

# WanderWise

Chau Anh Do

Yasin Cherif

# Gliederung

1. NUTZERVERSPRECHEN
2. WEB-VERHALTEN UND STRUKTUR
3. DESIGN DECISIONS
4. UMFANG UND PRIORITÄTEN



**Lara Schubert**  
Grafikdesignerin



"Ich liebe es, Berlin immer wieder neu zu entdecken – sei es ein perfekter Sonnenuntergang über der Spree oder ein versteckter Ort, den kaum jemand kennt."

### Lieblingsorte

Tempelhofer Feld,  
Oberbaumbrücke bei  
Sonnenuntergang und  
versteckte Cafés mit Blick  
auf den Landwehrkanal.

### Herausforderungen:

Viele Empfehlungen auf Social Media  
sind ungenau oder touristisch  
überlaufen.

Lara hat keine Lust auf übermäßig  
bearbeitete Bilder oder gesponserte  
Inhalte – sie sucht nach echten Tipps.

Auf Plattformen wie Instagram geht  
es oft nur um Selbstdarstellung, nicht  
um echten Austausch.

### Hobby

Lara ist eine kreative und naturverbundene Person, die es liebt, ihre Freizeit draußen zu verbringen. Sie ist immer auf der Suche nach neuen Orten, um schöne Fotos zu machen – sei es für ihre Arbeit, ihre Social-Media-Kanäle oder einfach, um ihre Freunde zu inspirieren. Lara ist umweltbewusst, <sup>93%</sup> liebt Spaziergänge in der <sup>95%</sup> Natur und entdeckt gern urbane Highlights wie Dachterrassen, versteckte Parks oder Flusssufer.

### Ziele & Motivationen

- Neue Orte entdecken
- Inspiration teilen
- Ästhetische Fotos schießen
- Sich mit Gleichgesinnten vernetzen

