

1 Datenstrukturen

In diesem Kapitel wird die Strukturierung der Daten ausführlich besprochen, die notwendig sind, um die technische Umsetzung zu realisieren und das Repräsentieren der Daten gerecht realisieren zu können.

Ein Benutzer gibt verschiedene Daten von sich preis. Ausserdem enthalten die Karten und Routen Informationen und Daten, genauso wie die Wetterfunktion. Um welche Daten es sich hierbei handelt soll die folgende Tabelle darstellen:

Funktion	Daten
Benutzer	Benutzer ID Vorname, Nachname E-Mail Bild Passwort
Karte	Geodaten (Longitude, Latitude) Standort Anfangsposition Zielposition Umgebungsdaten Zeit Reitstil Bericht
Wetter	Ort Temperatur Niederschlag Windgeschwindigkeit
Nachrichten	NachrichtenID Textnachricht

Table 1: Daten innerhalb der Anwendung

1.1 XML-Dateien

Da eine Android-Applikation entwickelt werden soll, ist die Repräsentation der Daten im Zusammenhang mit dem Repräsentationsformat XML die beste Möglichkeit für dieses Projekt, da das Format in der Entwicklung bereits involviert ist und in dem Modul 'Web-basierte Anwendungen 2' praktisch umgesetzt wurde. Aus diesem Grund soll die Strukturierung der Daten anhand von XML-Schemata dargestellt und erläutert werden. Hierfür werden zunächst die XML-Dateien präsentiert werden. Im weiteren Verlauf werden zu den XML-Dateien zugehörigen XML-Schemata dargestellt und erläutert werden.

Die Sprache XML wurde entwickelt, um Daten zu beschreiben. Bevor die erstellten XML-Dateien dargestellt und erläutert werden, soll kurz auf die Wetterfunktion eingegangen werden. Die XML-Datei, die für die Wetterrepräsentation benötigt und entnommen wird kann unter der Referenz ? aufgerufen werden. Bei der vorgegebenen URL handelt es sich dabei um die Wetterdaten für die Stadt Frechen. Da die XML-Datei bereits vorgegeben ist und abgerufen werden kann, wird nicht mehr weiter darauf eingegangen.

Die XML-Datei der Route ist am umfangreichsten, weshalb diese beschrieben und erläutert werden soll. Die XML-Datei zur Nachricht und zu Profil wird nicht weiter erläutert, da es einen ähnlichen einfacheren Aufbau enthält.

```
<?xml version="1.0" encoding="UTF-8"?>
<Routen>
  <Route>
    <Id>123</Id>
    <Geodaten>
      <Startposition Längengrad="50.916672" Breitengrad="6.81667"/>
      <Zielposition Längengrad="50.916672" Breitengrad="6.81667"/>
    </Geodaten>
    <Streckenlänge Einheit="km" Länge="3.4"/>
    <Dauer Einheit="h" Stunden="2"/>
    <Bericht>
      <Startpunkt>Frechen</Startpunkt>
      <Endpunkt>Gummersbach</Endpunkt>
      <Beschreibung>Es ist eine schöne Strecke zum reiten..</Beschreibung>
      <Gefahren>Achtung ein Baum auf dem Weg eingestürzt..</Gefahren>
      <Datum>01.06.2015</Datum>
    </Bericht>
  </Route>
  <Route>
    <Id>124</Id>
    <Geodaten>
      <Startposition Längengrad="50.954672" Breitengrad="6.81623"/>
      <Zielposition Längengrad="50.432672" Breitengrad="6.84367"/>
    </Geodaten>
    <Streckenlänge Einheit="km" Länge="3.7"/>
    <Dauer Einheit="h" Stunden="2"/>
    <Bericht>
      <Startpunkt>Köln</Startpunkt>
      <Endpunkt>Gummersbach</Endpunkt>
      <Beschreibung>Super Strecke. Endlich mal etwas Neues.</Beschreibung>
      <Gefahren>Keine Gefahren unterwegs erkannt.</Gefahren>
      <Datum>03.06.2015</Datum>
    </Bericht>
  </Route>
</Routen>
```

Figure 1: XML-Datei 'Route'

Die Route enthält eine Menge von Daten. Aus der Abbildung 1 kann entnommen werden, dass das root-Element 'Routen' ist. Die Route hat ein Kindelemente und zwar 'Route'. Das Element 'Route' hat weitere Kindelemente

wie 'ID', 'Geodaten', 'Streckenlänge', 'Dauer' und 'Bericht'. Die ID soll die Route eindeutig identifizieren. Die 'Geodaten' unterteilen sich in weitere Elemente: 'Startposition' und 'Zielposition'. Diese beiden enthalten die Attribute 'Längengrad' und 'Breitengrad' für die Positionierung auf der Karte. Da der aktuelle Standort beim Öffnen der Karte automatisch angezeigt wird, sind diese Angaben Pflicht. 'Bericht' unterteilt sich nochmal in weitere Kindelemente. Diese soll Auskunft über die Route geben. Das heisst, es werden Informationen zum 'Startpunkt' und 'Endpunkt' der Route und Gefahren erfasst. Der Bericht muss ein Datum enthalten, damit die Benutzer sehen, ob es sich um einen aktuellen Bericht handelt.

```
<?xml version="1.0" encoding="utf-8"?>
<Nachrichten>
  <Nachricht>
    <ID>123</ID>
    <Absender>Michael Schmitt</Absender>
    <Datum>01.06.2015</Datum>
    <Zeit>19.03</Zeit>
    <Inhalt>Hallo liebe Hanne...</Inhalt>
  </Nachricht>
  <Nachricht>
    <ID>124</ID>
    <Absender>Ursula Frei</Absender>
    <Datum>01.06.2015</Datum>
    <Zeit>19.05</Zeit>
    <Inhalt>Hallo Hanne, wie geht es dir. Ich habe eine Frage bezüglich der neuen Route die du erstellt hast</Inhalt>
  </Nachricht>
</Nachrichten>
```

