

NTB - Interstaatliche Hochschule für Technik Buchs

IuK_IV
Semesterprojekt Webapplikation

Studierende: Mosberger Benjamin, benjamin.mosberger@ntb.ch

Tobias Schoch, tobias.schoch@ntb.ch

Referent: Studer Martin, martin.studer@htwchur.ch

Datum: FS 2016

Inhaltsverzeichnis

1	Projektidee	3
2	Projektskizze	4
3	Anforderungsdokumentation	5
3.1	Anforderungsliste	5
3.2	Datenmodell	6
3.3	Mockup	7
4	Architektur	9
4.1	Allgemein	9
4.2	Facebook-Login	9
4.3	Maps	11
5	Test	13
5.1	Test 1	13
5.2	Test 2	13
5.3	Test 3	13
5.4	Test 4	13
5.5	Test 5	13
5.6	Test 6	13
5.7	Test 7	13
5.8	Test 8	13
5.9	Test 9	13
6	Erfahrungen	14

1 Projektidee

Ein Webtool zur Suche von Ersatzteilen oder Dienstleistungen im Zusammenhang mit Autos soll erstellt werden. Dabei soll der Benutzer sich mit Facebook einloggen können, um allfällige Bewertungen abzugeben. Das Facebook-Login verhindert, dass eine Person zu viele gute oder schlechte Bewertungen verteilt. Der Benutzer kann nun eine Postleitzahl, eine maximale Reichweite und Stichwörter eingeben, daraus wird dem Benutzer eine Liste mit den verschiedenen Anbietern angezeigt mit einer kurzen Beschreibung, durch anklicken wird das komplette Profil des Anbieters angezeigt und es kann Kontakt aufgenommen werden. Die Liste der Anbieter Dienstleistungen sollen sich ebenfalls mit Facebook einloggen können und durch Anwählen eines Buttons zum Backend der Webapp gelangen, wo Sie Ihre Daten selber eingeben können. Es soll auch möglich sein, Fotos Ihrer Arbeiten zur Vorschau zur Verfügung zu stellen.

2 Projektskizze

Ziel:

Mit dem Dienstleistungsportal soll dem Nutzer die Suche nach geeigneten Anbietern im Automobilbereich in der näheren Umgebung massgeblich vereinfacht werden. Im Vergleich zu heute, wo die Suche hauptsächlich über Facebook und Empfehlungen von Fremden stattfindet soll ausserdem ein seriöses Bewertungssystem eingeführt werden.

Umsetzung:

Die Webapplikation soll mit Bootstrap erstellt werden. Zum Einloggen soll der Facebook-Account verwendet werden. Damit wird auch das Bewertungssystem der Anbieter überwacht, jeder Nutzer soll einen Anbieter nur maximal 1 mal bewerten können. Likes sollen auf dieser Plattform keinen Stellenwert haben.

Laufzeit:

Das Projekt erstreckt sich über das Frühlingssemester 2016. Das Abgabedatum ist der 03.06.2016.

Finanzierung:

