

Dokumentation – Webprojekt Linkvel

Inhaltverzeichnis/Sitemap:

1. [Definition der Themen und Anforderung des Projektes](#)
2. [Erstellung BUC, SUC und Aktivitätsdiagramme](#)
3. [Erstellung der Tabellenmodelle und Implementierung der Datenbank](#)
4. [Erstellung von Mockups und Deklaration Design der Weboberfläche](#)
5. [Navigationsbar](#)
6. [Registrierung-Page](#)
7. [Login-Page](#)
8. [FAQ](#)
9. [UserProfileEdit-Page](#)
10. [Employee-Page](#)
11. [Suchleiste](#)
12. [Create Post Backend](#)
13. [Nutzerprofil](#)
14. [LandingPage – UserPost](#)
15. [User-Like](#)
16. [User Comment Post](#)
17. [Map-Location](#)
18. [Update User Data By Employee](#)
19. [Delete Post By Employee](#)
20. [Optionfeld Mitarbeiter Daten](#)
21. [Known Bugs](#)
22. [Verbesserungen](#)
23. [Lesson Learned](#)

Mitglied	Tätigkeiten
Antonia Geschke	Suchleiste Front- und Backend, CreatePost Backend und Impressum, Datenbankimplementierung
Helen Laible	Navigationsleiste, CreatePost Frontend, Dokumentation, Profilseite Backend
Nguyen Duc Duong	Registrierungsseite, LogInseite, FAQ, UserProfileEdit-Page, Mitarbeiterseite Backend (Account-/Contactdetails, PostID+Headline)
Tran Anh Hoang	LogIn Backend, Registrierungsseite Backend, Mitarbeiterseite Backend (Suche AccountID, Post löschen, Accountdetails updaten), LandingPage Front-und Backend, SocialMedia Interaktionen Front-und Backend, generelle Unterstützung bei allen Seiten

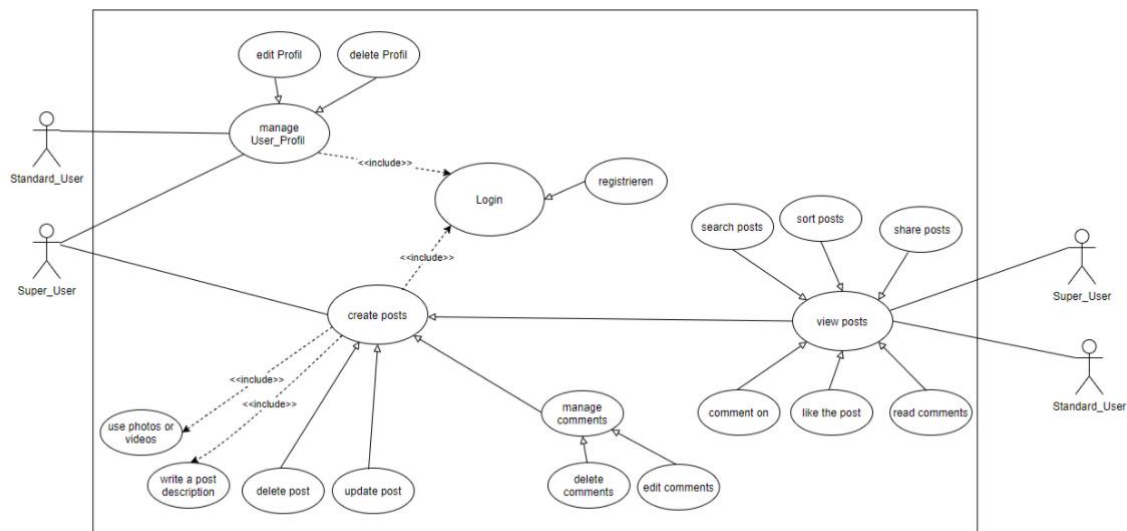
1. Definition der Themen und Anforderung des Projektes

Zu Beginn des Projektes haben wir uns einen ersten Eindruck der gestellten Anforderungen gemacht. Um in naher Zeit möglichst strukturiert zu können, haben wir die bevorstehenden Aufgaben eingeteilt und in verschiedenen Meilensteinen festgehalten. Wir haben des Weiteren geplant mindestens ein wöchentliches Treffen, je nach Pensum auch weitere zu vereinbaren, um ein aktives Verständnis für unsere Arbeit untereinander zu bekommen und um mögliche Probleme und unser weiteres Vorgehen zu besprechen. Dann kam es dann zu den ersten tatsächlichen Schritten und dem Start des Projektes. Ein Gruppenname war notwendig. Da es sich bei der Idee der Seite um einen Reiseblog handeln sollte, kombinierten wir letzten Endes die Wörter linked (vernetzt) und Travel (Reise), wodurch unser Projekt seinen Namen bekam. Linkvel.

2. Erstellung BUC, SUC und Aktivitätsdiagramme

Der nächste Schritt beinhaltete die Erstellung verschiedener Diagramme, die wir zu gleichen Teilen untereinander aufteilten. Darunter Business Use-Cases und System-Use-Cases und Flussdiagramme, die die Funktionen, Eigenschaften und Möglichkeiten der späteren Website darstellt. So entstanden Use-Cases zu den verschiedenen Rollen wie Super-User, User und Admin mit den jeweiligen Berechtigungen und Funktionen. Das System-Use-Case zeigt das nochmal im Detail. Das Flussdiagramm verdeutlicht den Vorgang innerhalb der Website beim Vorgang der Registrierung.

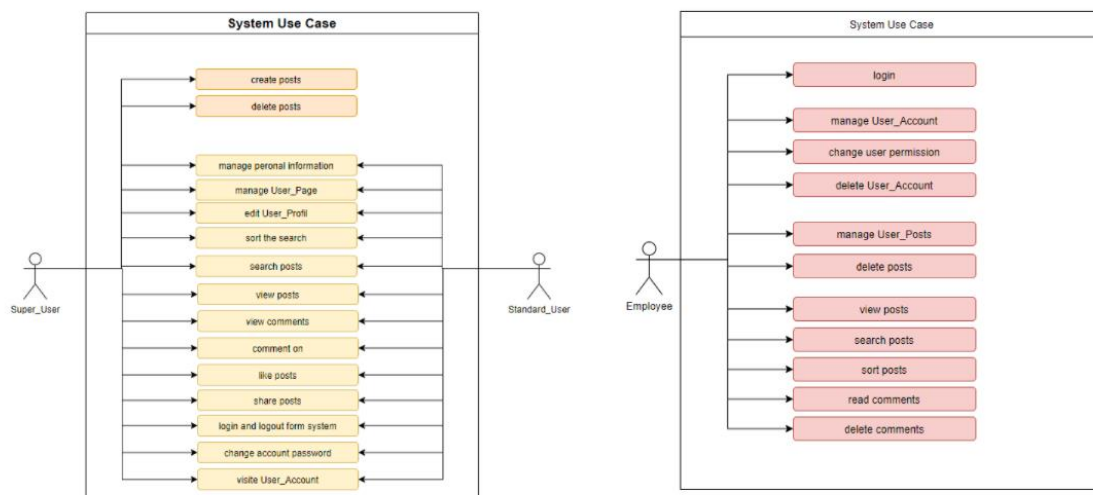
2.1 USE-CASE-Diagramm:



Der Actor ist entweder ein Standard_User oder ein Super User. Ein User kann nicht beides sein. Bevor der Standard_User oder der Super_User etwas machen kann, muss er sich in sein Account einloggen. Im Falle, dass der User neu ist und keinen vorhandenen Accountdaten hat, kann er sich registrieren. Nach der Registrierung wird der neue User als Standard_User gesetzt. Im Nachhinein kann der Administrator/Mitarbeiter den Standard_User zu einen Super_User machen. Nach dem Login haben beide die Funktionalität ihren User Profil zu managen, d.h. sie können ihre persönlichen Daten ändern, ihr Profil so gestalten und auch ihr User Account löschen, wenn sie das wollen. Der Unterschied zwischen den Standard_User

und den Super User ist es, dass der Super_User ein Post erstellen kann, welches alle besuchen und lesen können, die eingeloggt sind. Das wäre auch der einzige Unterschied. Um ein Post zu erstellen benötigt der Post unbedingt mindestens ein Bild oder ein Video und dazu eine passende Text/Beschreibung des Posts. Außerdem hat der Super_User vollen Zugriff auf die Kommentare seiner Posts. Er kann sowohl seine Kommentare noch dazu packen als auch auf andere Kommentare antworten. Falls es ein Kommentar gibt, dass dem Ersteller nicht gefällt, kann er sie löschen. Als Standard_User und Super_User kann man sich Posts anschauen die vom Ersteller (Super_User) freigegeben wurden anschauen. Beide haben die Möglichkeit nach deinem Post zu suchen und oder nach bsp. Kategorien, Erstelldatum, am meisten „geliket“, sortieren. Sie können dann die Posts lesen ein Kommentar zu dem Posts hinterlassen und auch teilen. Als Kommentar kann sowohl eine Frage als auch eine Antwort sein. Die Kommentare werden dann für alle eingeloggten Accounts sichtbar und so können andere User auf die jeweiligen Kommentare entweder fragen oder antworten.

2.2 SYSTEM-CASE-Diagramm:



Der Actor ist entweder ein Standard_User oder ein Super_User. Ein User kann nicht beides sein. Bevor der Standard_User oder der Super_User etwas machen kann, muss er sich in sein Account einloggen.

Im Falle, dass der User neu ist und keinen vorhandenen Accountdaten hat, kann er sich registrieren. Nach dem Login haben beide die Funktionalität ihren User_Profil zu managen, d.h. sie können ihre persönlichen Daten ändern, ihr Profil so gestalten und auch ihr User_Account löschen, wenn sie das wollen.

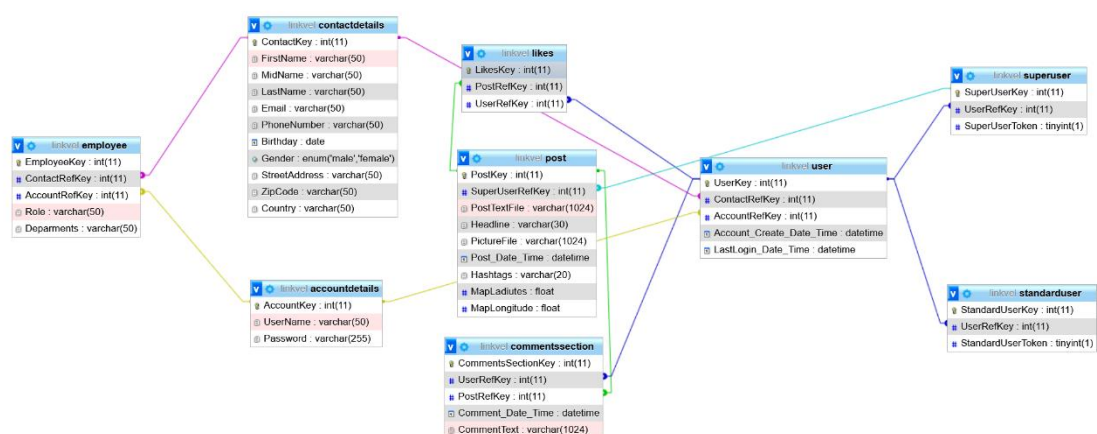
Der Unterschied zwischen den Standard_User und den Super_User ist es, dass der Super_User ein Post erstellen kann, welches alle besuchen und lesen können, die eingeloggt sind. Das wäre auch der einzige Unterschied. Um ein Post zu erstellen benötigt der Post unbedingt mindestens ein Bild oder ein Video und dazu eine passende Text/Beschreibung des Posts. Außerdem hat der Super_User vollen Zugriff auf die Kommentare seiner Posts. Er kann sowohl seine Kommentare noch dazu packen als auch auf andere Kommentare antworten. Falls es ein Kommentar gibt, dass dem Ersteller nicht gefällt, kann er sie löschen. Als Standard_User und Super_User kann man sich Posts anschauen die vom Ersteller (Super_User) freigegeben wurden anschauen. Beide haben die Möglichkeit nach deinem Post zu suchen und oder nach bsp. Kategorien, Erstelldatum, am meisten „geliket“, sortieren. Sie können dann die Posts lesen ein Kommentar zu dem Posts hinterlassen und auch teilen. Als

Kommentar kann sowohl eine Frage als auch eine Antwort sein. Die Kommentare werden dann für alle eingeloggtten Accounts sichtbar und so können andere User auf die jeweiligen Kommentare entweder fragen oder antworten.

3. Erstellung der Tabellenmodelle und Implementierung der Datenbank

Infolgedessen kam es zur Grundlage der Datenbank. Die Erstellung des ER-Modells und des darauf aufbauenden Tabellen-Modells. Die Datenbank sollte Datensätze über User, der in Standard-User und Super-User unterteilt ist, enthalten sowie Mitarbeiter. Über beide sollten Datensätze wie Adresse und Account-Details und Kontaktdaten gespeichert sein. Des Weiteren werden Posts, Likes und Kommentare gespeichert. Das erstellte ER-Modell war die Grundlage für das darauffolgende Tabellen-Modell, das alle Bereiche unserer zu erstellender Datenbank beinhaltet. In einem unserer Meetings im November fand eine grobe Ideensammlung für die finale Datenbank statt, dessen Implementierung kurz darauffolgte.

