

Mikko Laitinen

WEB-POHJAINEN ESTEETTÖMYYSKARTOITUSTYÖKALU

Tietotekniikan koulutusohjelma  
Ohjelmoinnin suuntautumisvaihtoehto  
2014

## WEB-POHJAINEN ESTEETTÖMYYSKARTOITUSTYÖKALU

Laitinen, Mikko  
Satakunnan ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Huhtikuu 2014  
Ohjaaja: Aarinen, Reino  
Sivumäärä: (44)  
Liitteitä: 0

Asiasanat: MySQL, PHP, jQuery Mobile, esteettömyyskartoitus

---

Opinnäytetyön aiheena oli suunnitella ja toteuttaa web-pohjainen esteettömyyskartoitussovellus Satakunnan ammattikorkeakoulun esteettömyys ja saavutettavuus tutkimusryhmän käyttöön. Sovelluksen tuli toimia kaikilla yleisillä tableteilla ja älypuhelimilla.

Sovellus kehitettiin dynaamiseksi, jotta jatkokehitys olisi helpompaa. Dynaamisuuden vuoksi tietokannan rakenne ja käytetyt sql-kyselyt nousivat tärkeimmiksi tekijöiksi tässä sovelluksessa.

Sovelluksen toteutus on kuvattuna asiakkaan raportissa, mikä pidetään salaisena.

Lopullinen esteettömyyskartoitussovellus täyttää kaikki toiminnallisuudet, joita siltä aluksi lähdettiin hakemaan. Sovellus toimii eri mobiilialustoilla ja sillä saa tehtyä sulavasti esteettömyyskartoituksia. Sovellus sisältää pääkäyttäjälle ja peruskäyttäjälle erilaiset toiminnallisuudet. Kartoitussovellus sisältää kaikki tarvittavat kartoituslomakkeet julkisen rakennetun ympäristön ja liikuntapaikkojen kartoittamiseen. Sähköinen kartoitussovellus nopeuttaa ja helpottaa esteettömyyskartoituksen valmistelua, kartoituksen tekemistä sekä raportointivaihetta. Lisäksi työkalun käyttö vähentää paperinkulutusta ja varmistaa, että kartoittajalla on mukanaan aina tarvittavat kaavakkeet.

Tässä kirjallisessa osassa perehdytään sovelluksen käyttöliittymään, toimintoihin ja tietokantaan koodi- ja kuvaesimerkkien avulla.

## WEB-BASED ACCESSIBILITY SURVEY TOOL

Laitinen, Mikko

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Information Technology

April 2014

Supervisor: Aarinen, Reino

Number of pages: 44

Appendices: 0

Keywords: MySQL, PHP, jQuery Mobile, accessibility survey

---

The purpose of this thesis was to develop a web-based accessibility survey tool for the accessibility research and development group at Satakunta University of Applied Sciences. This application functions with all popular smartphones and tablets.

This application was developed through dynamic to help further development. The structure of the database and SQL-queries used became the most important factors in developing this software.

Development of this software is described in customer's version, which is kept secret.

The final web-based accessibility survey tool fulfills all functional requirements that were set at the beginning. The survey tool operates in different mobile platform and it functions smoothly. The tool includes different functions for the administrator and normal user. The survey tool includes all the forms needed to implement the accessibility survey in build surroundings and sport facilities. The electric survey tool clarifies and facilitates the assessment work and it includes all the documents needed replacing the prints and papers.

The written part of this thesis will focus on: user interface, functions, and database. Coding, and picture examples will also be provided.

# SISÄLLYS

1	JOHDANTO.....	5
2	JQUERY MOBILE .....	6
2.1	Yleistä .....	6
2.2	Käyttöönotto .....	6
2.3	Data-roolit ja syöttökentät.....	7
2.4	Metodit.....	12
2.4.1	click()- metodi .....	12
2.4.2	blur()- metodi .....	12
2.4.3	val()- metodi .....	13
2.4.4	focus()- metodi .....	14
2.4.5	show()- ja hide()- metodit.....	14
2.4.6	replaceWith()- metodi .....	15
3	PHP.....	16
3.1	Yleistä .....	16
3.2	SQL-lauseet.....	16
3.2.1	SELECT- lause .....	16
3.2.2	INSERT INTO- lause .....	17
3.2.3	UPDATE- lause.....	18
3.2.4	DELETE- lause .....	18
4	KÄYTTÖLIITTYMÄ .....	19
5	TIETOKANTA .....	20
6	YHTEENVETO .....	21
	LÄHTEET.....	22

## 1 JOHDANTO

Satakunnan ammattikorkeakoulussa (SAMK) on tehty pitkäjänteistä ja päämäärätietoista esteettömyyden kehitys- ja tutkimustyötä. Kehitystyössä huomiota kiinnitetään fyysisiin olosuhteisiin ja sosiaaliseen ympäristöön levittämällä esteettömyystietoisuutta ja kartoittamalla toimintaympäristöjen esteettömyyttä./1/. Vuoden 2013 alusta SAMKin esteettömyys ja saavutettavuus työryhmä on panostanut entisestään fyysisen ympäristön esteettömyyden edistämiseen toteuttamalla lukuisia esteettömyyskartoituksia, kouluttamalla esteettömyyskartoittajia sekä kehittämällä uusia työkaluja ja kriteerejä esteettömyyden kartoittamiseen.

