



Meilenstein 4

ENTWICKLUNGSPROJEKT INTERAKTIVE SYSTEME

ausgearbeitet von

Burcu Özata und Selin Öztürk

vorgelegt an der

TECHNISCHE HOCHSCHULE KÖLN
CAMPUS GUMMERSBACH
FAKULTÄT FÜR INFORMATIK UND
INGENIEURWISSENSCHAFTEN

im Studiengang
MEDIENINFORMATIK

Prof. Dr. Kristian Fischer
Prof. Dr. Gerhard Hartmann

Betreut von: David Bellingroth
 Franz-L. Jaspers
 Daniela Reschke

Gummersbach, 30.11.2015

Inhaltsverzeichnis

1 Datenstrukturen	2
1.1 XML-Dateien	2
1.2 XML-Schemata	3
1.3 Entity-Relationship-Modellierung	4
2 WBA-Modellierung	6
2.1 Ressorcen	6
2.1.1 Profil	6
2.1.2 Liste	6
3 Prototyp UI	8
3.1 Gestaltungslösungen	8
3.1.1 Paperbased-Prototyping	8
3.2 Fazit zu der Gestaltungslösung	13

1 Datenstrukturen

Damit dieser Projekt realisiert werden kann, wird die notwendige Strukturierung der Daten ausführlich besprochen.

Die unten dargestellte Tabelle zeigt um welche Daten es sich handelt:

Funktion	Daten
Benutzer	<ul style="list-style-type: none">• User Gruppe• Vorname, Nachname• E-Mail• Passwort• SecCode• MailChecked
User Gruppe	<ul style="list-style-type: none">• ID• Gruppen Name
Parkplätze	<ul style="list-style-type: none">• ID• SecCode• Anzahl Parkplätze• Anzahl freie Parkplätze• Geodaten (Längen-/Breitengrad)• Name Parkplatz• Entfernung zu TH• Adresse• Öffnungszeiten• Kosten• Anzeige von Frauenparkplatz• Anzeige von Behindertenparkplatz

Tabelle 1: Daten innerhalb der Anwendung

1.1 XML-Dateien

Für dieses Projekt wurde eine Android-Applikation entwickelt. Zur Übermittlung von Daten eignet sich das Repräsentationsformat XML. Dieses wird in diesem Projekt in 3 Fällen genutzt.

Beispielhaft wird das XML Dokument für einen neuen Nutzer betrachtet.

Zunächst wird die XML-Dateien präsentiert und in der Folge durch das zugehörigen XML-Schemata dargestellt und erläutert.

```
<newuser>
    <email>selin.oetztuerk@smail.th-koeln.de</email>
    <password>vollSicher</password>
    <usergroup>3</usergroup>
</newuser>
```

Tabelle 2: XML-Datei Registrierung

```

<list>
  <list>
    <parkingPlace>
      <id>1</id>
      <parkingPlaces>111</parkingPlaces>
      <freeParkingPlaces>112</freeParkingPlaces>
      <longitude>6.78363318</longitude>
      <latitude>4.13645645</latitude>
      <name>Testparkplatz 1</name>
      <distanceToFH>2.7</distanceToFH>
      <addressStreet>Teststraße</addressStreet>
      <addressNumber>74b</addressNumber>
      <addressPostalCode>55555</addressPostalCode>
      <addressCity>Kummersbach</addressCity>
      <openingTimes>
        <mondayOpening>07:00:00</mondayOpening>
        <mondayClosing>20:00:00</mondayClosing>
        <tuesdayOpening>07:00:00</tuesdayOpening>
        <tuesdayClosing>20:00:00</tuesdayClosing>
        <wednesdayOpening>07:00:00</wednesdayOpening>
        <wednesdayClosing>20:00:00</wednesdayClosing>
        <thursdayOpening>07:00:00</thursdayOpening>
        <thursdayClosing>20:00:00</thursdayClosing>
        <fridayOpening>07:00:00</fridayOpening>
        <fridayClosing>20:00:00</fridayClosing>
        <saturdayOpening>08:00:00</saturdayOpening>
        <saturdayClosing>22:00:00</saturdayClosing>
        <sundayOpening>09:00:00</sundayOpening>
        <sundayClosing>21:00:00</sundayClosing>
      </openingTimes>
      <freeTime>00:15:00</freeTime>
      <priceFirstHour>2.5</priceFirstHour>
      <priceFurtherHour>1.5</priceFurtherHour>
      <parkingPlacesForWomen>true</parkingPlacesForWomen>
      <parkingPlacesForHandicapped>true</parkingPlacesForHandicapped>
      <availableForProfessors>true</availableForProfessors>
      <availableForWorkers>true</availableForWorkers>
      <availableForStudents>true</availableForStudents>
      <availableForGuests>false</availableForGuests>
    </parkingPlace>
  </list>
</list>