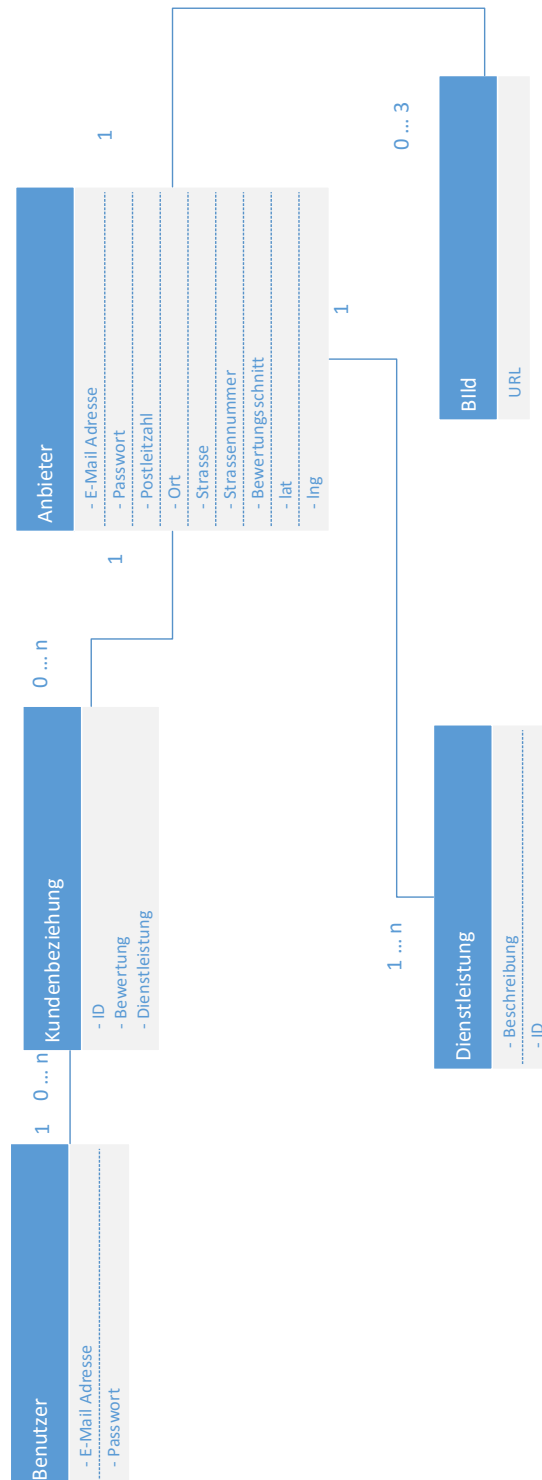
Das Projekt soll finanzielle Mittel auskommen. Es sind lediglich 60 Arbeitsstunden pro Student gerechnet.

3 Anforderungsdokumentation

3.1 Anforderungsliste

- A.1 Der Kunde kann sich mit dem Facebook Account anmelden
- A.2 Der Kunde kann sein Profil editieren
- A.3 Der Dienstleister kann sein Profil editieren
- A.4 Der Kunde kann nach Dienstleistung suchen
- A.5 Der Kunde kann nach Radius suchen
- A.6 Der Kunde kann den Dienstleister mit 1-5 bewerten
- A.7 Der Kunde kann zwischen den Ansichten Liste und Map wählen
- A.8 Der Dienstleister kann 0-3 Bilder hoch laden
- A.9 Der Dienstleister kann 1-3 Dienstleistungen angeben

3.2 Datenmodell



3.3 Mockup

A Web Page

http://

anmelden login

Autozubehörverzeichnis

facebook

Email

Password

Log In

A Web Page

http://

anmelden login

Autozubehörverzeichnis

Anmelden:

Name:

E-Mail:

Passwort:

