Tatyana Merlo – Natalie Stalder – Nadja Stadelmann

Travelling I

M133 – Webapplikation realisieren

# Einleitung

## Systemanforderungen

## Vorwissen

## Dokument

### Grund

### Aufbau

* Analyse
  + Beschrieb und erste Aufsplittung in einzelne Teilbereiche
* Vorgehen
  + Abhängigkeiten, Aufgabenübersicht
* Design
  + Detaillierte Analyse DB-Design
  + Detaillierte Analyse Programmstruktur
* Implementation
* Testing
  + Testcases basieren auf UseCases

Inhalt

[Einleitung 1](#_Toc494377893)

[Systemanforderungen 1](#_Toc494377894)

[Vorwissen 1](#_Toc494377895)

[Dokument 1](#_Toc494377896)

[Grund 1](#_Toc494377897)

[Aufbau 1](#_Toc494377898)

[Vision 3](#_Toc494377899)

[Analyse 4](#_Toc494377900)

[Anforderungen 4](#_Toc494377901)

[Must have 4](#_Toc494377902)

[Nice to have 4](#_Toc494377903)

[Use Cases 4](#_Toc494377904)

[Design 5](#_Toc494377905)

[DB 5](#_Toc494377906)

[Tabellenbeschrieb 5](#_Toc494377907)

[User 5](#_Toc494377908)

[Location 5](#_Toc494377909)

[Post 5](#_Toc494377910)

[Picture 5](#_Toc494377911)

[Visited 5](#_Toc494377912)

[Designanpassungen 5](#_Toc494377913)

[Programm 5](#_Toc494377914)

[Model 5](#_Toc494377915)

[View 5](#_Toc494377916)

[Controller 5](#_Toc494377917)

[Designanpassungen 5](#_Toc494377918)

[Implementation 7](#_Toc494377919)

[Implementierung 7](#_Toc494377920)

[Testing 8](#_Toc494377921)

# Vision

Man bereist die Welt und möchte möglichst viele Orte sehen. Doch sich dabei auf einen Schlag an alle Orte zu erinnern, die besten Fotos an einem zentralen Ort lagern und allenfalls Freunden eine Hotel- oder Restaurantempfehlung liefern – das fällt oftmals schwer.

Für all diese Probleme bringen wir mit dem Travelblog ‘Travelling I’ die Lösung: Auf einer Karte kann eingetragen werden, an welchem Ort man bereits gewesen ist. Das entsprechende Land wird dabei farbig eingefärbt und die Stadt mittels eines Punktes markiert. So erhält man eine Übersicht über seine vergangenen Reisen, die man überall auf der Welt über das Internet abrufen kann!

Zu jedem Ort kann auch ein Blogeintrag geschrieben werden, um tolle Hotels, Restaurants oder Ausflugsziele festzuhalten. Dazu kann man seine schönsten, lustigsten oder emotionalsten Fotos hochladen, um immer wieder in diesen Erinnerungen zu schwelgen.

Damit diese Daten nicht einfach öffentlich preisgegeben werden, muss man sich dazu zuerst registrieren und jeweils einloggen. So werden die Ferienerlebnisse vor unerwünschten Blicken geschützt.

## Erweiterungsmöglichkeiten

In weiteren Schritten stellen wir uns vor, dass sich die User Restaurants oder vorhandene Aktivitäten an den spezifischen Orten anzeigen lassen können. Da das Wetter für Ausflüge und das Gepäck ebenfalls wichtig ist, wird auch dieses eingebunden.

Um die Suche durch seine Einträge zu erleichtern, werden später Kategorien eingebunden. So kann der User zum Beispiel die Kategorie ‘Hotel’ angeben und später ganz einfach die entsprechend markierten Einträge filtern.

Doch bei den vergangenen Reisen soll es nicht bleiben! Pläne zu schmieden für neue Ferien ist ebenfalls spannend und erfreulich. Damit dies auch locker vom Hocker geht, werden den Usern entsprechende Flüge und Hotels angezeigt, die für sie geeignet sein können.