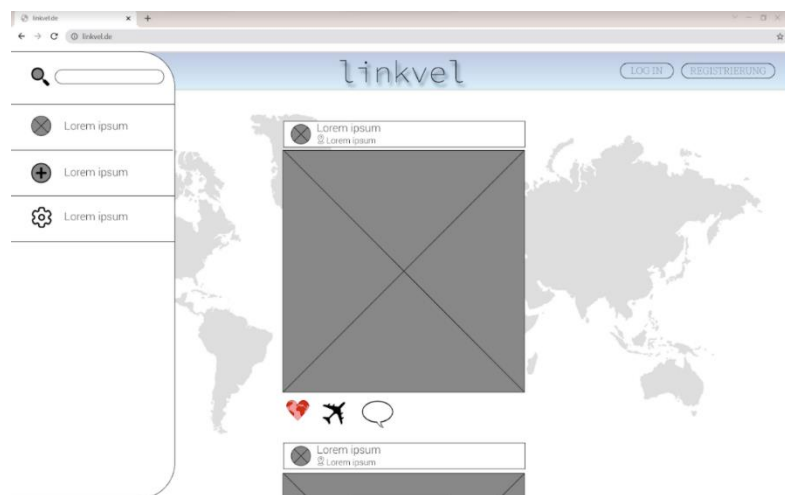
ER--Modell & Tabellen-Modell:



4. Erstellung von Mockups und Deklaration Design der Weboberfläche

Nahezu zeitgleich gab es erste Überlegungen wie die Seiten strukturiert und anschaulich umgesetzt werden können. Mit Hilfe von Mockups kamen wir schnell auf eine zufriedenstellende Möglichkeit das Optische mit den Funktionalitäten und Anforderungen zu kombinieren die einerseits gestellt worden sind und die wir selbst gerne hätten. Im nächsten Meeting einigten wir uns darauf das Design durch CSS, und die nice-to-have Funktionen hintenanzustellen und uns zunächst auf die wichtigsten Sachen zu konzentrieren.

Mock-Up Linkvel:



5. Navigationsbar

Die Navigationsleiste stellte uns im Hinblick auf die Responsivität der Seiten kurzzeitig vor einige Probleme. Um die Bedienung der Website möglichst einfach zu machen wollten wir auf ein Burger-Menü verzichten. Die Idee einer normalen Navigationsleiste in Kombination, diese ab einer bestimmten Breite zum Burger-Menü zusammenzufassen, löste das Problem. Die Navigationsleiste umfasst die Funktionen entweder durch das Anklicken der "Startseite" oder "Linkvel" jeder Zeit auf die Startseite (LandingPge) zurückzukehren. Unter "Mein Konto" erscheint ein Dropdown Menu, dass einem die Optionen gibt sein Profil anzeigen zu lassen oder es zu bearbeiten. Sollte der jeweilige Besucher zum Zeitpunkt nicht angemeldet sein, wird auch eine Fehlerseite weitergeleitet, die ihm die Option gibt, auf die LogIn-Seite zu kommen. Die Ansicht der Navigationsleiste orientiert sich an dem jeweiligen Zustand der Session. Ist man angemeldet so zeigt sie den Option Logout statt Login an. Ist man nicht angemeldet eben Login. Ist ein Admin angemeldet entfällt das Dropdown-Menu. Um den Nutzern oder Besuchern ein Gefühl der Persönlichkeit zu vermitteln, wird je nachdem ob man angemeldet ist mit Hallo du oder Hallo (Nutzername) auf der Seite willkommen geheißen. In der minimierten Ansicht enthält das Burgermenu die identischen Funktionen, jedoch entfällt das Logo und die Willkommensworte ab einer bestimmten Breite. Jetzt kann der Nutzer nur noch über "Startseite" auf die Landingpage zurückkehren. Je nachdem welcher User sich eingeloggt hat gibt es verschiedene Varianten der Navigationsbar.

Linkvel
Hallo, du!



STARTSEITE

MEIN KONTO

LOGIN



STARTSEITE

MEIN KONTO

LOGIN

6. Registrierung-Page

1.Phase: Registrierung

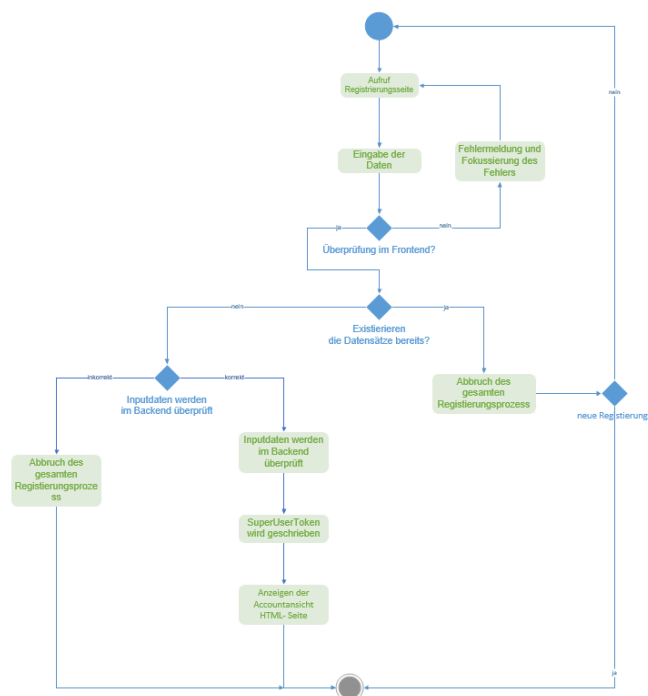
Der User registriert sich, indem er alle Pflichtfelder ausfüllt. Die Pflichtfelder sind Vorname, Nachname, Adresse, Postleitzahl, Land (woher der User kommt), Username, E-Mail, Geburtsdatum, Passwort und Handynummer. Ob die Daten, die der User eingibt, auch in der realen Welt existieren, kann man nicht überprüfen. Dies kann zur schlechten Datenqualität führen. Dieses Problem gilt auch auf andere Seiten wie UserProfileEdit, wo Daten manipuliert werden.

Jedoch ist es möglich, dass man die Syntax bestimmte Zeichenketten und auch das Alter der Personen überprüfen kann, um sich Registrieren zu können. Das wird auch bei unserer Registrierung realisiert.

Unsere Überprüfung im Hintergrund wird wie folgt durchgeführt:

Im ersten Schritt wird zunächst im Frontend die Telefonnummer als auch die Email-Adresse auf einen spezifischen Syntax überprüft. Ist dieser nicht korrekt wird ein Javascript Alerttextbox ausgelöst, der den User auf seinen Fehler aufmerksam macht. Desweiterem wird auch das Alter überprüft, ist der User unter 16 Jahre alt, wird auch eine Fehlermeldung ausgegeben. Falls alles korrekt eingegeben und der Haken bei „Ich akzeptiere die AGB und Datenschutz“ gesetzt wurde (ist auch Pflichtfeld), wird im zweiten Teil die Daten aus den Formulartextboxen von PHP über die Postmethode abgeholt und auch im Backend überprüft. Nun können zwei Szenarien eintreten, dieses wird in den beiden Fällen im folgenden Bereich beschrieben.

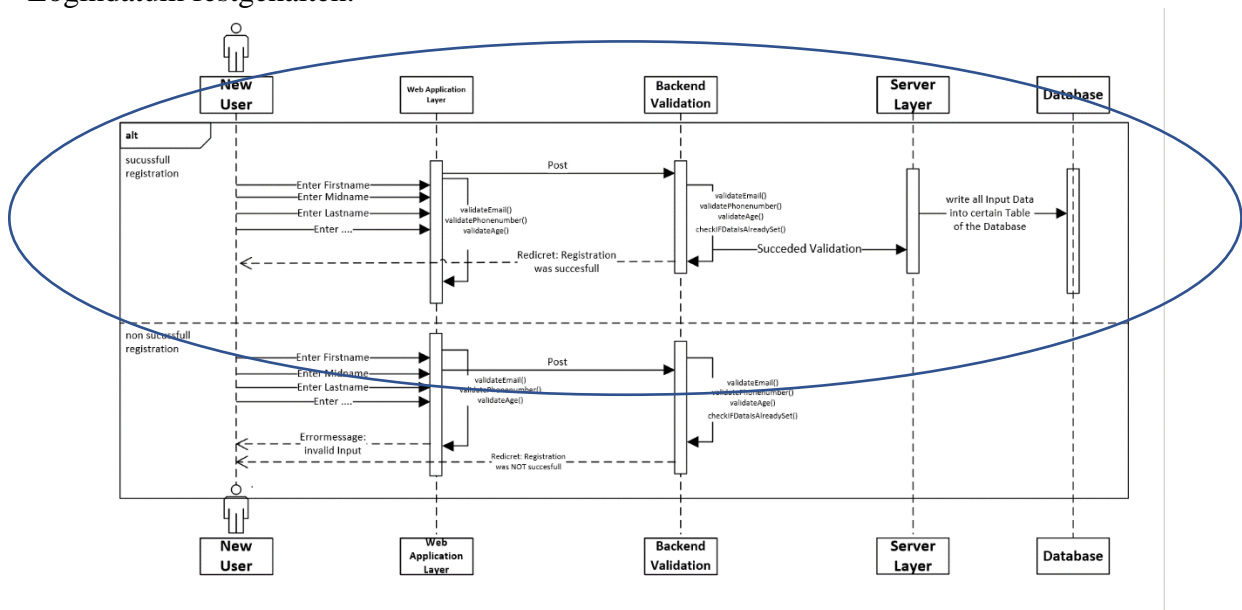
Hier im Flussdiagramm zusammengefasst:



1. Fall – Daten werden korrekt angegeben

Hat der User korrekte Daten eingegeben, so werden diese in die Datenbank eingetragen und abgespeichert. Nach der erfolgreichen Registration wird der User auf eine Bestätigungsseite geleitet, wo dem User angezeigt wird, dass er sich ein Account auf der Linkvel-Webseite erstellt hat, Danach kann er auf „Beginne deine virtuelle Reise“ klicken, dann wird der User auf die Landing-Page (Startseite) geleitet, jedoch ist er noch nicht angemeldet und muss die noch tun bevor seine richtig „virtuelle Reise“ beginnen kann (siehe Registrierung-Page 1.Phase 6.Bild).

Dabei wird im Backend zusätzliche der User auf einen Standard User und auch sein Account_Create_Time_Date und LastLog_In_Date_Time gesetzt. Diese sind jeweils Daten, welche die Rolle des Users angibt und seinen Account Registrierungsdatum sowie Logindatum festgehalten.



The registration form includes the following fields and elements:

- Buttons: Login, Registration
- Social media icons: Facebook, Twitter, Instagram
- Form fields:
 - Geschlecht
 - Vorname *
 - Mittlename
 - Nachname *
 - Straße und Hausnummer *
 - Postleitzahl *
 - Land *
 - Username *
 - Email *
 - Geburtsdatum *
 - Passwort *
 - Handynummer *
- *Pflichtfelder (Mandatory fields)
- Checkbox: Ich akzeptiere die AGB und Datenschutzz
- Register button (highlighted with a green arrow)



Beginne deine wunderbare Reise!

Willkommen auf Linkvel! Wir hoffen,
du hast einen wunderschönen Tag!

WUNDERSCHÖNES LISSABON

Von: Pflor
Im letzten Sommerbesuche ich die Stadt
Lissabon. Eines der schönsten Städte in Europa!



👤 1 💬 0 📍

Create
New Post

FAQ

Mein
Account

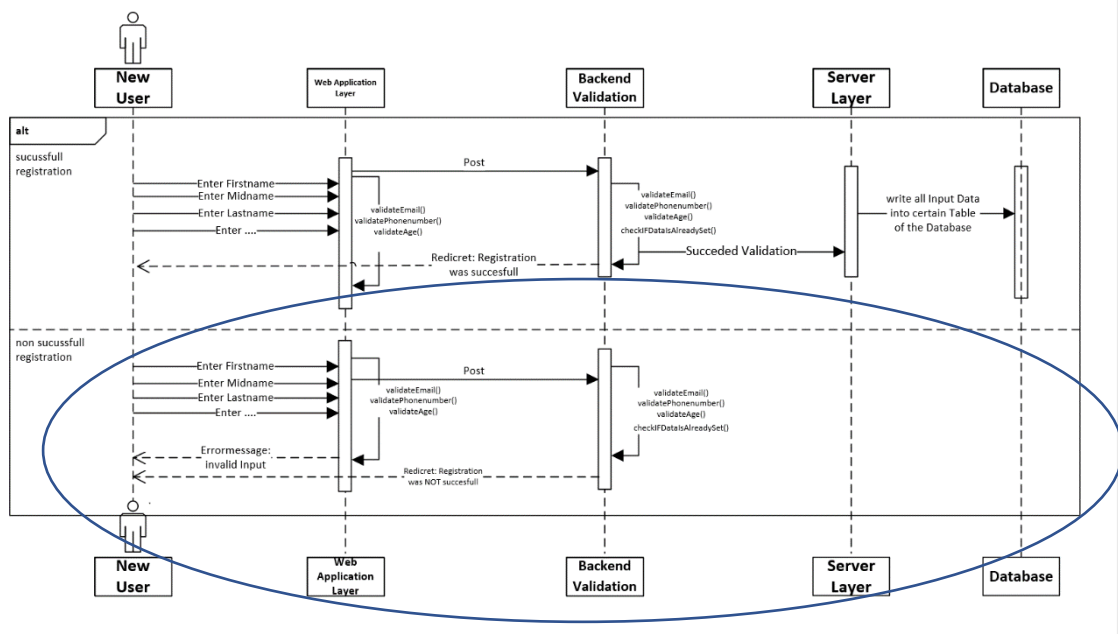
Logout



2. Fall

Pflichtfelder wie Email, Handynummer oder/und Alter werden geprüft auf die richtige Syntax überprüft, wie bei Email muss ein „@“ in dem Feld enthalten sein oder das bei Handynummer, dass nicht weniger als 10 Zahlen eingetragen werden darf und zusätzlich wird das Alter des Users überprüft. Der User muss dabei älter als 16 Jahre Alt sein. Außerdem wird auch im Backend überprüft ob der User sich bereits bei linkvel registiert hat. Dabei wird ein abgleich der Emailadresse und der Username in der Datenbank mit den eingegebenen Userdaten. Sind diese bereits in der Datenbank hinterlegt, so kann der User sich nicht noch einmal registrieren. Wurden die Daten falsch eingegeben, so kann der User sich auch nicht registrieren. Der Registrierung wird unterbrochen und der User wird auch eine Fehlermelde Seite weitergeleitet.