```

Tabelle 3: XML-Datei Park Liste

1.2 XML-Schemata

Die XML-Schemata verdeutlichen die Strukturierung der Daten noch einmal vor.

```

<xss:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xss="http://www.w3.org/2001/XMLSchema">
  <xss:element name="newuser">
    <xss:complexType>
      <xss:sequence>
        <xss:element type="xs:string" name="email"/>
        <xss:element type="xs:string" name="password"/>
        <xss:element type="xs:int" name="usergroup"/>
      </xss:sequence>
    </xss:complexType>
  </xss:element>
</xss:schema>

```

Tabelle 4: XML-Schemata Registrierung

```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="list">
    <xs:complexType>
      <xs:sequence>
        <xsi:element name="list">
          <xsi:complexType>
            <xsi:sequence>
              <xsi:element name="parkingPlace">
                <xsi:complexType>
                  <xsi:sequence>
                    <xsi:element type="xs:int" name="id"/>
                    <xsi:element type="xs:int" name="parkingPlaces"/>
                    <xsi:element type="xs:int" name="freeParkingPlaces"/>
                    <xsi:element type="xs:double" name="longitude"/>
                    <xsi:element type="xs:double" name="latitude"/>
                    <xsi:element type="xs:string" name="name"/>
                    <xsi:element type="xs:float" name="distanceToFH"/>
                    <xsi:element type="xs:string" name="addressStreet"/>
                    <xsi:element type="xs:int" name="addressNumber"/>
                    <xsi:element type="xs:string" name="addressPostalCode"/>
                    <xsi:element type="xs:string" name="addressCity"/>
                    <xsi:element name="openingTimes">
                      <xsi:complexType>
                        <xsi:sequence>
                          <xsi:element type="xs:time" name="mondayOpening"/>
                          <xsi:element type="xs:time" name="mondayClosing"/>
                          <xsi:element type="xs:time" name="tuesdayOpening"/>
                          <xsi:element type="xs:time" name="wednesdayOpening"/>
                          <xsi:element type="xs:time" name="thursdayOpening"/>
                          <xsi:element type="xs:time" name="fridayOpening"/>
                          <xsi:element type="xs:time" name="saturdayOpening"/>
                          <xsi:element type="xs:time" name="saturdayClosing"/>
                          <xsi:element type="xs:time" name="sundayOpening"/>