# Analyse

## Anforderungen

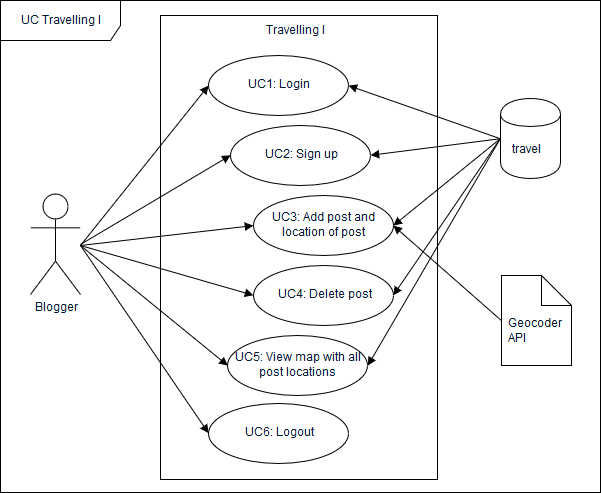
### Must have

1. Login / Registrierung
2. Karte – wo warst du?
   1. Nadel auf Stadt
   2. Blogeintrag mit Fotos

## Nice to have

1. Google Maps einbindung
   1. Restaurants in der jeweiligen Stadt suchen
2. Wetter anzeige
   1. Name der Stadt eingeben 🡪 entsprechendes Wetter angeben
3. Vorhandene Aktivitäten anzeigen lassen
   1. Europaweit / Weltweit
   2. Tripadvisor API?
4. Blogmässige Notizen erweitern z.B. Kategorien
   1. Restaurant
   2. Hotels
   3. Festivals
   4. Einkaufscentern
   5. ….
5. Plan für nächste Reise(n)
   1. gute Flüge und Hotels anzeigen lassen

## Use Cases



|  |  |
| --- | --- |
| **Identifikation** | **UC1: Login** |
| **Beschreibung** | Ein Blogger loggt sich ein. |
| **Primärer Actor** | Blogger |
| **Vorbedingungen** | Der Blogger muss bereits einen Account haben (Sign up) |
| **Trigger** | Blogger wählt "Login" |
| **Standardablauf** | 1. Link von Travelling I eingeben  2. Benutzernamen und Passwort eingeben und "Login" klicken  3. Blogger wird eingeloggt |
| **Alternative Abläufe** | 2a. Falscher (nicht in Datenbank erfasster) Benutzernamen eingegeben 🡪 Blogger kann sich nicht einloggen, Login-Seite wird erneut – mit einer Fehlermeldung - angezeigt  2b. Blogger gibt falsches Passwort an 🡪 Es wird ebenfalls eine Fehlermeldung ausgegeben und die Login-Seite neu geladen, der Benutzername bleibt.  2c. Kein Input in den Feldern 🡪 Blogger kann sich nicht einloggen, Login-Seite wird erneut – mit einer Fehlermeldung - angezeigt |

|  |  |
| --- | --- |
| **Identifikation** | **UC2: Sign up** |
| **Beschreibung** | Ein Blogger erstellt einen Account. |
| **Primärer Actor** | Blogger |
| **Vorbedingungen** | - |
| **Trigger** | Blogger wählt "Sign up" |
| **Standardablauf** | 1. Link von Travelling I eingeben  2. Auf der Login-Seite auf die Sign-Up-Seite wechseln: "Sign up here" klicken  3. Heimatsstadt suchen, Submit Button wird freigegeben  4. Benutzernamen, Passwort sowie eine Passwortbestätigung eingeben und "Sign up" wählen  5. Bloggerkonto wird erstellt und der Blogger gelangt auf die Startseite im Logged-In-Bereich |
| **Alternative Abläufe** | 4a. Bereits in Datenbank erfasster Benutzernamen eingegeben 🡪 Bloggerkonto kann nicht erstellt werden 🡪 Sign-Up-Seite wird erneut geladen mit einer Fehlermeldung  4b. Blogger gibt nicht übereinstimmende Passwortbestätigung ein 🡪 Es wird eine Fehlermeldung ausgegeben ohne die Seite neu zu laden.  4c. Blogger sendet ein leeres Formular ab 🡪 Fehlermeldung und der Blogger bleibt auf der Sign-Up-Seite  3a. Stadt nicht gefunden 🡪 Submit Button wird nicht freigegeben |

|  |  |
| --- | --- |
| **Identifikation** | **UC3: Add post and location of post** |
| **Beschreibung** | Ein Blogger fügt einen neuen Ort und den dazugehörigen Post hinzu |
| **Primärer Actor** | Blogger |
| **Vorbedingungen** | Benutzer eingeloggt |
| **Trigger** | Blogger klickt auf "Add Location to Map " |
| **Standardablauf** | 1. Auf Home Seite Ort hinzufügen, auf „Add Location to Map“ klicken  2. Posttitel, Posttext, Bilder für den Post hinzufügen  3. "Submit" wählen  4. Post wird erstellt und der Standort auf der Blogger spezifischen Weltkarte markiert |
| **Alternative Abläufe** | 1a. Location wird nicht gefunden, keine Weiterleitung auf „Edit Post“ wird vorgenommen  1b. Keine Internetverbindung vorhanden 🡪Location kann nicht gefunden werden  3a. Die ausgewählte Location hat bereits einen Posteintrag 🡪 eine Fehlermeldung wird angezeigt  3b. Die Bilder, die hochzuladen sind, sind nicht im richtigen Format/ in der richtigen Grösse etc. 🡪 Es wird ebenfalls eine Fehlermeldung ausgegeben, der Post wird ohne Bilder resp. mit den Bildern abgespeichert, welche hochgeladen werden konnten |