Fehlermeldung Registrierung:



The screenshot shows a registration form overlaid on a background image of an airplane flying over a sunset sky. The form has a 'Login' button and a 'Registration' button at the top. Below the buttons are social media icons for Facebook, Twitter, and Instagram. The form fields are: Geschlecht, Vorname *, Mittleiname, Nachname *, Straße und Hausnummer *, Postleitzahl *, Land*, Username *, Email *, Geburtsdatum *, Passwort *, and Handynummer *. There is a section for 'Pflichtfelder' (required fields) with a green checkmark and the text 'Ich akzeptiere die AGB und Datenschutz'. At the bottom is a 'Registrieren' button.

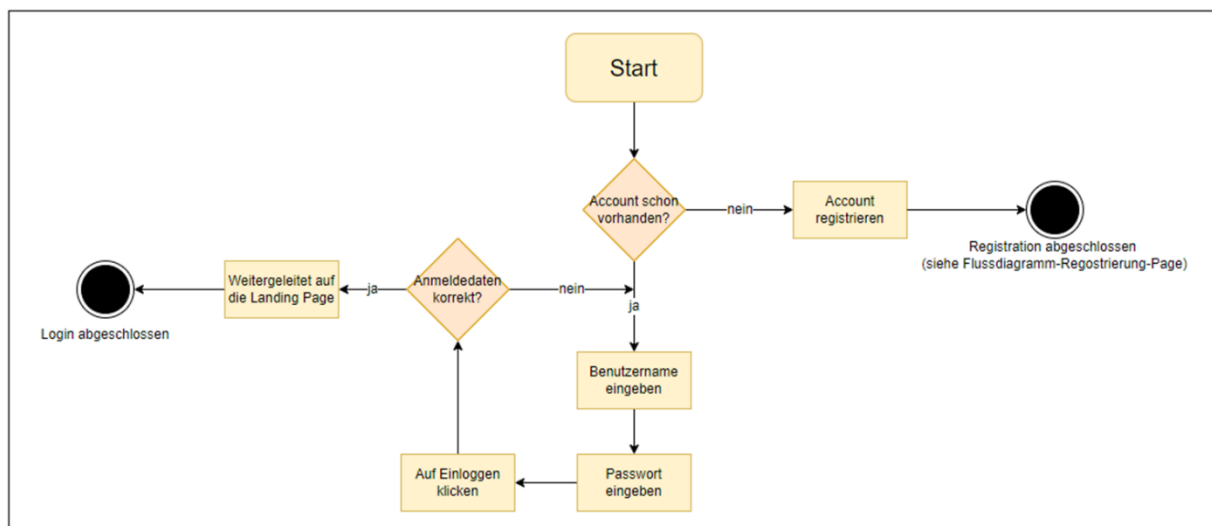
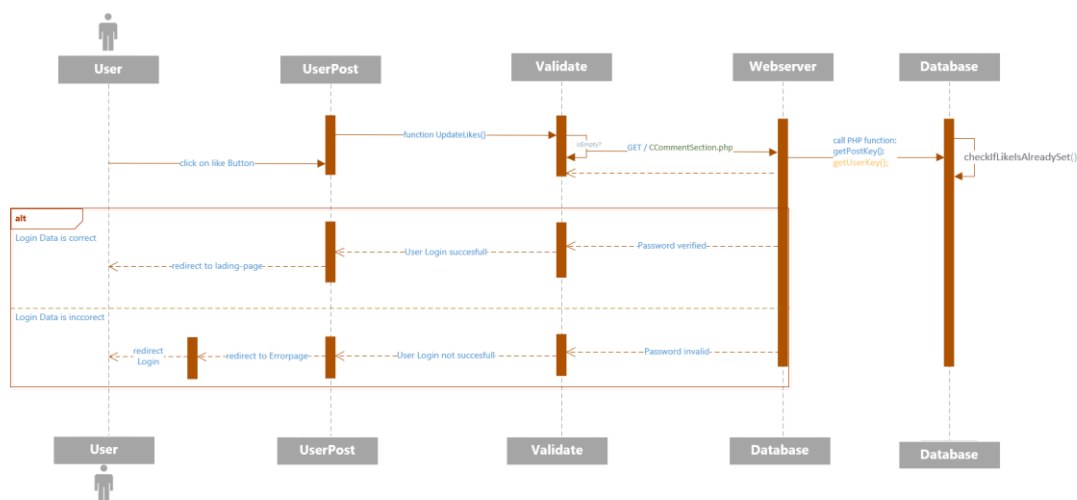
Dieser User existiert bereits. Bitte melde dich bei unseren Mitarbeitern für weiteren Support!

[Zurück zur Startseite](#)

7. Login-Page

1. Phase: Login

Der User muss sich einloggen, bevor er irgendetwas machen kann. Falls der User noch kein Account hat, kann er auf Registration klicken und er wechselt zur Registration (siehe Registrierungs-Page). Nach der Registration kann er sich dann mit seinen Anmeldedaten (Username & Password) einloggen, indem er dann auf den Login-Knopf klickt. Im Hintergrund wird in der Datenbank überprüft, ob es übereinstimmende Anmeldedaten in der Datenbank gibt oder nicht.

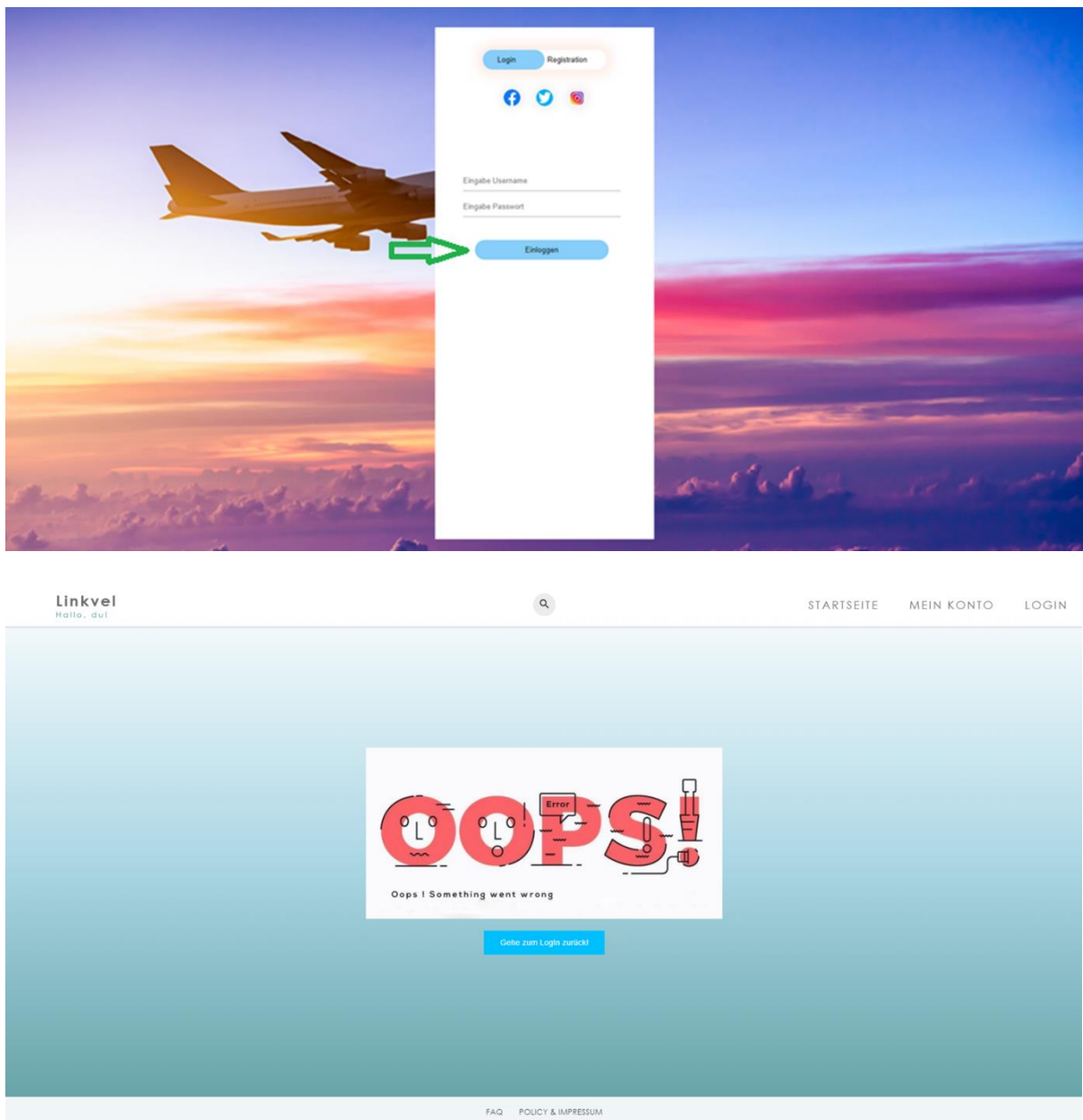


1. Fall

Daten stimmen einem Account überein: Dann wird der User zu der Landing Page (Hauptseite) weitergeleitet. Im Backend wird außerdem das die Anmeldezeit und -datum der Anmeldung des User in der Datenbank überschrieben

2. Fall

Daten stimmen keinen Account überein: Dann dem User auf eine Error-Page weitergeleitet, dass seine Anmeldedaten keinem Account übereinstimmt und er kann dann die Anmeldung erneut versuchen.



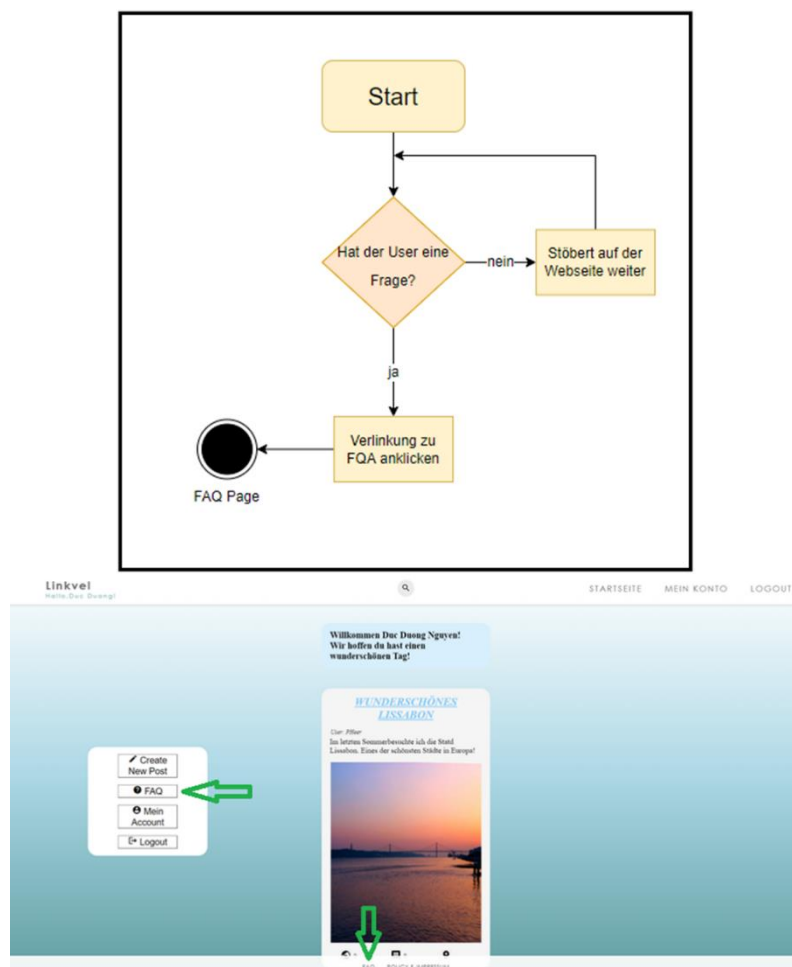
8. FAQ-Page

1. Phase: Login

Siehe Login-Page => 1.Phase

2. Phase: Der User hat eine Frage

Der User hat eine offene Frage, die er sich nicht beantworten kann. Um seine Frage/Fragen zu beantworten sollte er sich auf die FAQ-Seite begeben. Dazu klickt er auf eins der beiden maskierten Verlinkungen (siehe FAQ Page 2.Phase-2.Bild), danach wird er zu der FAQ Page weitergeleitet. => 3.Phase



3. Phase: Die Frage vom User wird/wurde beantwortet

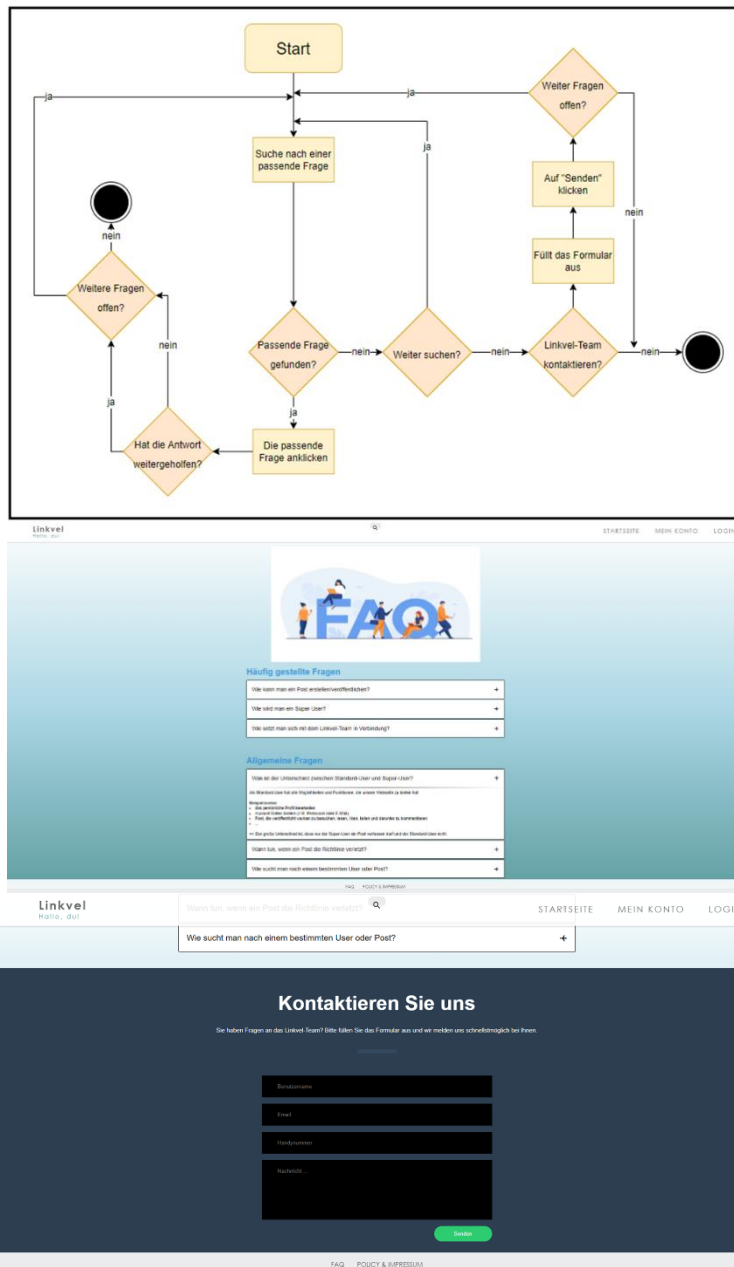
Die FAQ-Seite ist dazu da, damit der User auf seine Fragen eine Schnelle Antwort bekommen. Auf der FAQ-Seite gelandet, kann der User ein paar vorgefertigte Fragen von dem Linkvel-Team durchlesen und schauen, ob es eine Frage gibt, die der vom User ähnelt. Um die Antworten zu den Fragen zu sehen, muss der User auf die Frage anklicken und es erscheint die Antwort direkt darunter.