                          <xsi:element type="xs:time" name="sundayClosing"/>
                        </xsi:sequence>
                      </xsi:complexType>
                    </xsi:element>
                    <xsi:element type="xs:time" name="freeTime"/>
                    <xsi:element type="xs:double" name="priceFirstHour"/>
                    <xsi:element type="xs:double" name="priceFurtherHours"/>
                    <xsi:element type="xs:string" name="parkingPlacesForWomen"/>
                    <xsi:element type="xs:string" name="parkingPlacesForHandicapped"/>
                    <xsi:element type="xs:string" name="availableForProfessors"/>
                    <xsi:element type="xs:string" name="availableForWorkers"/>
                    <xsi:element type="xs:string" name="availableForStudents"/>
                    <xsi:element type="xs:string" name="availableForGuests"/>
                  </xsi:sequence>
                </xsi:complexType>
              </xsi:element>
            </xsi:sequence>
          </xsi:complexType>
        </xsi:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Tabelle 5: XML-Datei XML-Schemata Park Liste

1.3 Entity-Relationship-Modellierung

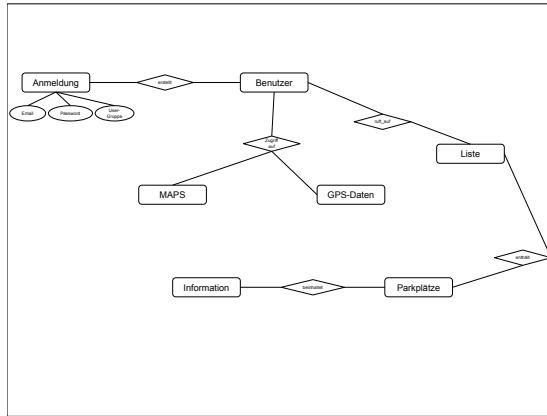


Tabelle 6: ER-Modellierung zur Datenstruktur

Durch ein Entity-Relationship-Modellierung-Diagramm soll die gesamte Datenmodellierung dargestellt werden. Betrachtet man die Entität "Anmelden", dies beinhaltet die Attribute "Usergruppe", "Email" und "Passwort" der TH-Campus Parkplatz. Die Entität "Benutzer" hat die Möglichkeit die Entität "Map's" aufzurufen, dies enthält die Attribute "GPS Daten" und "Umgebene Parkhäuser". Außerdem kann der Benutzer eine Liste mit Parkhäusern und die dazugehörige Informationen aufrufen. Die Modellierung ist eine große Hilfe um zu zeigen welche Entitäten vorhanden sind, und wie diese in Zusammenhang stehen.

2 WBA-Modellierung

2.1 Ressourcen

Die festgelegten Ressourcen:

2.1.1 Profil

Die Ressource Profil enthält alle angaben eines Users. Jeder Nutzer kann das eigene Profil erstellen, ändern und löschen. Es beinhaltet Name, E-Mail Adresse, Password und User-Gruppe. Die Usergruppe wird für die passende Auswahl von Parkplätzen benötigt.

Methoden: GET, POST, PUT, DELETE

Attribute:

- Name
- EMail
- Passwort
- User-Gruppe

Einsatz: Erstellung eines Profil, löschen des Accounts, ändern des Accounts

2.1.2 Liste