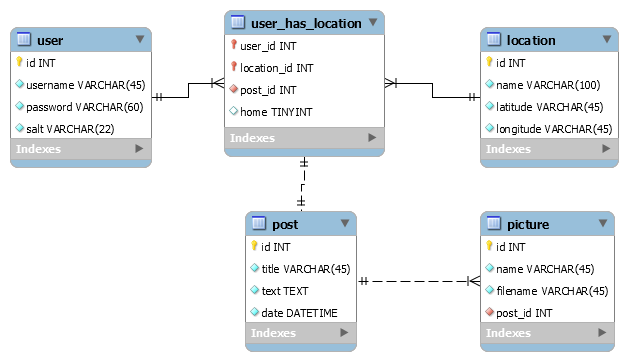
|  |  |
| --- | --- |
| **Identifikation** | **UC4: Delete Post** |
| **Beschreibung** | Ein Blogger löscht einer seiner Posts. |
| **Primärer Actor** | Blogger |
| **Vorbedingungen** | Benutzer eingeloggt |
| **Trigger** | Blogger wählt für bestimmten Post "Delete". |
| **Standardablauf** | 1. Blogger geht auf seinen Blog und wählt die "Delete"-Option, welche unter jedem Post angezeigt wird.  2. Post wurde gelöscht |
| **Alternative Abläufe** | - |

|  |  |
| --- | --- |
| **Identifikation** | **UC5: View map with all post locations** |
| **Beschreibung** | Ein Blogger sieht sich die Home Seite an |
| **Primärer Actor** | Blogger |
| **Vorbedingungen** | Benutzer eingeloggt |
| **Trigger** | Klick auf Home Menü |
| **Standardablauf** | 1. Klick auf Home Menü  2. Karte mit markierten Locations wird angezeigt |
| **Alternative Abläufe** | 2a. Internetverbindung nicht vorhanden 🡪 Karte wird nicht angezeigt |

|  |  |
| --- | --- |
| **Identifikation** | **UC6: Logout** |
| **Beschreibung** | Ein Blogger loggt sich aus. |
| **Primärer Actor** | Blogger |
| **Vorbedingungen** | Benutzer eingeloggt |
| **Trigger** | Blogger wählt die Aktion "Logout" |
| **Standardablauf** | 1. "Logout" wählen  2. Benutzer ausgeloggt |
| **Alternative Abläufe** | - |

# Design

## DB



### Tabellenbeschrieb

#### User

Der User braucht einen Usernamen sowie das Passwort. Der Username ist unique definiert. Das Passwort ist mittels BCrypt verschlüsselt – das heisst, es benötigt eine Länge von VARCHAR(60). Ausserdem wird der Salt VARCHAR(22) abgespeichert, er besteht aus einer generierten Zeichenkette.

#### Location

Hier werden die besuchten Orte abgespeichert. Wir speichern den Namen der Location, sowie die Längen- und Breitengrade VARCHAR(45) in dieser Tabelle ab, da Google GeoChart die Nadeln mittels der Koordinaten setzt.

#### Post

Im Post wird das Datum, der Titel und ein Text eingespeichert.

#### Picture

Der Bildernam, wird gespeichert. Weiter wird der Filename inklusive relativem Pfad gespeichert, damit das passende Bild geladen werden kann.

#### Visited

Zwischentabelle zwischen dem User und der Location. Mehrere User können dieselbe Stadt besuchen und ein User kann mehrere Städte besuchen. Weiter ist hier der Fremdschlüssel zum zugehörigen Post eingespeichert. Auch wird hier gespeichert, ob eine Location für den Benutzer die «Home City» ist, da diese speziell gehandhabt hat(kann nicht gelöscht werden, hat keinen dazugehörigen Post welcher angezeigt wird).

### Designanpassungen

## Programm

### Model

### View

### Controller

### Designanpassungen

# Implementation

## Implementierung

# Testing