1.Fall – Passende Antwort gefunden:

Prozess beendet / Wenn der User noch offene Fragen hat, kann er sich noch rumschauen oder es gibt noch andere Anliegen ... => 2.Fall

2.Fall – Keine passende Antwort gefunden:

Im unteren Teil der FAQ-Seite befindet sich ein Kontaktbereich, wo der User das Formular ausfüllt und dann auf „Senden“ klickt.



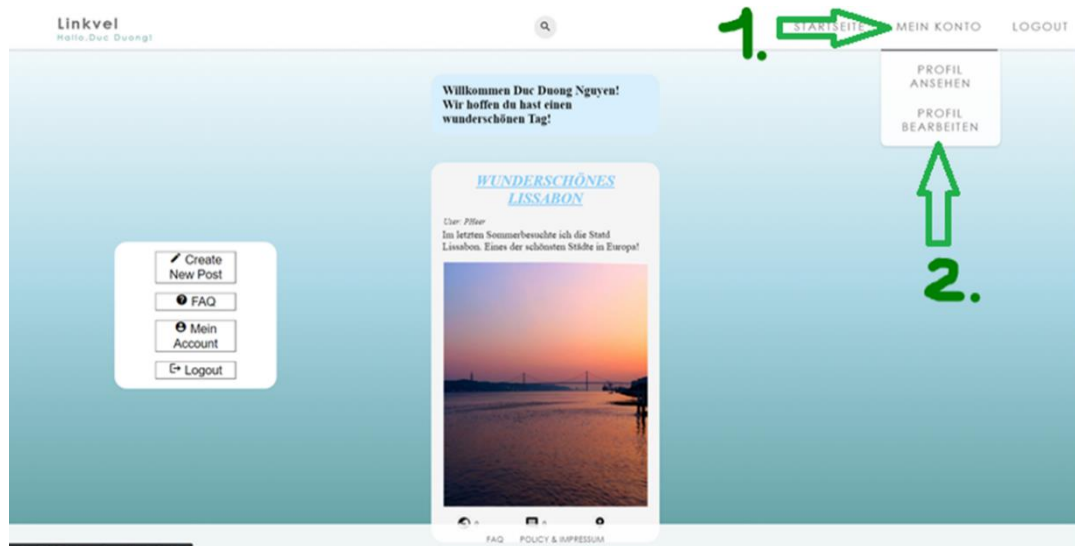
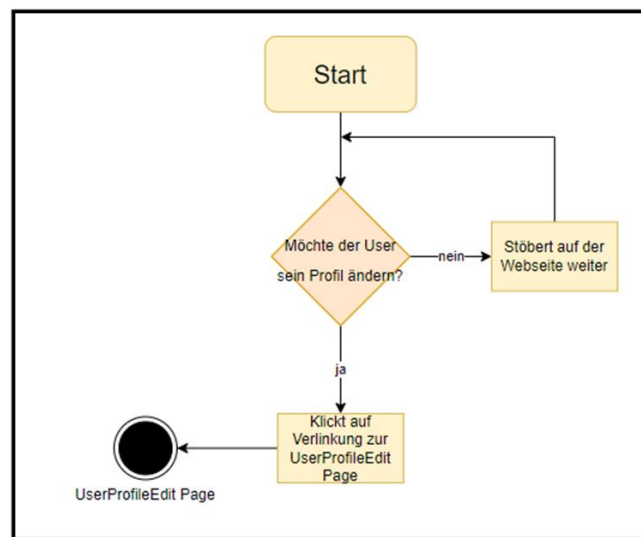
9. UserProfileEdit-Page

1.Phase: Login

Siehe Login-Page => 1.Phase

2.Phase: Der User möchte sein Profildaten ändern

Der User hat Daten, die er auf sein Profil ändern möchte. Um seine persönlichen- also auch seine Accountdaten ändern zu können, folgt der User die beiden Steps, die auf dem Bild markiert sind (siehe UserProfileEdit Page 2.Phase-2.Bild) danach wird er zu der UserProfileEdit Page weitergeleitet. => 3.Phase

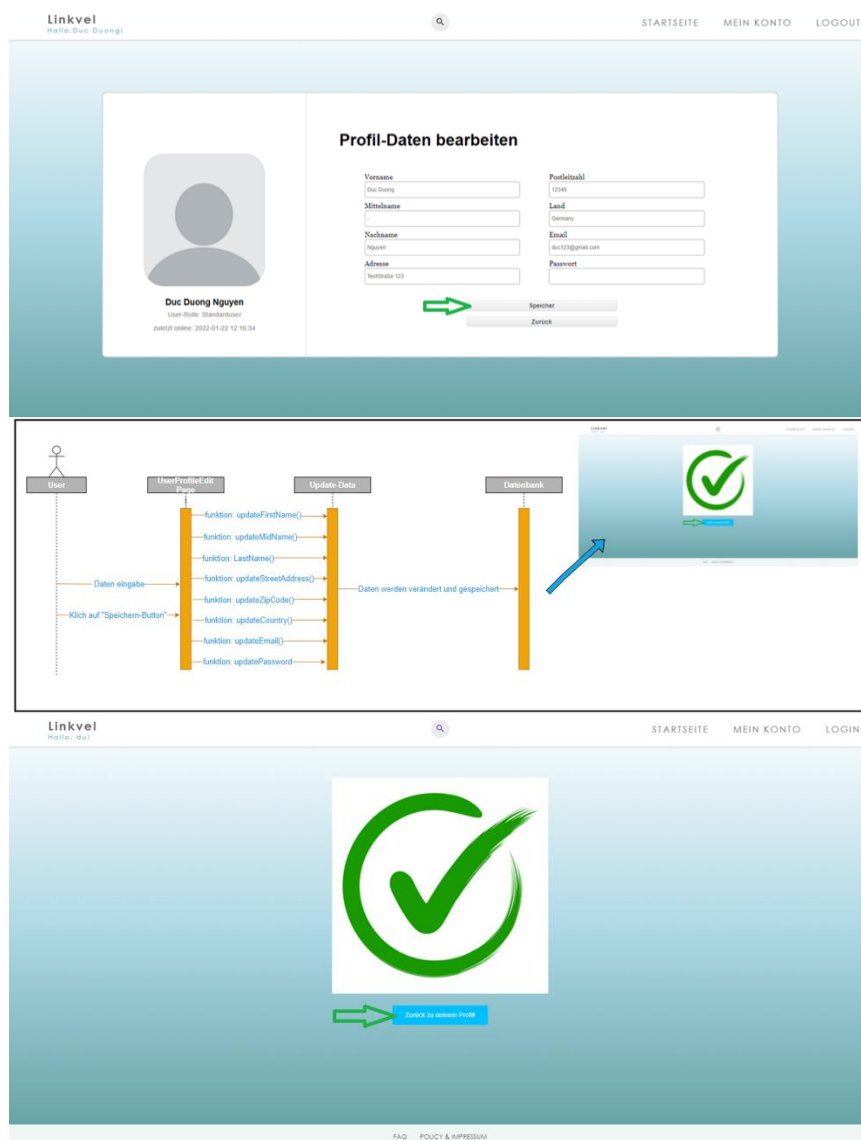


3. Phase: Userdaten werden geändert und gespeichert

Auf der UserProfileEdit-Seite angekommen, kann der User seine persönlichen- also auch seine Accountdaten ändern.

1. Fall – User möchte seine Daten ändern:

Daten, die der User ändern kann, sind seine Vor-, Mittel-, Nachname, Adresse, Postleitzahl, Land, E-Mail und Passwort. Andere Daten wie Geburtstag, Geschlecht, Benutzername und Telefonnummer sind nicht erlaubt. Am Anfang stehen in den Felder, wo der User später seine Änderungen einträgt, noch die Daten, die noch bis zur Änderung gelten in einem Label. Um die Daten von dem User zu ändern, füllt er die Felder aus, die dem User zutreffen. Der User muss nicht alle Felder ausfüllen damit er die Änderungen abspeichern kann. Die Felder, die der User nicht ändert, bleiben die vorigen Daten bestehen. Für die Abspeicherung klickt der User auf den Speicher-Button, dann wird er auf eine Weitere Seite geleitet, wo der User eine Bestätigung bekommt, dass seine Daten erfolgreich geändert wurden (sieht UserProfileEdit-Page 3.Phase 3.Bild), dann kann der User auf den Zurück-zu-deinem-Profil-Button klicken, dann wird er zu sein Profil Page weitergeleitet.



2. Fall – User möchte seine Daten nicht ändern:

Um seine Daten nicht mehr zu ändern, klickt der User auf den Zurück-Button, dann wird der User zurück auf sein Profil Page geleitet.

Linkvel
Hallo, Duc Duong!

STARTSEITE MEIN KONTO LOGOUT

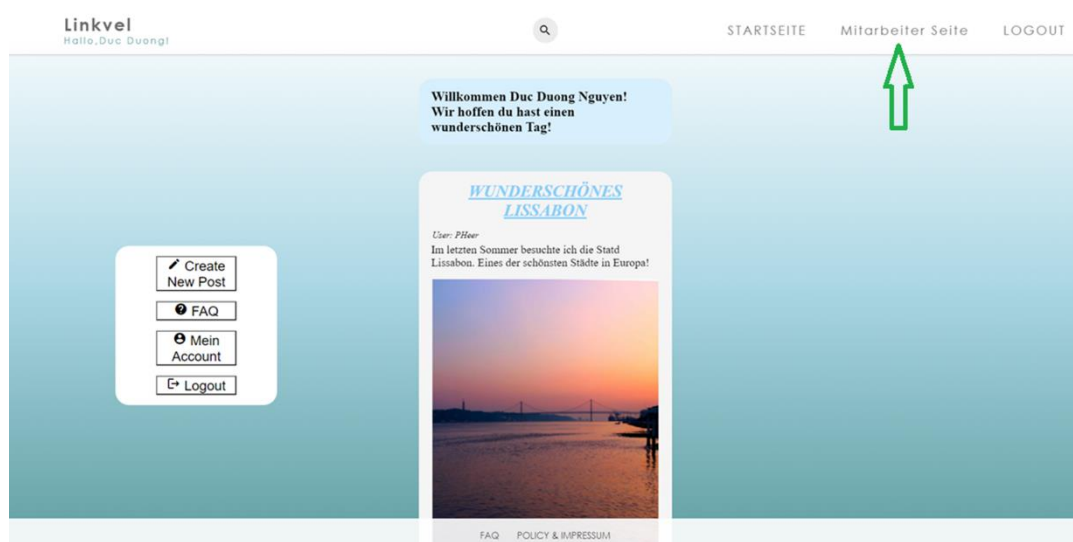
Profil-Daten bearbeiten

Vorname: Duc Duong
Mittelname: -
Nachname: Nguyen
Adresse: Teststraße 123
Postleitzahl: 12345
Land: Germany
Email: duc123@gmail.com
Passwort:
Speicher
Zurück