In der Liste werden die nahe stehenden Parkhäuser dargestellt und Informationen über diese angezeigt, diese können übers Display an getippt werden und so entweder Informationen angezeigt werden, oder zu diesen Parkplätzen navigiert werden.

Methoden: GET

Attribute:

- GPS-Daten
- Umgebungsinformationen
- Straßennamen

Einsatz: Zeigt auf der Karte die aktuelle Position, navigieren zu einem ausgewählten Parkplatz, zeigt Informationen über freien Parkplätzen, Öffnungszeiten und Kosten, in der Nähe stehenden freien Parkhäuser auf einer Karte anzeigen lassen.

Das System besteht aus Text Software Komponenten und aus deren Aufgaben. Um eine beliebige Route Starten zu können, wurde die im Web integrierte Map

benutzt. Hierbei werden die Komponenten auf verteilten Systemen sich aufgeteilt, unszwar in "Java-Server und Web-Server auf Server", "Parkplatz-Client auf Geräten in/an den Parkplätzenünd in Äpp auf Smarthpohne". Außerdem werden unterschiedliche Arten von Endgeräte in der App und Parkplatz-Client unterstellt. Die Kommunikation zwischen "Clients und Servern" läuft in einer Synchrone Interaktion und die "Erfassung und Anzeige von freien Parkplätzen" ebenso in der Asynchrone Interaktion ab. Die folgenden Daten wie die "Zustände über freien Parkplätze", "Daten eines neuen User" und "Daten über alle relevanten Parkplätze" wird in den einzelnen Interaktionschritten ausgetauscht. Diese Interaktion wird von der TCP/IP Middlewaresysteme unterstützt. Als Sprache benutzt man hier SQL auf MySQL Server. Um den Kern des Systems besser Verstehen zu können, sind die wesentliche Funktionalitäten bzw. Interaktionen notwendig wie beispielsweise die "Veränderung der Anzahl freier Parkplätze", "die Anzeigen von Parkplatz Informationen" und "Führen zu einem Parkplatz".

3 Prototyp UI

3.1 Gestaltungslösungen

Zur Gestaltungslösung sollte das Paperbased-Prototyping verwendet werden, was in Meilenstein 2 genau bestimmt wurde. Es ist für allgemeine und für Änderungen des Interfaces geeignet und benötigt keine Kosten, weil es sich um Zeichnungen auf Papier handelt. Grund dafür ist auch noch, dass die Möglichkeit besteht, dass Interaktionen schneller und einfacher umgesetzt werden können.

3.1.1 Paperbased-Prototyping

Der Prototyp wird auf die Eigenschaften der Anforderungsanalyse aufgebaut. Die funktionalen, organisatorischen und qualitativen Funktionen wurden versucht richtig umzusetzen.

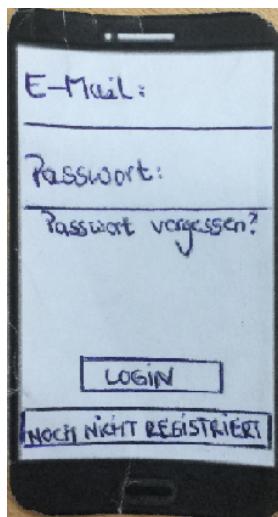


Abbildung 1: Paperbased Prototyp - Login