Figure 2: XML-Datei 'Nachricht'

```
<?xml version="1.0" encoding="utf-8"?>
<Profil>
  <Benutzer>
    <Vorname>Josef</Vorname>
    <Nachname>Groß</Nachname>
    <EMail>JosefGroß@JosefGroß.de</EMail>
    <Status>Profi-Reiter</Status>
  </Benutzer>
  <Benutzer>
    <Vorname>Marie</Vorname>
    <Nachname>Brings</Nachname>
    <EMail>MarieBrings@MarieBrings.de</EMail>
    <Status>Amateur-Reiter</Status>
  </Benutzer>
</Profil>
```

Figure 3: XML-Datei 'Profil'

1.2 XML-Schemata

Anhand der XML-Schemata ist die Strukturierung der Daten nochmals deutlich zu erkennen. Hier kann man die einzelnen Elemente und die zugehörigen Attribute entnehmen. So sind die Attribute Längengrad und Breitengrad beispielsweise vom Typ float.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Routenplanung">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Route" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:byte" name="Id"/>
              <xs:element name="Geodaten">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Startposition">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute type="xs:float" name="Längengrad" use="optional"/>
                            <xs:attribute type="xs:float" name="Breitengrad" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Zielposition">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute type="xs:float" name="Längengrad" use="optional"/>
                            <xs:attribute type="xs:float" name="Breitengrad" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Streckenlänge">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:string" name="Einheit" use="optional"/>
                      <xs:attribute type="xs:float" name="Länge" use="optional"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="Dauer">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:string" name="Einheit" use="optional"/>
                      <xs:attribute type="xs:byte" name="Stunden" use="optional"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="Bericht">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="Startpunkt"/>
                    <xs:element type="xs:string" name="Endpunkt"/>
                    <xs:element type="xs:string" name="Beschreibung"/>
                    <xs:element type="xs:string" name="Gefahren"/>
                    <xs:element type="xs:string" name="Datum"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figure 4: XML-Schema 'Route'

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Nachrichten">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Nachricht" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:byte" name="ID"/>
              <xs:element type="xs:string" name="Absender"/>
              <xs:element type="xs:string" name="Datum"/>
              <xs:element type="xs:float" name="Zeit"/>
              <xs:element type="xs:string" name="Inhalt"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Figure 5: XML-Schema 'Nachricht'

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Profil">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Benutzer" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="Vorname"/>
              <xs:element type="xs:string" name="Nachname"/>
              <xs:element type="xs:string" name="EMail"/>
              <xs:element type="xs:string" name="Status"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Figure 6: XML-Schema 'Profil'

1.3 Entity-Relationship-Modellierung

Die gesamte Datenmodellierung der Applikation soll durch ein Entity-Relationship-Diagramm verdeutlicht werden. Die Entität 'User' hat beispielsweise die Attribute 'User id', 'Name', 'E-Mail', 'Passwort' und kann die Entität 'Karte' aufrufen, die die Attribute 'Geodaten', 'Umgebene Reiterhöfe' und 'Rastplätze' enthält. Durch die Modellierung soll dargestellt werden, welche Entitäten vorhanden sind und wie diese in Zusammenhang stehen.

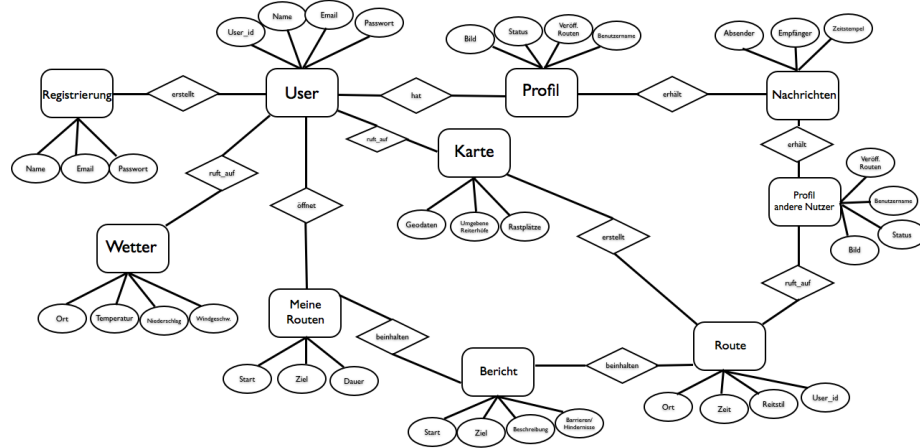


Figure 7: ER-Modellierung zur Datenstruktur