10. Employee-Page

1. Phase: Login

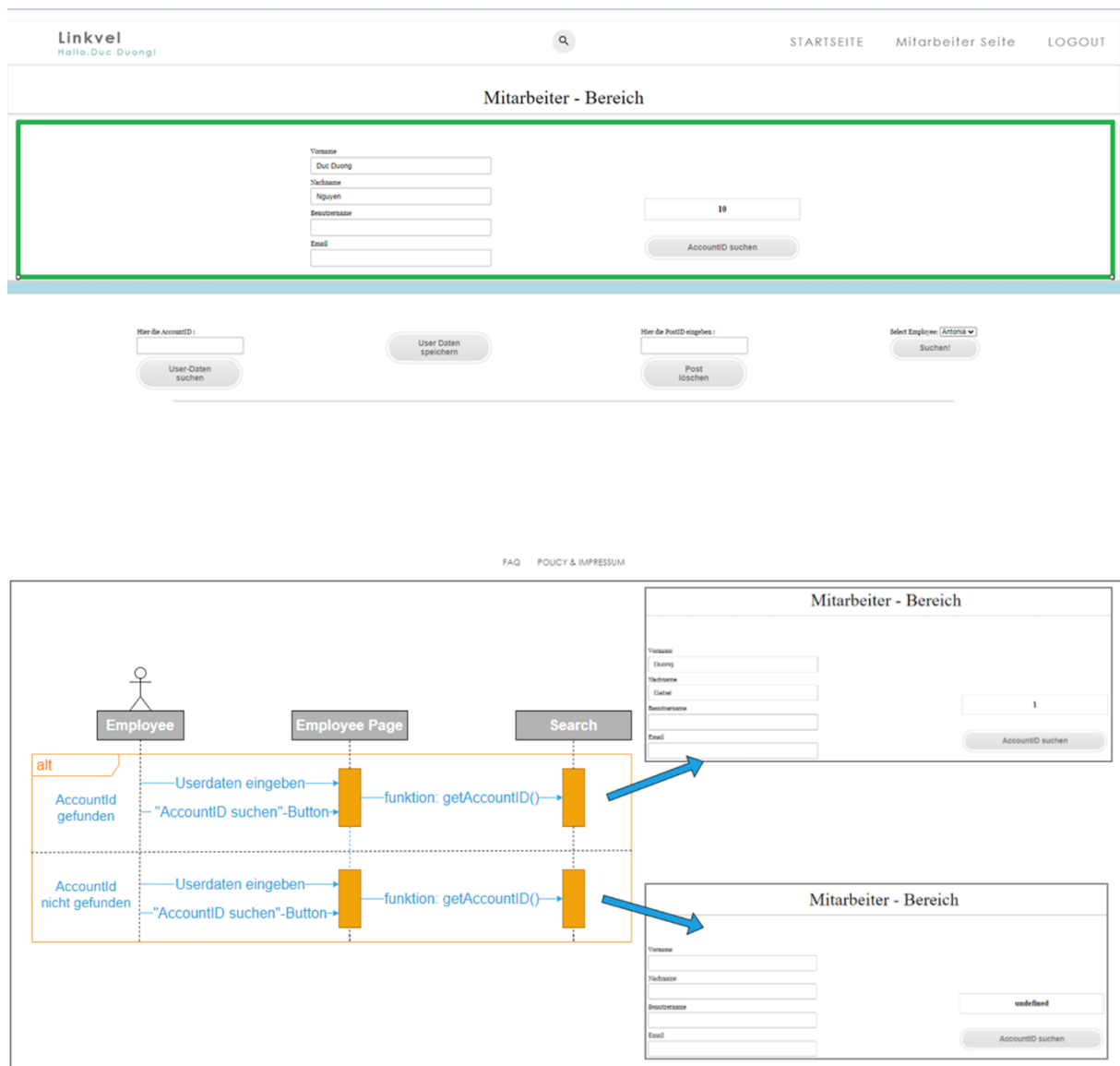
Siehe Login-Page => 1.Phase Der Employee meldet sich wie ein normaler User an, aber der Unterschied ist, dass der Employee in der Navigationsbar sein extra Feld hat, welches ihn zu den Mitarbeiterbereich führt.



2. Phase: Als Mitarbeiter im Mitarbeiter-Bereich

1. Bereich: AccountId suchen

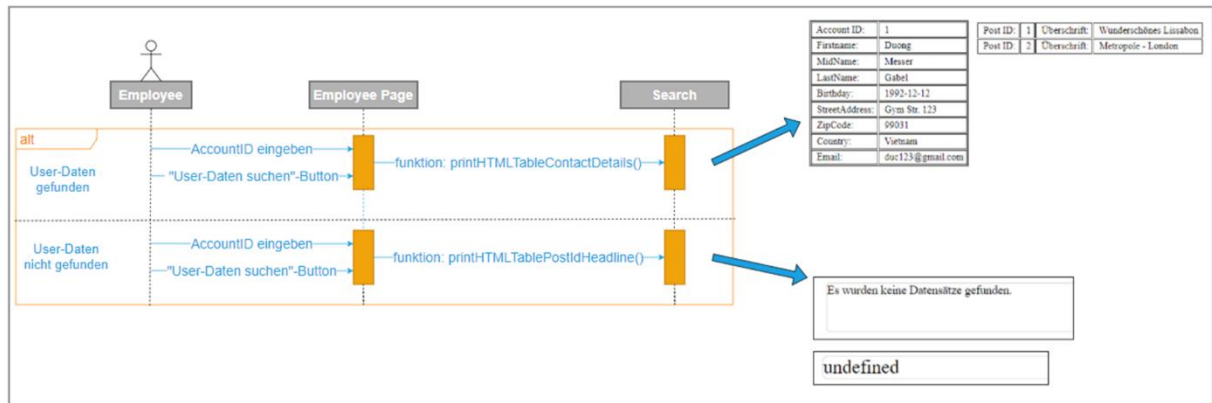
Im ersten Bereich kann der Mitarbeiter den AccountId von einem User herausfinden, in dem der Mitarbeiter der Vor- und Nachname oder Benutzername und Email des gesuchten Users richtig ein eingibt. Dann erscheint im AccountID-Feld die Id vom gesuchten User oder es wird das Wort „undefined“ ausgegeben, wenn zu den Daten kein User gefunden wurde.



2. Bereich: User-Daten anzeigen

Wenn der Mitarbeiter den AccountId vom gesuchten User nicht kennt, muss er im 1. Bereich zurück und dann den AccountId erst suchen. Wenn er den AccountId kennt, trägt er den AccountId in das Feld „Hier die AccountID:“ und drückt dann auf User-Daten-Button suchen. Wenn keine Eingabe vorhanden ist und den der Button gedrückt wird, erscheint ein „undefined“ (siehe Employee-Page 2.Phase 6.Bild) oder „Es wurden keine Daten gefunden“ (siehe Employee-Page 2.Phase 7.Bild). Falls in dem Feld ein AccountId drinsteht, wird im Hintergrund geprüft, ob es in der Datenbank ein AccountId gibt der mit den AccountId übereinstimmt, der als Mitarbeiter eingegeben wurde. Bei einer Übereinstimmung der beiden

AccountID wird einige Daten des gefundenen User in einer Tabelle ausgegeben. Die ausgegebenen Daten als Standard-User sind AccountID, Vorname, Mittelname, Nachname, Geburtstag, Adresse, Postleitzahl, Land, E-Mail und StandardUserToken/SuperUserToken. Dann wird es noch eine Tabelle geben, wenn der User ein SuperUser ist. Die Tabelle enthält die PostID und die Überschrift und ist dafür geeignet, damit der Mitarbeiter bestimmte Post löschen kann.



Vorname

Nachname

Benutzername

AccountID

Email

AccountID suchen

Hier die AccountID:

User Daten speichern

Hier die PostID eingeben:

Post löschen

User-Daten suchen

Select Employee: Antonia

Account ID	1
Firstname	Paulina
MidName	
LastName	Heer
Birthday	1992-12-12
StreetAddress	Am Standpark 2
ZipCode	99012
Country	Germany
Email	pherr@gmail.com

Post ID	1	Überschrift	Wunderschönes Lissabon
---------	---	-------------	------------------------

Hier die AccountID:

User Daten speichern

Hier die PostID eingeben:

Post löschen

User-Daten suchen

Select Employee: Antonia

undefined

Hier die AccountID:

User Daten speichern

Hier die PostID eingeben:

Post löschen

User-Daten suchen

Select Employee: Antonia

Es wurden keine Datensätze gefunden.

11. Suchleiste

Der User gibt seinen Suchbegriff in die Suchleiste der Navigationsbar ein und bestätigt die Eingabe.

Linkvel
Hallo, du!

SUCHE



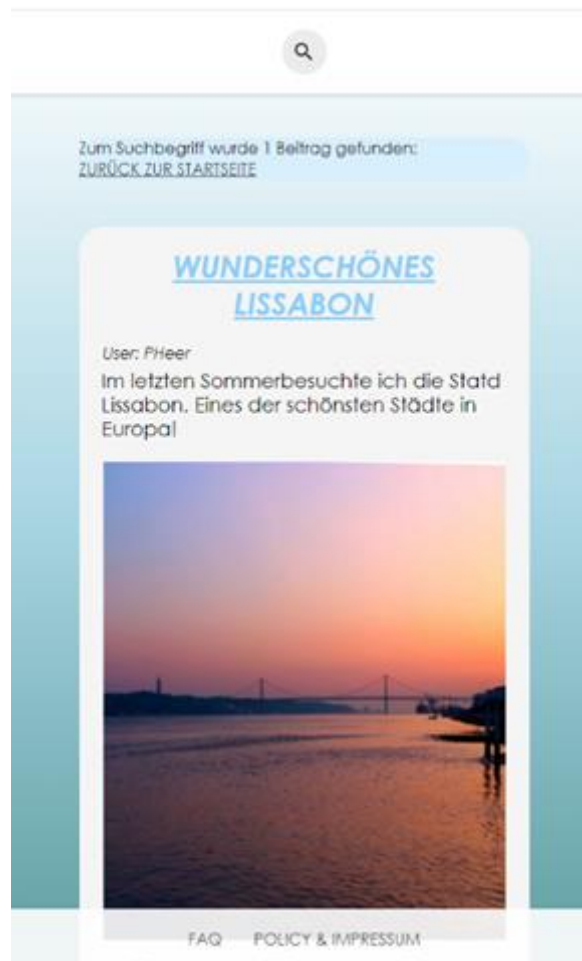
STARTSEITE

MEIN KONTO

LOGIN

Intern speichert die „Search.php“ Seite den Suchbegriff in einer Variablen und ruft mit dem Begriff die Funktion Search (String \$Statement) der Klasse „CSearch“ auf. Innerhalb dieser Funktion wird geprüft, ob es sich bei dem Suchbegriff um eine Hashtag oder eine Überschrift handelt und abhängig davon die Funktion SearchHashtag(String \$Statement) oder SearchHeadline(String \$Statement) aufgerufen. Innerhalb der Funktionen wird in der Datenbank mittels einer SQL-Abfrage nach dem Begriff gesucht und die Ergebnisse in einem Array gespeichert und zurückgegeben. Währenddessen werden die Ergebnisse in einer neuen Variablen bei „Search.php“ gespeichert.

Außerdem wird dem Nutzer ein Text angezeigt, der ihm mitteilt, wie viele Beiträge gefunden wurden.

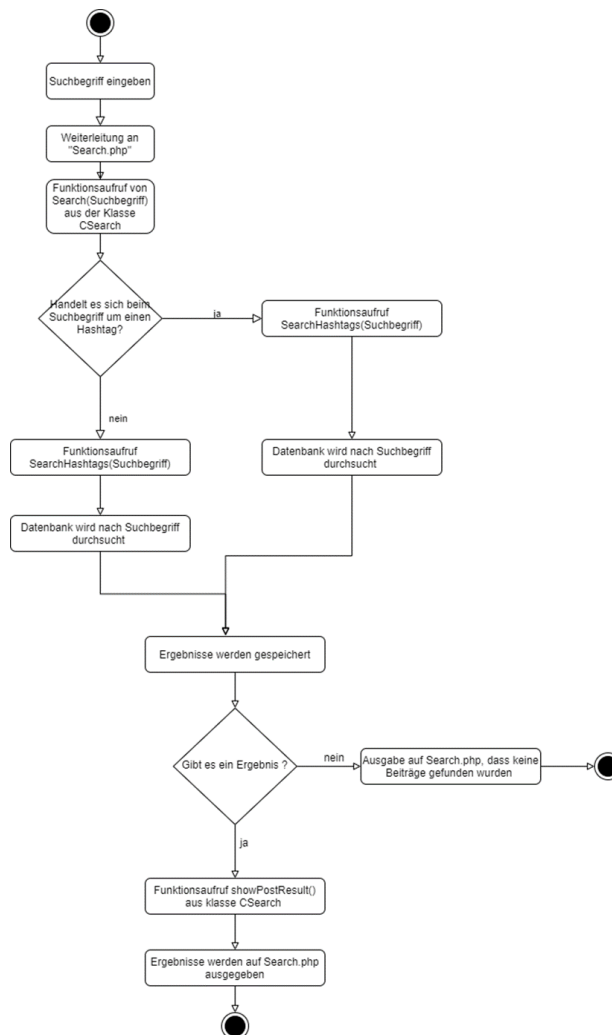


Falls keine Ergebnisse gefunden wurden, wird dem Nutzer ein Text ausgegeben, der darauf hinweist, dass keine Beiträge gefunden wurden und die Möglichkeit bietet zur Startseite zurückzugehen. Wenn die SQL-Abfrage zu Ergebnissen führt wird die Funktion showPostResult() aufgerufen. Dies führt dazu, dass alle Posts, die zu dem Suchbegriff passen ausgegeben.