# 1. VALUE PROPOSITION

Probleme, die  
unsere  
Website löst

Fehlende Übersicht über schöne Orte in Berlin

Schwierigkeit, authentische Tipps zu finden

Soziale Interaktion rund um visuelle Erlebnisse

Uninspiriertheit bei Freizeitgestaltung

Mangel an Plattformen, die auf lokale Schönheit fokussiert sind

# Nutzerversprechen

**Inspirationsquelle  
für neue Orte**

**Authentizität und  
Community-  
Ansatz**

**Personalisierte  
Erfahrungen**

**Interaktive  
Features**

**Fokus auf Ästhetik  
und Natur in der  
Großstadt**

**Visuelle und  
geographische  
Orientierung**

## 2. App-Struktur

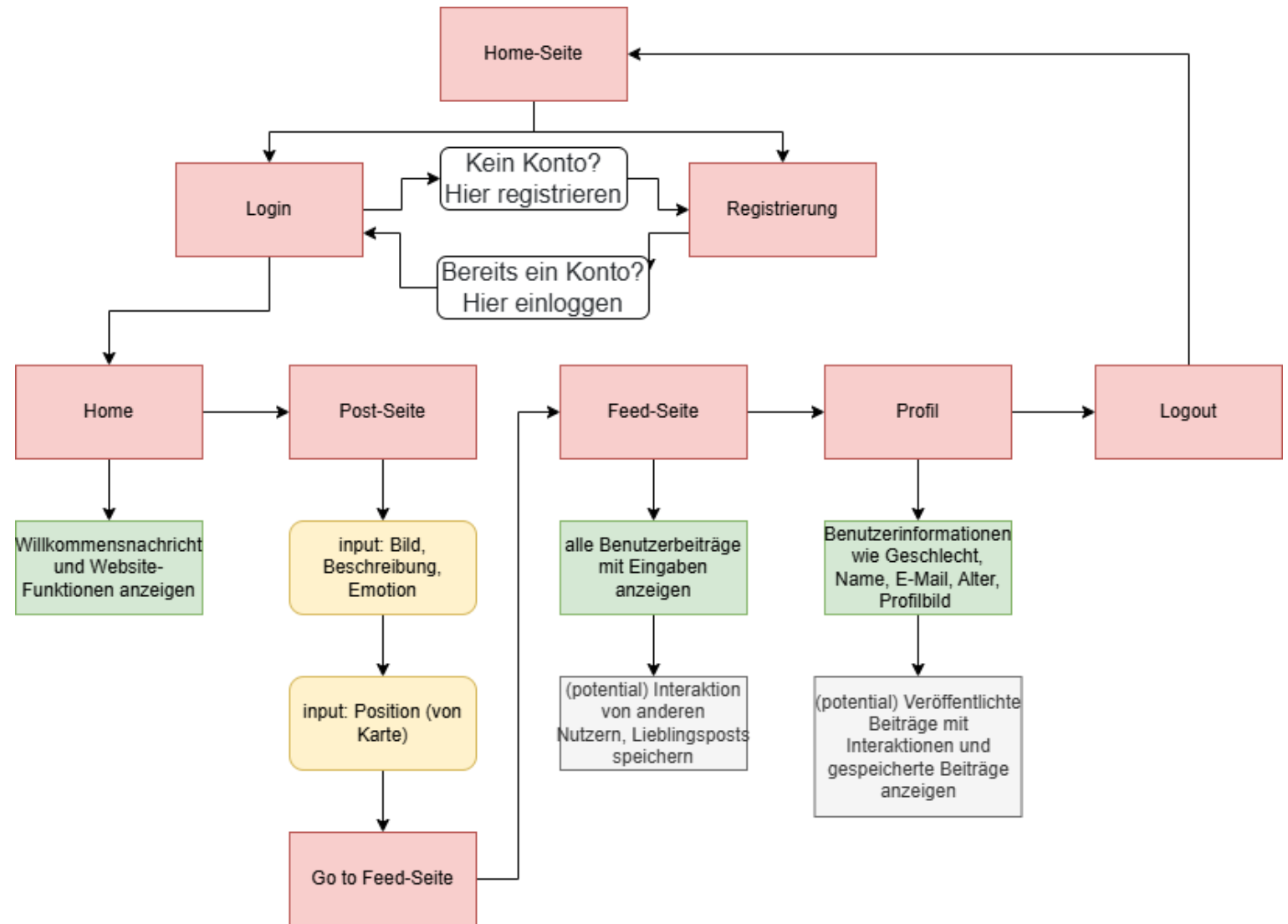


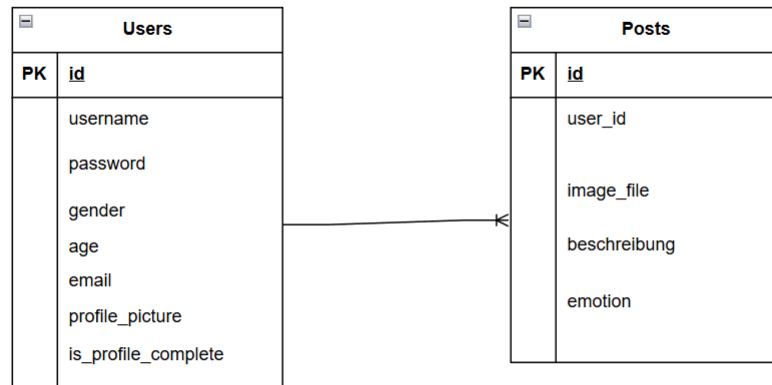
Abbildung 1: Ablaufsdiagramm von App.



# 3, Design Decisions: Datenbank SQLite

| Datenbank                          | Leicht<br>&<br>Einfach | Gut für<br>Komplexe<br>Abfragen | Gut<br>Unterstützt | Gut für Große<br>Datenmengen | Offline-<br>Unterstützung | Kein Setup<br>erforderlich | Plattformübergreifend | Schnelle<br>Performance | Gut für<br>Lokale<br>Nutzung |
|------------------------------------|------------------------|---------------------------------|--------------------|------------------------------|---------------------------|----------------------------|-----------------------|-------------------------|------------------------------|
| SQLite                             | ✓                      | ✓                               | ✓                  | ✓                            | ✓                         | ✓                          | ✓                     | ✓                       | ✓                            |
| MySQL /<br>MariaDB                 |                        | ✓                               | ✓                  | ✓                            |                           |                            | ✓                     | ✓                       |                              |
| PostgreSQL                         |                        | ✓                               | ✓                  | ✓                            |                           |                            | ✓                     | ✓                       |                              |
| LiteDB                             | ✓                      |                                 |                    |                              |                           | ✓                          |                       |                         | ✓                            |
| LevelDB                            | ✓                      |                                 |                    |                              |                           |                            |                       | ✓                       |                              |
| NeDB                               | ✓                      | ✓                               |                    |                              | ✓                         | ✓                          | ✓                     |                         | ✓                            |
| PouchDB                            | ✓                      | ✓                               |                    |                              | ✓                         |                            | ✓                     |                         | ✓                            |
| TinyDB                             | ✓                      |                                 |                    |                              |                           | ✓                          |                       |                         | ✓                            |
| Flat Files<br>(JSON,<br>CSV, YAML) | ✓                      |                                 |                    |                              | ✓                         | ✓                          |                       |                         | ✓                            |
| Firebird<br>SQL                    | ✓                      | ✓                               |                    | ✓                            |                           |                            | ✓                     | ✓                       |                              |
| CouchDB                            | ✓                      | ✓                               |                    |                              | ✓                         |                            | ✓                     |                         | ✓                            |