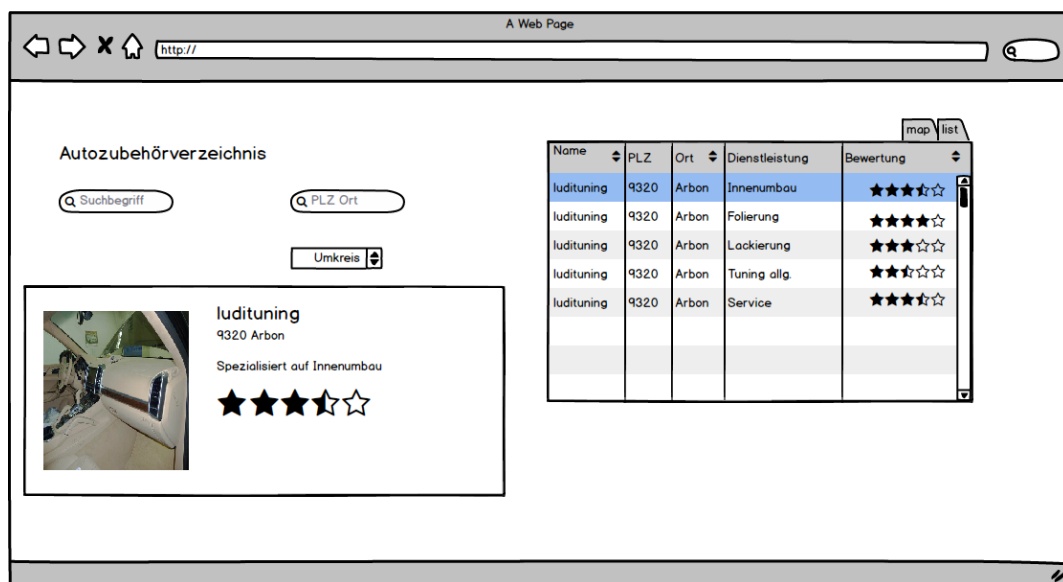
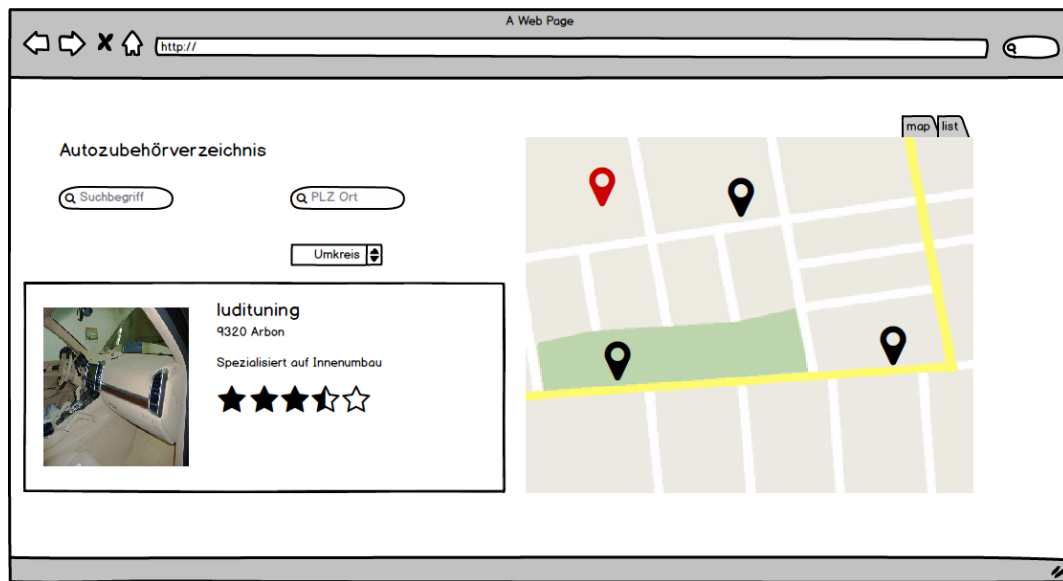
Passwort wiederholen:

PLZ:

Ort:

Dienstleistungen:

Bilder:



4 Architektur

4.1 Allgemein

Die Webapplikation wurde mit dem Framework Bootstrap erstellt. Um die Benutzung des JavaScript Codes zu vereinfachen, wurde die Bibliothek JQuery verwendet. Für PHP wurde kein Framework verwendet. Im folgenden ist der Facebook-Login und die Verwendung von Googlemaps in unserer Webapplikation beschrieben.

Links:

GitHub-Repo: <https://github.com/tschoch/webapplikation>

Webapplikation: http://614879-13.web.fh-htwchur.ch/projekt-web/Website/search_list.php

4.2 Facebook-Login

Der Facebook login wurde mit dem PHP SDK [1] von Facebook realisiert. Um ein Facebook Login auf der eigenen Seite zu ermöglichen, muss der Entwickler ein Facebook Developer Account erstellen. Danach kann die App auf der Facebook Developerseite erstellt werden und es wird eine App-ID und ein App-Secret generiert. Diese beiden credentials müssen in der Anfrage auf den Facebookserver mitgegeben werden, sonst funktioniert die Anfrage nicht. Das Hauptskript für den Facebooklogin ist im folgenden beschrieben.

```
<?php
// einbinden des Autoloaders
require_once 'autoload.php';

...
// Eingabe der App ID & des App Secret
FacebookSession::setDefaultApplication( 'App_ID', 'App_Secret' );

// erzeugen einer FacebookRedirectLoginHelper Instanz
$helper = new FacebookRedirectLoginHelper(
    'http://localhost/projekt_web/phplogin/fbconfig.php' );
try {

    //behandelt Facebook Rückmeldung und gibt Facebook Session zurück
    $session = $helper->getSessionFromRedirect();
} catch( FacebookRequestException $ex ) {
    // Wenn Facebook einen Error zurück gibt
} catch( Exception $ex ) {
    // Wenn die Überprüfung fehlschlägt oder andere Probleme
}

// Überprüfen ob Session vorhanden
if ( isset( $session ) ) {
```

```
// graph api anfrage für user data
$request = new FacebookRequest(
$request = new FacebookRequest(
    $session, 'GET', '/me?locale=en_US&fields=id,name,email');
$response = $request->execute();

// Antwort mit Daten
$graphObject = $response->getGraphObject();
    $fbid = $graphObject->getProperty('id');
    $fbfullname = $graphObject->getProperty('name');
    $femail = $graphObject->getProperty('email');

// Variablen in Session variablen abspeichern
$_SESSION['FBID'] = $fbid;
$_SESSION['FULLNAME'] = $fbfullname;
$_SESSION['EMAIL'] = $femail;

// DB aktualisieren
checkuser($fbid, $fbfullname, $femail);

//header location after session
header("Location:_../Website/search_list.php");

// wenn noch keine Session
} else {

//Umleitung zur Loginseite, definition der genutzten daten
$loginUrl = $helper->getLoginUrl( array('scope' => 'email'));
header("Location:_".$loginUrl);
}
?>
```