Wenn der Benutzer bereits registriert ist, kann er sich durch die Eingabe seines TH S-Mails und Passwort einloggen (O70). Da der Besucher kein Mitglieder der TH ist, muss er sich mit seiner normalen E-Mail einloggen oder gegenbenfalls registrieren. Ist der Benutzer nicht registriert, so muss er zunächst eine Registrierung durchführen. Ziel war, dass der Benutzer sich sehr schnell und leicht in das System registrieren kann.

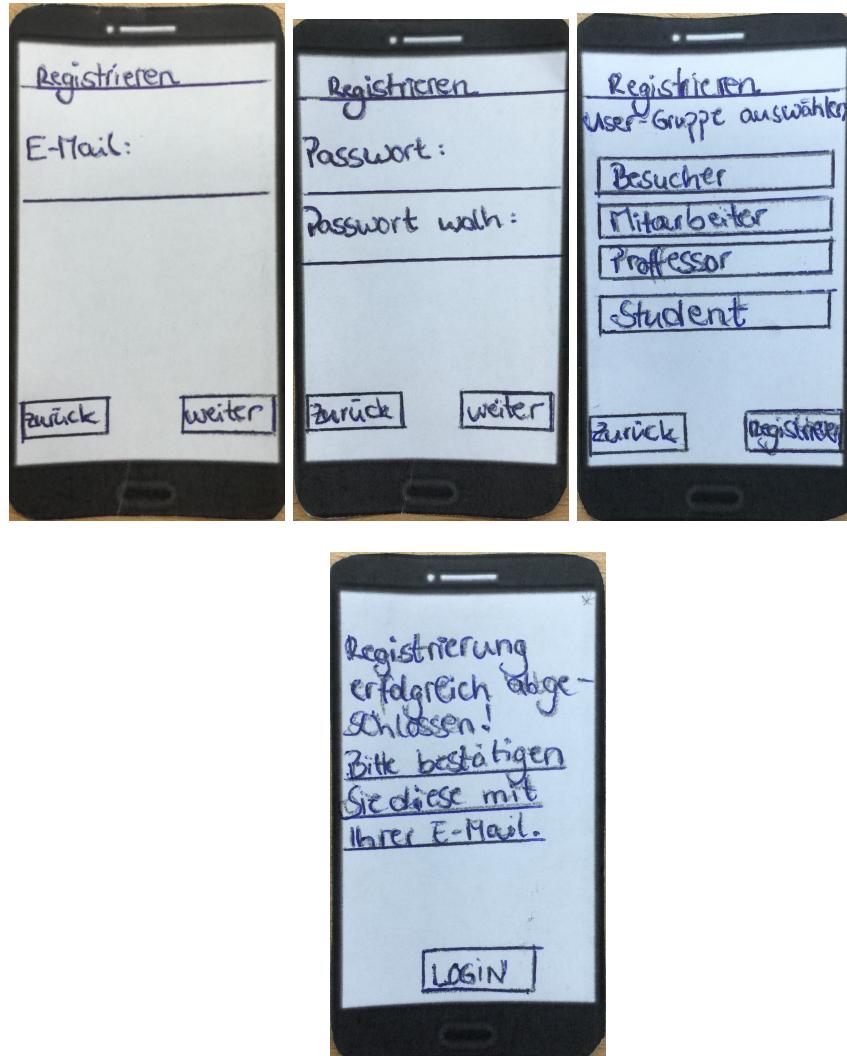


Abbildung 2: Paperbased Prototyp - Registrierung

Die Registrierung erfolgt durch die Eingabe von TH S-Mail, Passwort und User-Gruppe.

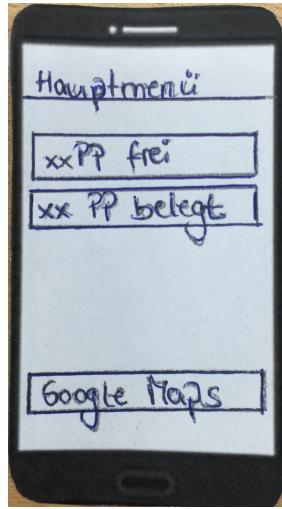


Abbildung 3: Paperbased Prototyp - Hauptmenü

Das Hauptmenü besteht aus einer Anzeige, was die Belegtheit der Parkplätze darstellt und einem Button "GoogleMap" (F40/Q10). Das Button steht für einen der Funktionen und man sollte durch dieses das Nutzungsziel erreichen. Q30 setzt eine einfache und ersichtliche Benutzeroberfläche vorraus. Da Q40 eine nachvollziehbare und übersichtliche Darstellung der angezeigten Informationen bereitstellen muss, werden die Buttons eine einfache Darstellung mit verständlichen Titeln haben. Damit der Benutzer schnell zum Nutzungsziel geführt werden kann, werden die Buttons eine einfache Struktur und Größe zum effektiven Bedienung haben.

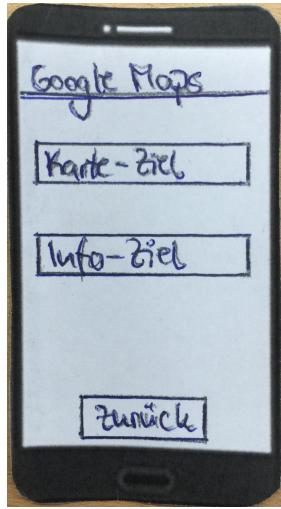


Abbildung 4: Paperbased Prototyp - Google Map

Angenommen wenn der Button "GoogleMap" betätigt wird, erscheinen im Display zwei weitere Buttons. "Karte-Ziel" und "Info-Ziel" geben nähere Informationen zum Endziel.



Abbildung 5: Paperbased Prototyp - Karte

Tätigt der Benutzer einer der SZiele”, so besteht die Möglichkeit die Navigation zu starten. So gelangt der Benutzer von seinem Standort aus zu einer der Parkplätze, die in der Nähe der Th liegen (O20).

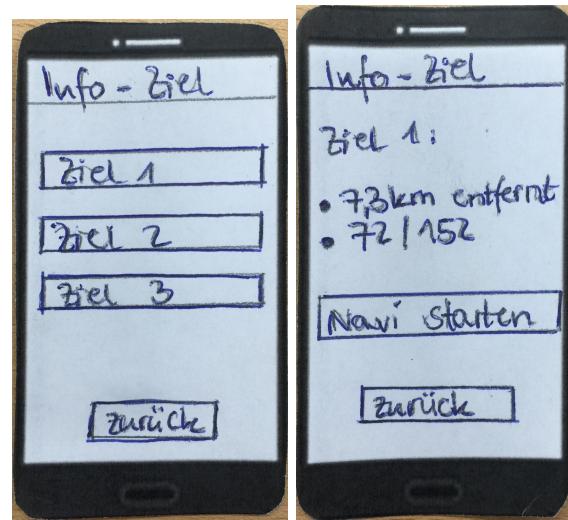


Abbildung 6: Paperbased Prototyp - Informationen von Ziel

Der Benutzer wird schriftlich und auditiv zum Ziel geführt. Man hat auch die Möglichkeit, die Navigation jederzeit zu beenden (F10).

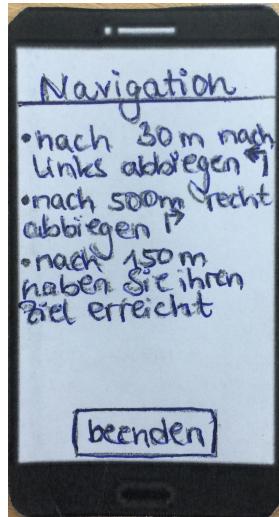


Abbildung 7: Paperbased Prototyp - Navigation

Bevor der Benutzer sich zum Ziel führen lässt, hat er die Gelegenheit sich über die einzelnen Zielmöglichkeiten zu informieren. Man hat die Information zur Entfernung, Preis und Belegtheit.

3.2 Fazit zu der Gestaltungslösung

Die Anforderungen, die in der Anforderungsanalyse festgelegt wurden, werden in der Gestaltungslösung erfüllt.