Zu dem Suchbegriff konnte kein Beitrag gefunden werden
[ZURÜCK ZUR STARTSEITE](#)

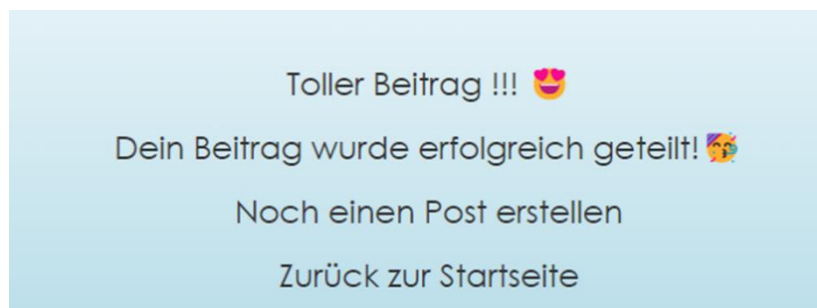
Flussdiagramm für die Suchfunktion:



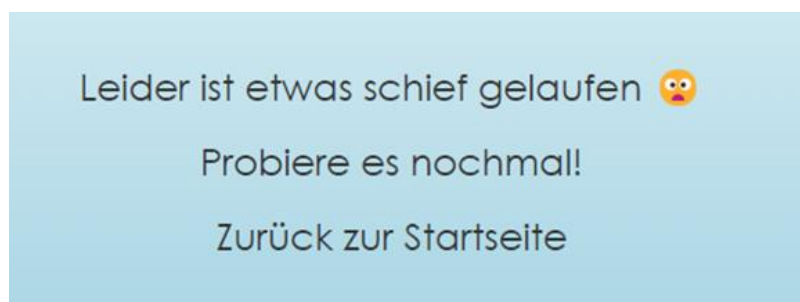
12. Create Post Backend

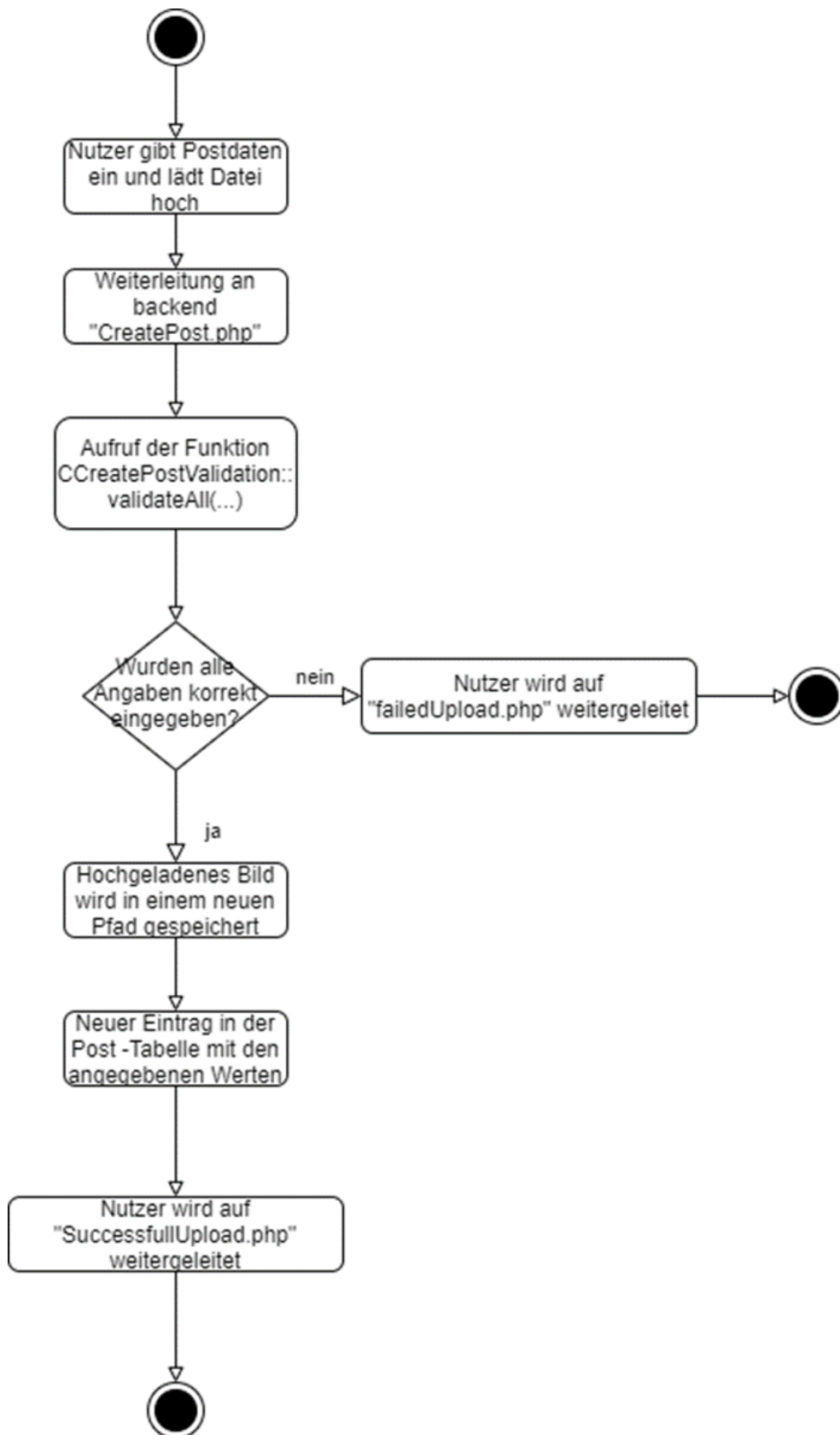
Das Backend bekommt mittels `$_POST` und `$_FILES` Methode die Daten für das Erstellen des Beitrags. Diese werden in Variablen gespeichert. Im darauffolgenden Schritt wird die Richtigkeit der Eingaben durch die Funktion `validateAll(...)` der Klasse „CCreatePostValidation“ geprüft. Bei der Funktion wird beispielsweise geprüft, dass Hashtags mit „#“ starten und es sich bei der Bilddatei um eine „jpg“, „png“ oder „jpeg“ Dateien handelt. Des weiteren werden alle Eingaben auf ihre Länge und Werte geprüft. Falls alle eingegebenen Werte auf ihre Korrektheit geprüft wurden wird das hochgeladene Bild in einem neuen Pfad gespeichert. Dieser besteht aus dem Benutzernamen und dem aktuellen Zeitstempel. Dieser Pfad wird mit den anderen eingegebenen Werten und dem aktuellen Datum in der Datenbank gespeichert.

Der User wird im Anschluss auf die Seite „SuccessfulUpload.php“ weitergeleitet. Dort wird der Nutzer informiert, dass der Beitrag erstellt wurde und hat die Möglichkeit mittels eines Links entweder noch einen Beitrag zu erstellen oder zur Startseite zurückzukehren.



Falls die Angaben nicht richtig sind, wird der Nutzer nur an die „failedUpload.php“ Seite weitergeleitet und informiert, dass etwas schiefgelaufen ist und erhält die Möglichkeit zurückzukehren.





13. Profil ansehen

Zu Beginn startet eine Funktion die Mithilfe der Session, die Accountkey den Vornamen gegebenenfalls den zweiten Namen und Nachnamen ermittelt. Außerdem wird die Rolle des Users ermittelt, sowie der letzte Login. Im Hauptteil der Seite werden über weitere Datenbankabfragen wieder Vorname, gegebenenfalls der zweite Name und Nachname sowie Geburtsdatum, Telefonnummer, Adresse, Postleitzahl, Land, Benutzername und die Email des angemeldeten User importiert. Dazu gibt es die Klasse CUserDataLoading.php. Zu Beginn wird erstmal die ServerConnention hergestellt. Mittels get-Funktionen und dem AccountKey wird durch query eine Datenbankabfrage gestartet, die die angeforderten Datensätze aus der Datenbank lesen soll. Execute führt dann im nächsten Schritt die Anforderung aus und das Abfrageergebnis wird zurückgegeben und gefeicht.

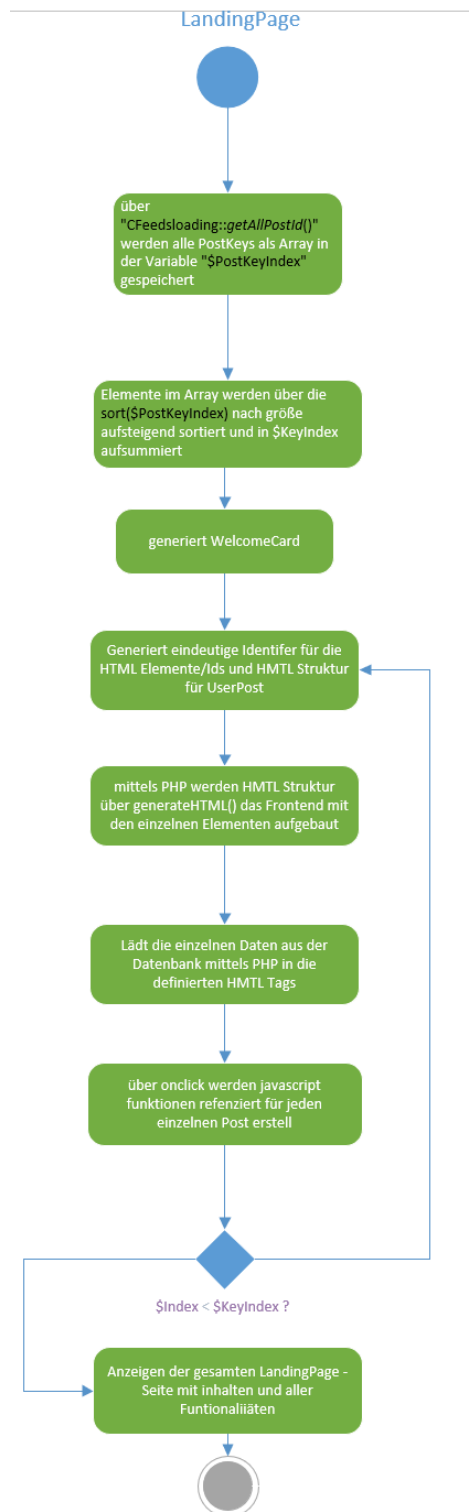
The screenshot shows the 'Linkvel' application interface. The top navigation bar includes the logo 'Linkvel', a search icon, and links for 'STARTSEITE', 'MEIN KONTO', and 'LOGOUT'. The main content area is titled 'Profil-Daten bearbeiten'. On the left, there is a user profile card for 'Duc Duong Nguyen' with a placeholder image and the text 'User-Rolle: Standarduser' and 'zuletzt online: 2022-01-22 12:16:34'. To the right is a form with the following fields:

Vorname	Postleitzahl
Duc Duong	12345
Mittelnname	Land
-	Germany
Nachname	Email
Nguyen	duc123@gmail.com
Adresse	Passwort
TestStraße 123	

Below the form are two buttons: 'Speicher' (Save) and 'Zurück' (Back). A green arrow points to the 'Speicher' button.

14. Landing Page – UserPost

Wie im Flussdiagramm beschrieben, lässt sich die LandingPage mittels PHP im Backend als HTML Generator erzeugen. Wichtig ist hierbei zu beachten, dass jeder der einzelnen Identifier einzigartig sein muss. Diese sind notwendig um die Erstellung der Post JavaScript, AJAX anwenden zu können um die User Interaktionen in Kraftsetzen zu können.



Aktiv anzeigen

Ansicht der Datensätze: 25

Zurück

Neu

Neu

Neu

Neu

Neu

Neu

Neu

PostKey

SuperheadlineKey

PostText

Headline

PictureFile

Post_Date_Time

Hashtags

MapLatitude

MapLongitude

Bearbeiten

Kopieren

Löschen

1

Im letzten Sommer besuchte ich die Stadt Lissabon.

Wunderschönes Lissabon

./UserMedia/PhotoMedia/2021/20210711_lissabon.jpg

2021-12-28 22:22:30

#lissabon

40.0750

-8.1721

Bearbeiten

Kopieren

Löschen

2

Lissabon - eine Mischung aus Superlativen mit über

Monteagle - Lissabon

./UserMedia/PhotoMedia/2021/20210711_lissabon.jpg

2021-12-28 22:30:14

#lissabon

41.5033

-8.1197

Bearbeiten

Kopieren

Löschen

3

Barcelona ist eine Weltstadt in vielerlei Hinsicht.

Barcelona - Mehr als ein Club

./UserMedia/PhotoMedia/2021/20210711_barcelona.jpg

2021-12-28 23:29:17

#barcelona

41.3825

2.1788

Bearbeiten

Kopieren

Löschen

4

New York ein Vorzeigebauwerk

New York - mehr Stadt als der

./UserMedia/PhotoMedia/2021/20210711_newyork.jpg

2022-01-13 00:36:03

#newyork

40.7306

-73.8352

Bearbeiten

Kopieren

Löschen

5

Test Text

Test

./UserMedia/PhotoMedia/2022/20220205210.jpg

2022-01-23 02:52:10

#test

40.0750

-8.1721

Speichert als Array in KeyIndex

\$KeyIndex

Willkommen auf Linkvel! Wir hoffen,
du hast einen wunderschönen Tag!

Welcome Card

WUNDERSCHÖNES
LISSABON

HeadlineID

User: PHeer
Im letzten Sommer besuchte ich die Stadt
Lissabon. Eines der schönsten Städte in Europa!



2



3



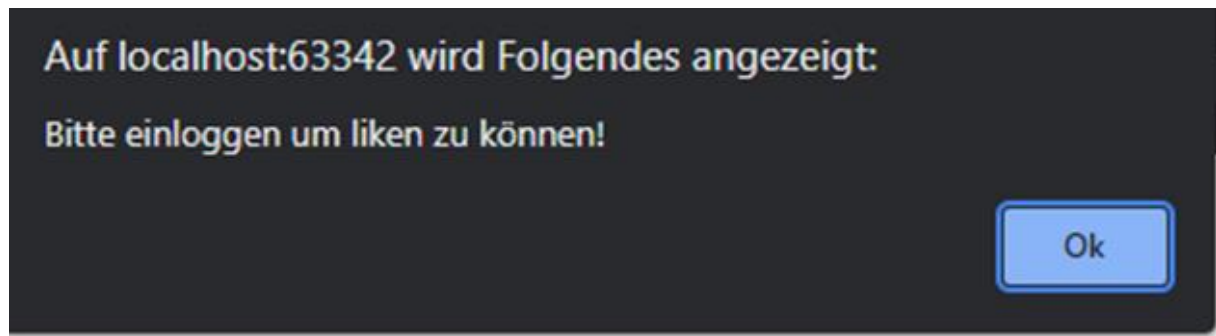
LikeButtonID
& LikeLabelID

CommentButtonID
& CommentLabelID

MapLocationButton

15. User-Like

Der Like - Icons und die Funktionalität zu einem Post wird mittels PHP im backend erstellt. Dabei werden die speziellen Identifier dazu genutzt, um die Interaktionen mit dem Post umzusetzen. Drückt der User auf den Like-Button so wird JavaScript an getriggert. Dabei wird zunächst geprüft, ob der User sich eingeloggt hat, wenn dies nicht der Fall ist, so wird mittels der alert-Funktion den User dazu angefordert sich einzuloggen, um die Interaktion realisieren zu können.