# Data Modell



```
id INTEGER NOT NULL,  
image_file VARCHAR(120) NOT NULL,  
description VARCHAR(500) NOT NULL,  
emotion VARCHAR(500) NOT NULL,  
PRIMARY KEY (id)  
--> /tableuseruser  
CREATE TABLE user (  
id INTEGER NOT NULL,  
username VARCHAR(150) NOT NULL,  
password VARCHAR(150) NOT NULL,  
gender VARCHAR(50),  
age INTEGER,  
email VARCHAR(150),  
profile_picture VARCHAR(150),  
is_profile_complete BOOLEAN,  
PRIMARY KEY (id),  
UNIQUE (username)
```

Abbildung 2: Datentyp und Datenbank im SQLite



# UI Design: CSS Grid + Bootstrap

## CSS Grid:

- Layout einer Seite mit großer Flexibilität und Präzision zu definieren.
- Header, Sidebar und Hauptinhalt angeordnet werden.

## Bootstrap:

- Hintergrundfarben und Transparenz (rgba) hinzugefügt.
- Buttons und Links reagieren auf Hover.
- Info-Card-Hintergrund leicht abgedunkelt.
- Unicode-Symbole (🌐, 📅) integriert.

```
/* Hauptcontainer */
.container {
  display: grid;
  grid-template-rows: 80px auto 1fr auto 50px; /* Definiert die Reihen */
  grid-template-columns: 1fr 4fr 1fr; /* Definiert die Spalten */
  grid-template-areas:
    "header header header"
    "left-aside main right-aside"
    "left-aside main right-aside"
    "left-aside main right-aside"
    "footer footer footer"; /* Definiert den Bereichsaufbau */
  height: 100vh; /* Die gesamte Höhe des Viewports */
  gap: 10px; /* Abstand zwischen den Bereichen */
  padding: 10px; /* Innenabstand für das Container-Element */
  color: #fff; /* Textfarbe */
}
```

Abbildung 3: Code mit CSS Grid

```
.nav-button:hover {
  background-color: #0056b3;
  color: #fff;
}

.nav-button.btn-danger {
  background-color: #dc3545; /* Rote Standardfarbe */
}

.nav-button.btn-danger:hover {
  background-color: #a71d2a; /* Abdunkeln beim Hover */
}
```

Abbildung 4: Hover-Interaktionen für Buttons

# Standort:



Choose your location:

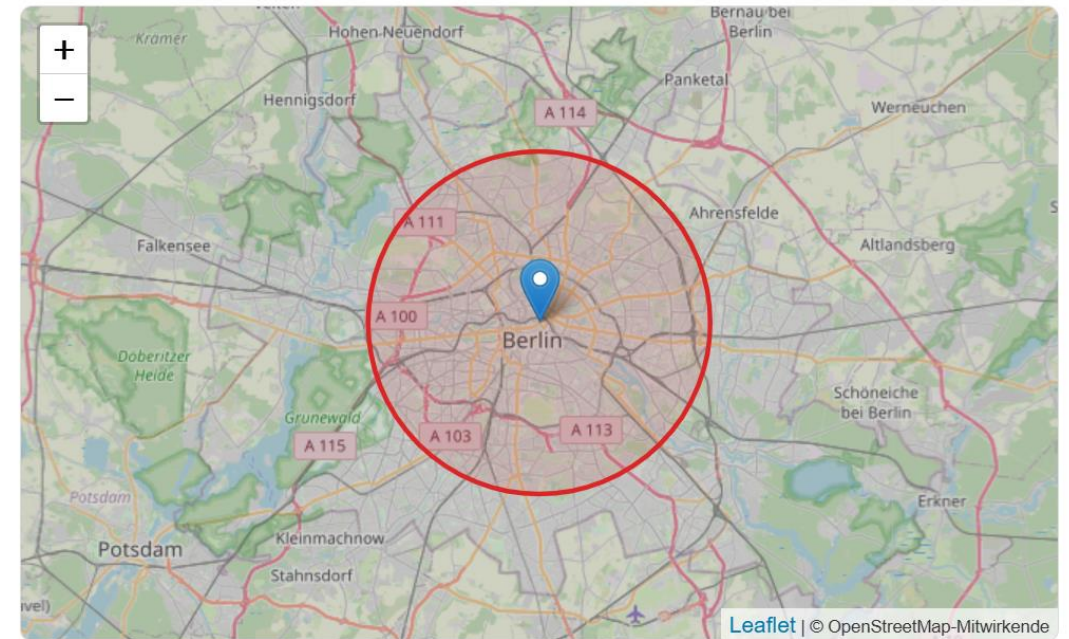


Abbildung 5: Interaktive Karten mit Leaflet erstellen.

## 4. Umfang und Prioritäten

- Website vollständig erledigen
- Karte-Funktion fertig machen
- Hashtag setzen und Such-Funktion hinzufügen

# Quelle:

Bilder und Video auf der Website:

- <https://www.istockphoto.com>
- <https://artlist.io/>