Listing 4.1: fbconfig.php

Am Anfang einer Seite wird geprüft ob man bereits bei Facebook eingeloggt ist. Dies geschieht indem geprüft wird ob die Session variable FBID bereits gesetzt ist.

```
<?php
//session start
session_start();
...
//prüfe ob eingeloggt
if ( \$_SESSION[ 'FBID' ] ):
...
php else :
...
php endif
?>
```

Listing 4.2: search_list.php

4.3 Maps

```
<!DOCTYPE html>
<html>
<head>
  <style type="text/css">
    html, body { height: 100%; margin: 0; padding: 0; }
    #map { height: 100%; }
  </style>
</head>
<body>
  <div id="map"></div>
  <script type="text/javascript">
var map;
function initMap() {
  map = new google.maps.Map(document.getElementById( 'map' ), {
    center: {lat: -34.397, lng: 150.644},
    zoom: 8
  });
}
  </script>
  <script async defer
    src="https://maps.googleapis.com/maps/api/js?
key=YOUR_API_KEY&callback=initMap">
  </script>
</body>
</html>
```

Für die Suche auf der Karte wird Google Map verwendet, auf dem vorhergehendem Bild ist die Erstellung einer einfachen Karte zu sehen. Wichtig ist, damit man dies tun kann, einen Entwickler Account bei Google zu machen, damit kann man sich dann einen API-Key für seine App generieren lassen.

Um die Adressen aus der Datenbank auf der Map darzustellen muss aus der Adresse die Längen- und Breitengrade herausgefunden werden. Erneut liefert Google dazu ein API, genannt Geocode, welches einem dies abnimmt. Dabei wird die Adresse von welcher die Geolocation gesucht wird dem Parameter `address` in der Adresszeile des

Browsers mitgegeben: `https://maps.googleapis.com/maps/api/geocode/json?address=Pulverm%C3%BChlestrasse+57,+7000+Chur` Die Antwort kommt in diesem Fall mit JSON zurück, alternativ wäre noch XML verfügbar. Um diese Antwort in JavaScript zu verarbeiten wird

```
$.getJSON('http://maps.googleapis.com/maps/api/geocode/json? +
address='+adresse+'&sensor=false ', null, function (data) {
    var p = data.results[0].geometry.location
    var latlng = new google.maps.LatLng(p.lat , p.lng );
    new google.maps.Marker({
        position: latlng ,
        map: map
    });
});
```

Listing 4.3: script.php

aufgerufen. Mit dieser Funktion wird dynamisch ein Marker in der Map gesetzt.

5 Test

	T1	T2	T3	T4	T5	T6	T7	T8	T9
A.1	X								
A.2		X							
A.3			X						
A.4				X					
A.5					X				
A.6						X			
A.7							X		
A.8								X	
A.9									X

Tabelle 5.1: Tests

5.1 Test 1

5.2 Test 2

5.3 Test 3

5.4 Test 4

5.5 Test 5

5.6 Test 6

5.7 Test 7

5.8 Test 8

5.9 Test 9

6 Erfahrungen

Literaturverzeichnis

- [1] Github. facebook-php-sdk-v4 URL: <https://github.com/facebook/facebook-php-sdk-v4/>
[Stand 6.4.2016].