Im technischen wird folgendes realisiert:

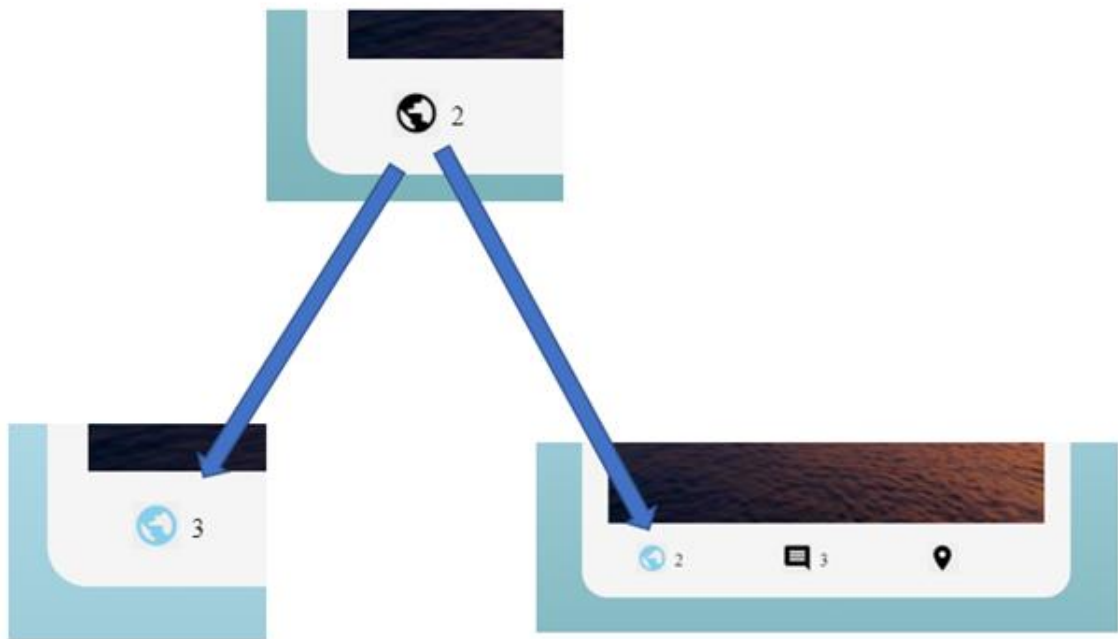
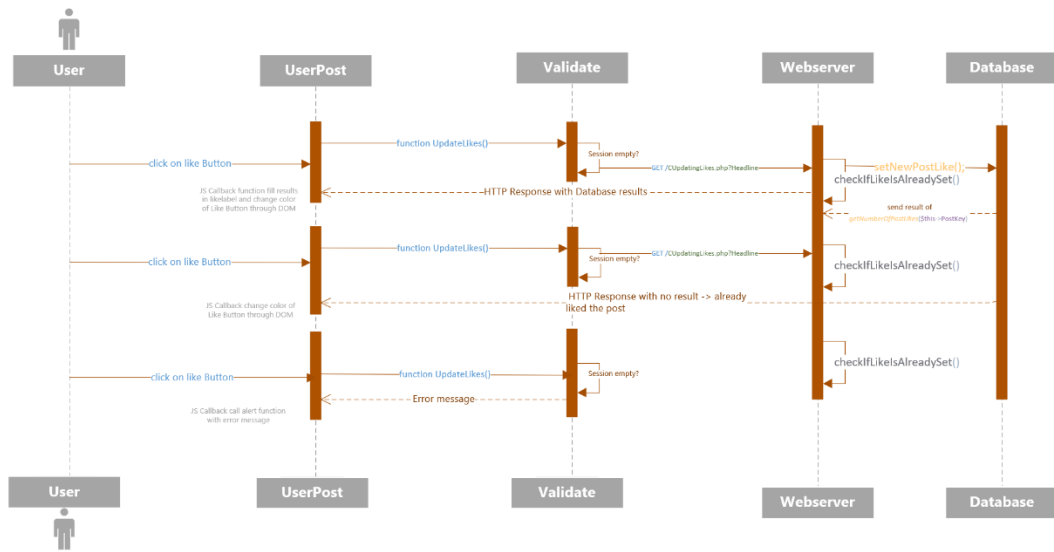
- Post wird mittels PHP im Backend realisiert
- Like Button wird beim Post Erstellung mit generiert und erhält einzigartige ID-Identifizier und einen JavaScript- Funktion UpdateLikes()
- dabei werden folgende Parameter übergeben, die sowohl die Überprüfung des eingeloggtten Users mittels der Session als auch die Überschrift und der zu verändernden Bereich durch DOM und JavaScript (in den Fall den Like Button) übergeben

```
<-icons' id='${LikeButtonID}' onClick='UpdateLikes("${LikeButtonID}", "${LikeLabelID}", "${HeadlineID}", "${Session}");' > &#xe80b; </button>
```

- mittels des Session Cookie wird überprüft, ob der User eingeloggt ist

Im Sequenzdiagramm wird detailliert beschrieben, wie die Daten an das Backend mittels AJAX umgesetzt wird. Um liken zu können ist es notwendig sich einzuloggen. Nachdem man eingeloggt ist, wird außerdem im Backend überprüft, ob ein User bereits diesen Post geliked hat. Dabei wird mittels des AccountKey eine SQL-Abfrage im Backend aufgerufen, welches überprüft, ob der User bereits in den Tabellen der Likes zu dem gewählten Post bereits hinterlegt wurde. Ist das der Fall, so wird zwar die Farbe des Like-Button geändert, aber die Gesamtanzahl der Likes wird nicht inkrementiert, da in der Datenbank keine neuen Daten in der Like-Tabellen geschrieben wurde.

Hat der User noch kein Like zum Post hinterlegt, so wird in der Tabelle like, die Daten geschrieben und die Farbe des Like-Button geändert.

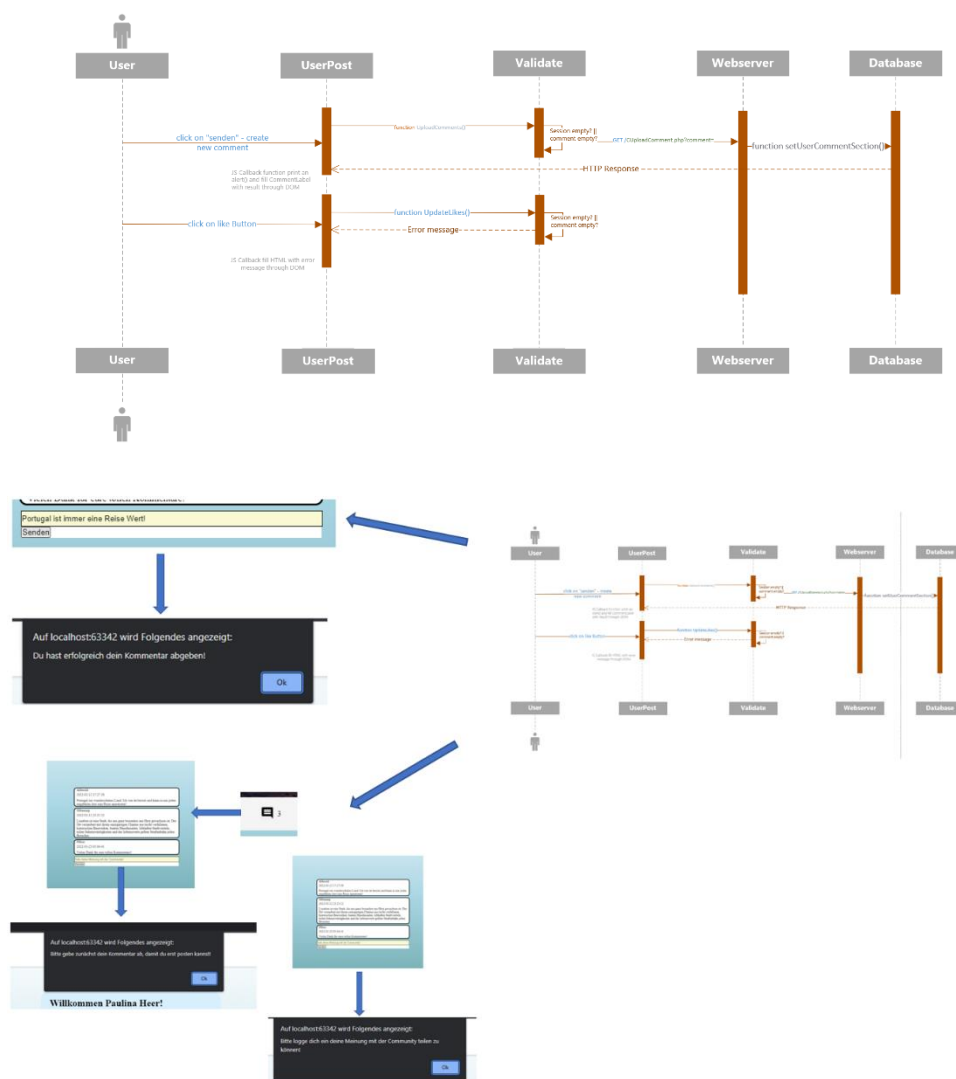


16. User Comment Post

Die Generierung des Icons, sowie Funktionen des User comment post werden sehr ähnlich zu Userlike erstellt. Auch hier wird mittels PHP im backend alles erstellt.

Um kommentieren zu können muss der User angemeldet sein und einen Text eingeben. Ist das Textfeld leer oder der User nicht angemeldet, so wird ein Fehler Meldung ausgegeben. Diese Überprüfungen erfolgen im JavaScript vor der Versendung an das Backend. Die verschiedenen Fehler werden mittels Alert-Boxen den User angezeigt.

Hat der User sich eingeloggt und auch einen Text verfasst, so kann er diese mittels des Button „Senden“ an das Backend mittels Ajax senden. Wird dieser in der Datenbank geschrieben, so wird die Gesamtanzahl der Kommentare zu einen Post inkrementiert und im „Comment-Label“ angezeigt.



(Sequenzdiagramm: UserCommentPost + Fehlermeldungen)

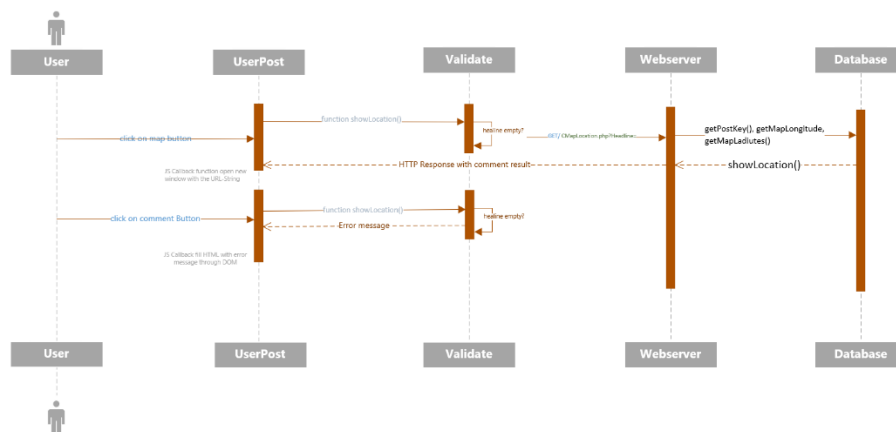
17. Map-Location

Das beste Feature in unserem Projekt ist sicherlich die „Map-Location Funktion“. Dieses war eines der ersten Ideen, die unser Projekt maßgeblich beeinflusst haben und dazu verleitet hat das Projekt durchzuführen.

Jeder Superuser ist verpflichtet Koordinaten zu seinem Post anzugeben. Dadurch ist es möglich mittels eines API, in den Fall Google Map Api und den Koordinaten aus der Datenbank herauszulesen und diesen der API als Parameter mitzugeben. Dadurch ist es möglich den Standort des Posts über Google Map zu referenzieren und anzuzeigen. Dies steigert den User Experience und gibt den Social Media Eigenschaft eine weiter besondere Duftnote.

Im technischen Detail wird mittels JavaScript die Funktion `showLocation()` aufgerufen. Dieses erhält als Parameter die Überschrift des Post. Damit kann man die ID des Post ausfindig machen und die dazugehörigen Koordinaten aus der Datenbank rausholen. Mittels der PHP-Methode „`showLocation()`“ werden die Koordinaten aus der Datenbank in eine String der API als Parameter mit übergeben und zurückgegeben. Dieser String wird mittels der Methode „`open()`“ vom Objekt „`window`“ in einen neuen Fenster geöffnet.

Hier im Bild: Standort des ersten Posts auf der Brücke „Pont 25 de Abril“ in Lissabon, Portugal.

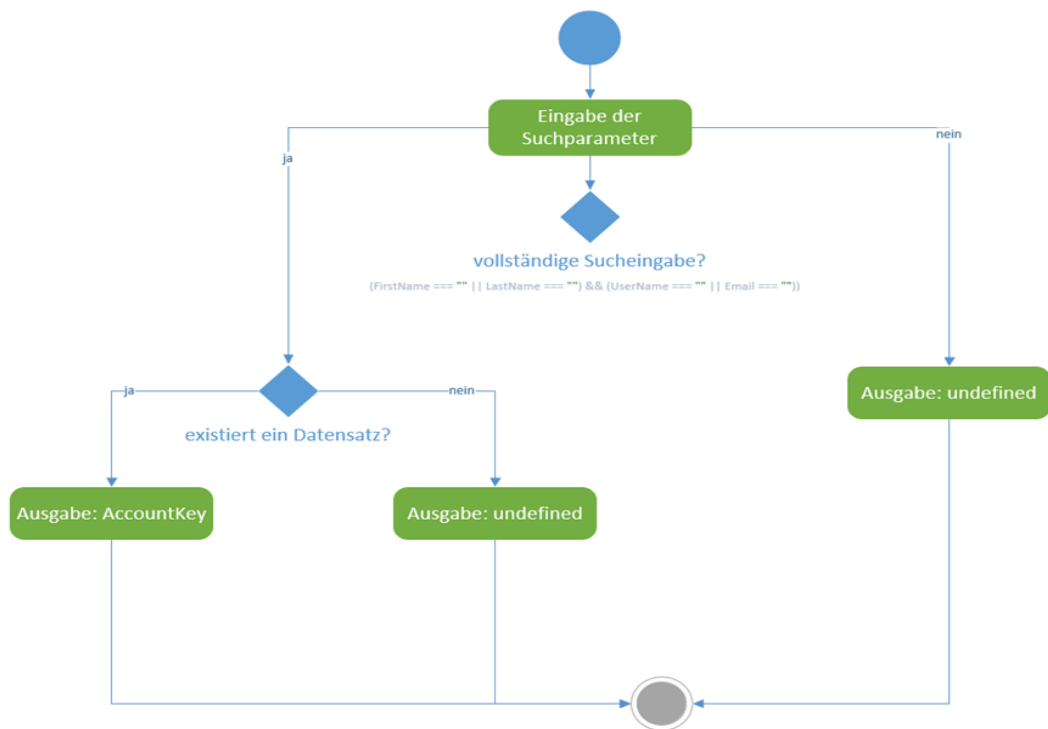


18. Search AccountKey – Ajax

Aus dem Diagramm anschaulich dargestellt.

