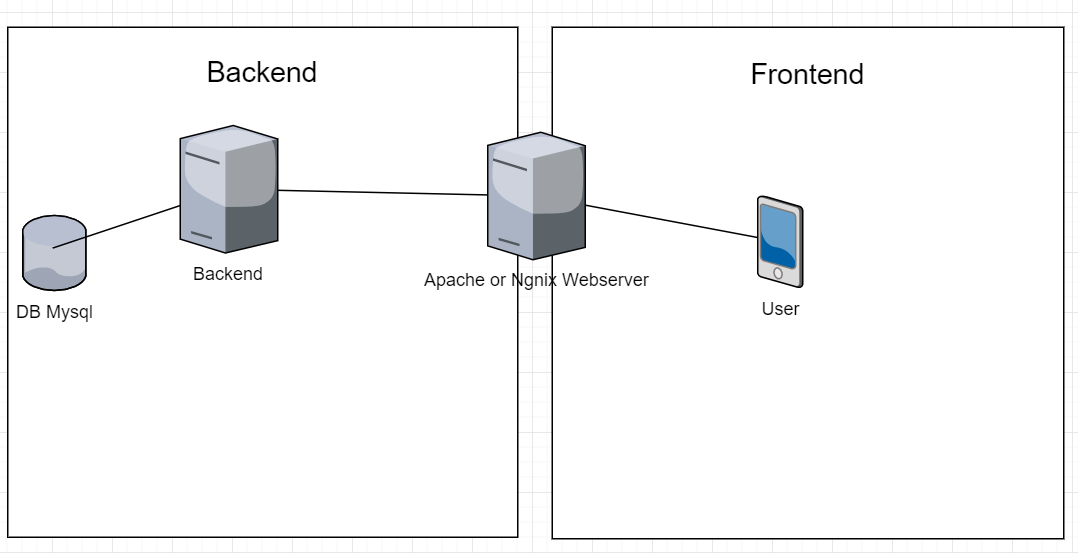
Jeder Benutzer besitzt eigene Zugangsdaten (E-Mail und Passwort).

Um unerwünschte Eingaben zu verhindern, werden folgende Massnahmen getroffen:

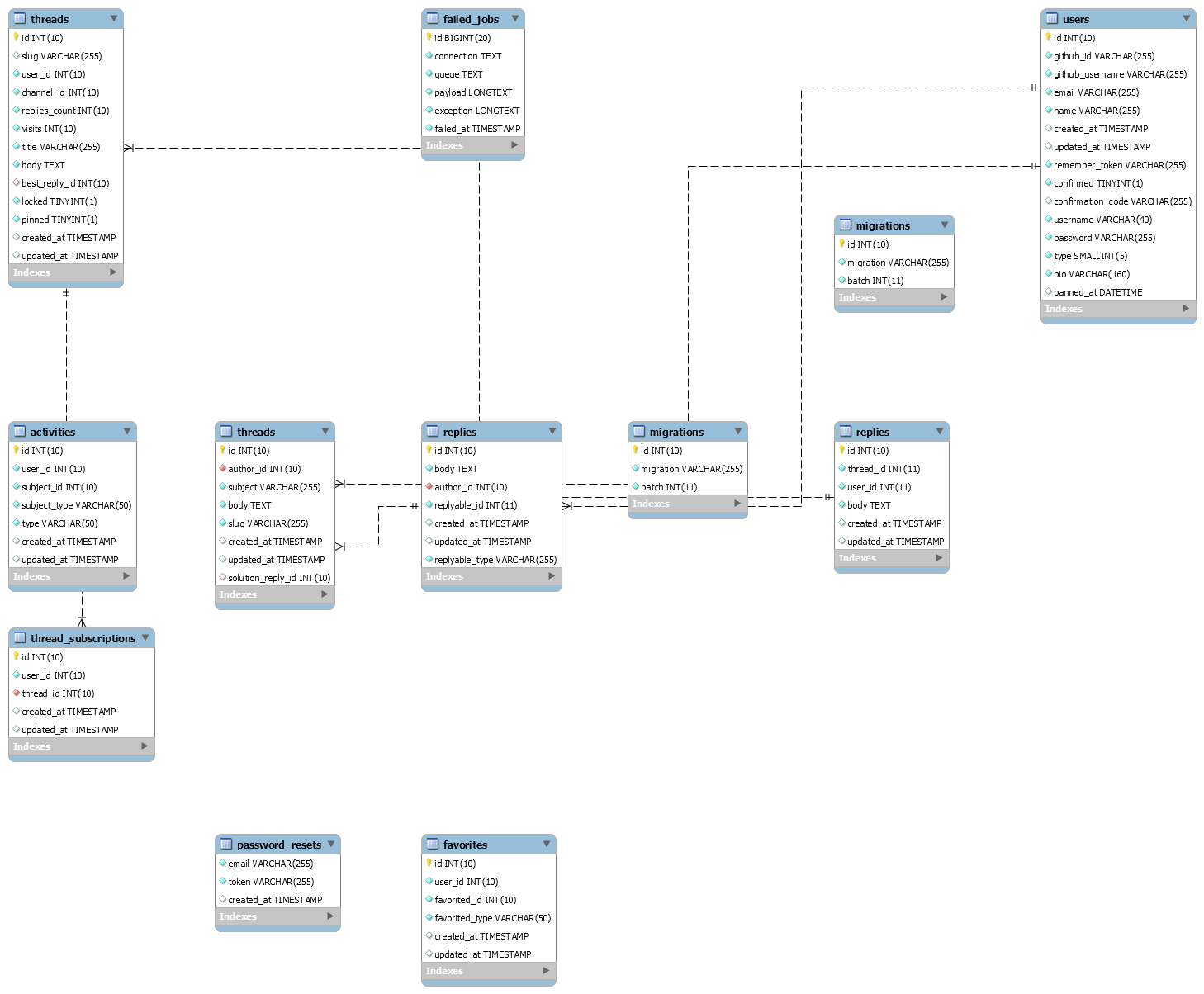
* Alle Eingabefelder werden längenbegrenzt.
* DB-Benutzer mit eingeschränkten Rechten
* Alle Werte werden mit der Validator Klasse validiert

# Technische Spezifikation

## Strukturierung des Systems



## Physikalische Datenbeschreibung (Datenstrukturtypen)



## Sicherheit

Um die Sicherheit der Applikation zu garantieren, läuft der Verkehr zwischen Frontend und Backend über HTTPS. Beim Anmelden, wird dem Benutzer ein eindeutiges und verschlüsseltes Token erstellt und zugewiesen. Es wird auf dem Client gespeichert und bei allen Interaktionen mit der Datenbank vom Backend abgeglichen. Passwörter werden "gehashed" in der Datenbank eingetragen.

## Anforderungen an die Entwicklungsumgebung

Für die Frontend Entwicklung benötigen wir einen Syntax Checker für den HTML, CSS und JS Code. Für das Backend brauchen wir ein IDE, welcher für uns optimal geeignet ist und wir den Code direkt mit einer Git Integration synchronisieren können. Ebenso sollte es eine eingebaute Konsole haben. Somit können wir den lokalen Webserver über diese starten.