Mittels diesen ID kann der Employee/Mitarbeiter weitere Aktionen ausführen.

Im technischen Detail wird mittels JavaScript und AJAX eine Datenbankabfrage im Backend durchgeführt. Gibt es ein Ergebnis so wird dieser im DOM in der Box angezeigt.



Datensatz wurde gefunden:

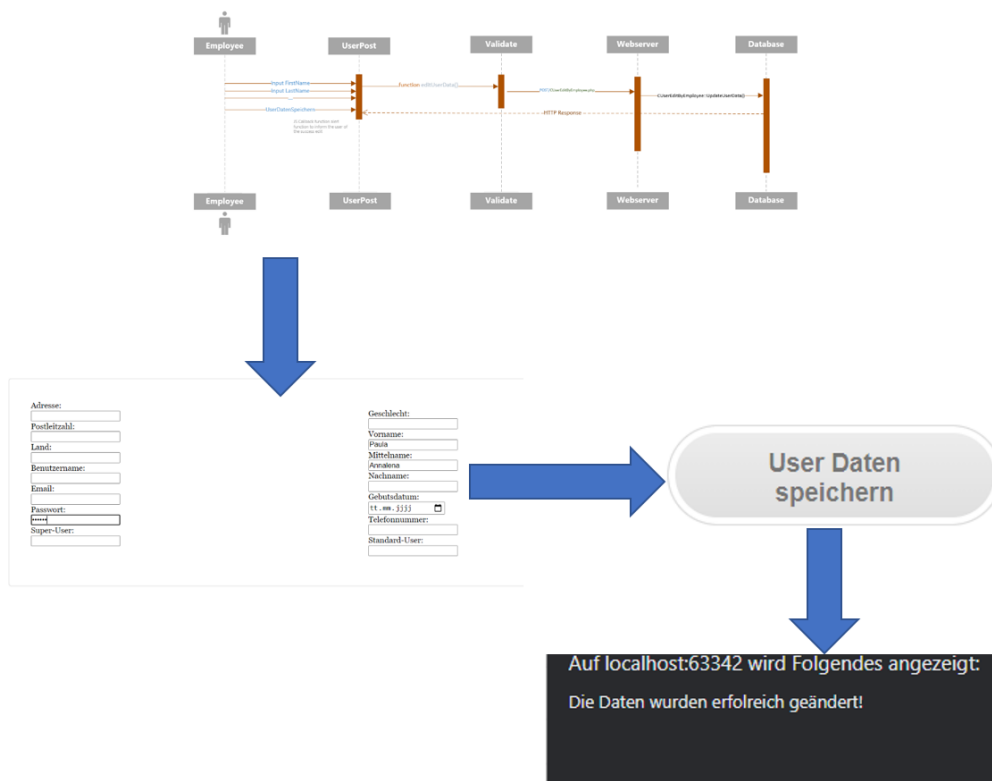
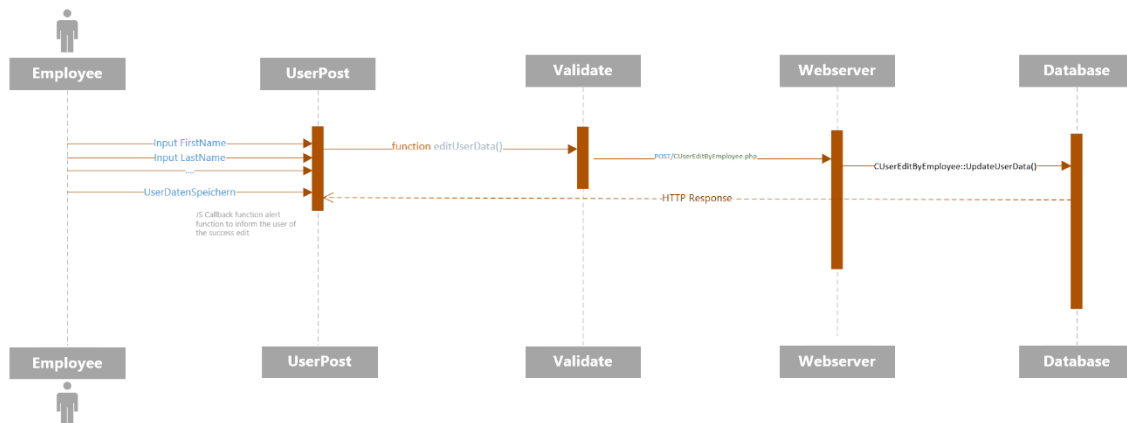
Vorname	<input type="text" value="Paulina"/>	<input type="text" value="1"/>
Nachname	<input type="text" value="Heer"/>	
Benutzername	<input type="text"/>	
Email	<input type="text"/>	
		<input type="button" value="AccountID suchen"/>

Kein Daten wurde gefunden:

Vorname	<input type="text"/>	<input type="text" value="undefined"/>
Nachname	<input type="text"/>	
Benutzername	<input type="text"/>	
Email	<input type="text"/>	
		<input type="button" value="AccountID suchen"/>

19. Update User Data By Employee

Über den AccountID erhält der Employee alle Daten, die er benötigt, um Account Daten eines Users anzuzeigen. (Ist auch im Optionfeld der Mitarbeiter → gleichen Prinzips). Nun könne die Mitarbeiter, nachdem er die Daten des Users gelesen hat auch verändern, dazu muss man ein AccountID im Formular eingeben und dazu auch noch die zu ändernde Daten in dem Formular der User Daten eintragen. Über AJAX wird es an das Backend gesendet und geschrieben. Beim erfolgreichen Überschreiben wird eine Meldung ausgegeben.



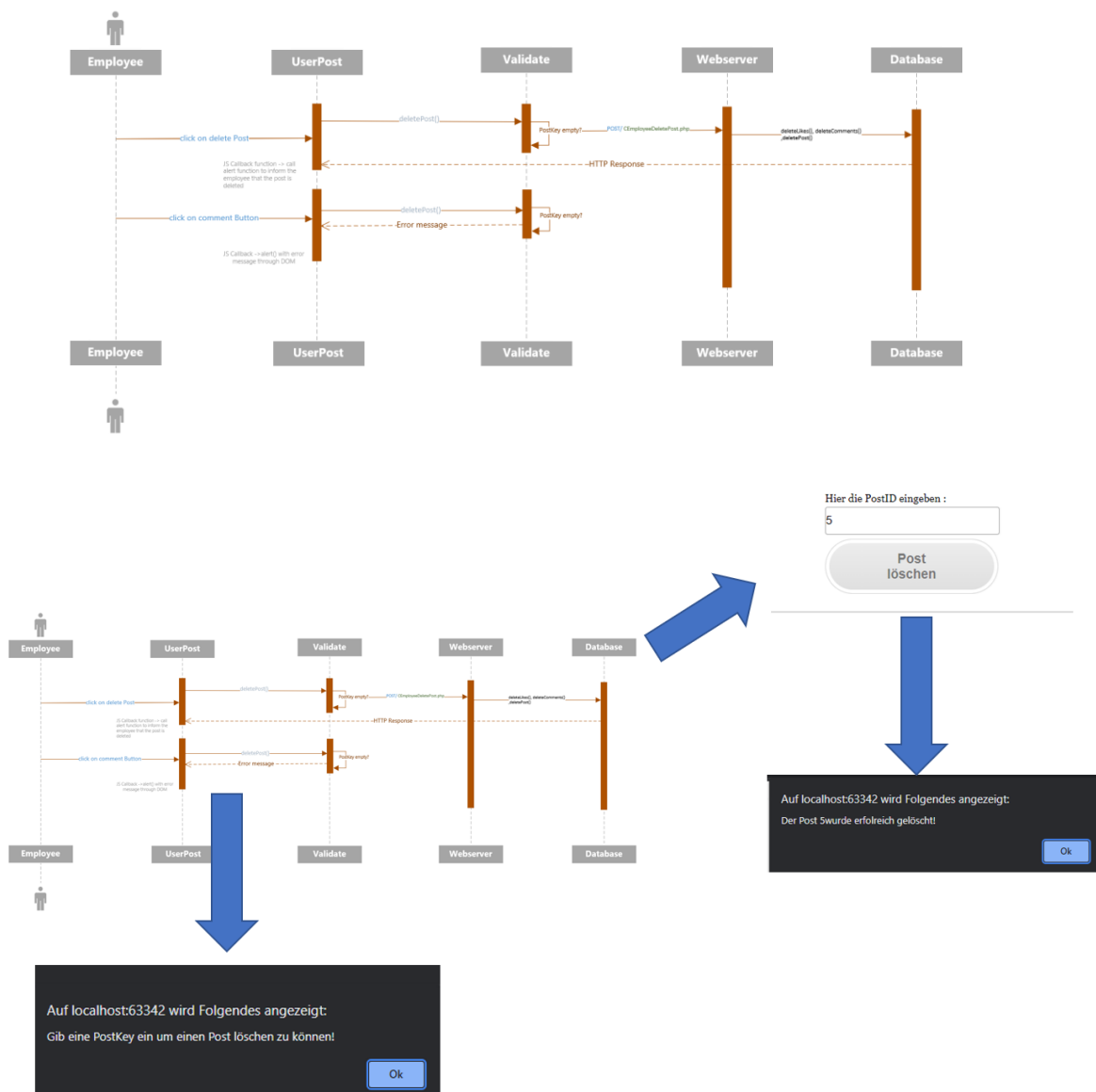
19. DeletePostByEmployee

Mittels des PostID/PostKey, welches beim Suchen des Users angezeigt wird, kann der Employee auch Post und die dazugehörigen Likes und Comment löschen. Dabei wird auch wie bei den anderen Funktionalitäten des Employee Page Ajax verwendet, um die Daten zu manipulieren bzw. zu löschen.

Dabei wird im Frontend mittels JavaScript überprüft, ob das PostID-Feld leer ist.

Tritt der Fall ein, dass dieser leer ist, so wird eine Fehlermeldung angezeigt.

Kommt es zum Fall, dass eine gültige PostID vorhanden ist, so wird beim erfolgreichen löschen der Daten mittels Alert-box angezeigt.



20. Optionfeld Mitarbeiterdaten

Beim Option Feld der Mitarbeiterdaten, wird im generellen, sehr ähnlich zum Search AccountKey, das gleiche Verfahren verwendet. Hierbei ist nur den Unterschied, dass man nicht mittels des Namens einen User sucht, sondern mittels des angegebenen Wertes im Option Feld direkt in der Mitarbeiter Tabelle in der Datenbank über diesen Wert gesucht. Auch hier wird mittels JavaScript und AJAX die Daten an das Backend geschickt und mittels SQL-Datenbank abfrage die Daten über PHP als Tabellenform angezeigt.

Select Employee: ▼

Suchen!

Account ID:	7
Firstname:	Antonia
MidName:	
LastName:	Geschke
Birthday:	2003-02-12
StreetAddress:	Am See 2
ZipCode:	99821
Country:	Germany
Email:	AGeschke@linkvel.de
Mitarbeiterrolle:	Employee
Abteilung:	Development

ALLE USERINTERAKTION IST ALS EIN FLUSSDIAGRAMM im DokemenationsFolder NACH ZU LESEN DA DIESES ZU GROß IST.

21. Known Bugs

- Footer passt sich nicht jede Seite an
- aus zeitlichen Gründen nicht alle Formulare mit Überprüfung, führt ggfs. zu schlechter Datenqualität
- Passwort Änderung mittels Employee Formular nimmt einen 0 Wert an, wenn es leer gelassen wird
- CSS des Headlines hat einen Bug im ersten Post im Landing Page → diese ist von manchen Winkeln verwackelt
- sicherheitstechnische Schwachstellen

22. Verbesserungen

- MCV -Pattern für besseren Projektstruktur

- besseren Code-Style
- mehr Userinteraktionen und Foto Gallery
- weiter Einbettung von Medien wie zum Beispiel Audio – Files
- bessere Überprüfung der Eingabe Daten in Backend und Frontend
- Unittest
- IT-Sicherheitslücken beheben
- Bessere Anpassung des CSS und bessere User Experience Design

23 Lesson Learned

- stetige Entwicklung und Verbesserung des Codes
- Erlernung der Programmiersprachen PHP, JavaScript
- Erlernung der Auszeichnungssprache HTML und CSS
- gute Zusammenarbeit der einzelnen Mitglieder
- Probleme die gelöst wurden:
 - Durchstich zwischen Backend und Frontend
 - Interaktive Webseite (Liken, Posten, Standort)
 - automatisiert Post aus erstellen und dazugehörigen Daten aus der Datenbank laden
 - Login und Session
 - AJAX für Formulare
 - Responsive Seiten mittels CSS
 - Schnittstellenerzeugung für Wiederverwendung von Codes