Tämän opinnäytetyön tavoitteena on tuottaa SAMKin esteettömyyden ja saavutettavuuden tutkimusryhmälle vaatimukset täyttävä esteettömyyden kartoitus- ja raportointisovellus. Työn tarpeellisuutta perustellaan tilojen esteettömyyskartoitukseen tarvittavien lomakkeiden laajuudella. Jopa pienet tilat tarvitsevat useita lomakkeita. Lomakkeiden täyttäminen ja tulostaminen on aikaa vievää ja siihen kaivataan parempaa ratkaisua. Merkittävänä vaikuttana asiana on myös asioiden toistuvuus raporteja kirjoitettaessa. Tietyt normit ja lauseet toistuvat lomakkeissa, siksi automatisointi nopeuttaa prosessia merkittävästi.

Vaatimuksena työlle on työkalu, joka mahdollistaa kartoituksen sekä usean lomakkeen luonnin kyseistä kartoituskertaa kohti. Tiedot tullaan tallentamaan tietokantaan. Koska esteettömyyskartoittajien halutaan pystyvän jatkokehittämään sovellusta luodaan sovellukselle editori, mikä mahdollistaa kartoittajien itse lisätä lomakkeita ja muokata niitä.

Tässä kirjallisessa osassa perehdytään siihen, miten PHP:ta ja MySQL tietokantaa voidaan hyödyntää, kun halutaan luoda dynaaminen järjestelmä. Työssä perehdytään myös käyttöliittymän luonnissa käytettyyn jQuery Mobileen.

## 2 JQUERY MOBILE

### 2.1 Yleistä

jQuery Mobile on HTML5-pohjainen käyttöliittymä, joka on suunniteltu tarpeisiin mukautuvia web sivuja ja sovelluksia varten. jQuery Mobilella luotuja sovelluksia voidaan käyttää kaikilla älypuhelimilla, tableteilla ja pöytäkoneilla. Se tarjoaa siirtymiä, kosketustoimintoja ja monia vimpaimia. Sen kevyt koodi on rakennettu asteittain parantamalla. Lisäksi jQuery Mobilessa on helppo vaihtaa teemoja./2/

Tässä opinnäytetyön kirjallisessa osassa ei käydä läpi kaikkia jQuery Mobilen mahdollisuuksia. Tämän sijaan pyritään keskittymään sovelluksessa käytettyihin asioihin.

### 2.2 Käyttöönotto

jQuery Mobile otetaan käyttöön määrittelemällä tarvittavat Javascript- sekä CSS tiedostot HTML-dokumentin header-osioon alla olevan esimerkin mukaisesti.

```
<!DOCTYPE html>
<html>
<head>
<title>jQuery Mobile Application</title>
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.0a4.1/
jquery.mobile-1.0a4.1.min.css" />
<script src="http://code.jquery.com/jquery-1.5.2.min.js"></script>
<script src="http://code.jquery.com/mobile/1.0a4.1/jquery.mobile-
1.0a4.1.min.js">
</script>
</head>
<body>
</body>
</html>
```

## 2.3 Data-roolit ja syöttökentät

jQuery Mobile käyttää kyseisiä attribuutteja generoidakseen tarpeellisen sisällön käytettävälle elementille.

### Tilanne ennen jQuery Mobilen initialisointia

```
<!-- begin first page -->
<section id="page1" data-role="page">
<header data-role="header"><h1>jQuery Mobile</h1></header>
<div data-role="content">
<p>First page!</p>
</div>
<footer data-role="footer"><h1>O'Reilly</h1></footer>
</section>
<!-- end first page -->
```

### Tilanne initialisoinnin jälkeen

```
<!-- begin first page -->
<section class="ui-page ui-body-c ui-page-active"
data-url="page1"
id="page1"
data-role="page">
<header role="banner"
class="ui-bar-a ui-header"
data-role="header">
<h1 aria-level="1"
role="heading"
tabindex="0"
class="ui-title">Header</h1>
</header>
<div role="main" data-role="content" class="ui-content">
<p>First page!</p>
</div>
<footer role="contentinfo"
class="ui-bar-a ui-footer"
data-role="footer">
<h1 aria-level="1"
role="heading"
tabindex="0"
```

```

class="ui-title">Footer</h1>
</footer>
</section>
<!-- end first page -->

```

Data-role="page"- attribuutti esitetään section-tagin yhteydessä ja sillä kuvastetaan yhdelle sivulle kuuluvaa aluetta. Data-role="header" määrittää otsikko rivin. Data-role="content" kattaa varsinaisen sivulle tulevan sisällön. Data-role="footer" määrittää sivukohtaisen alaviitteen.

Listojen käsittelyyn jQuery Mobile tarjoaa data-role attribuutin "listview".

**Esimerkki listan käytöstä.**

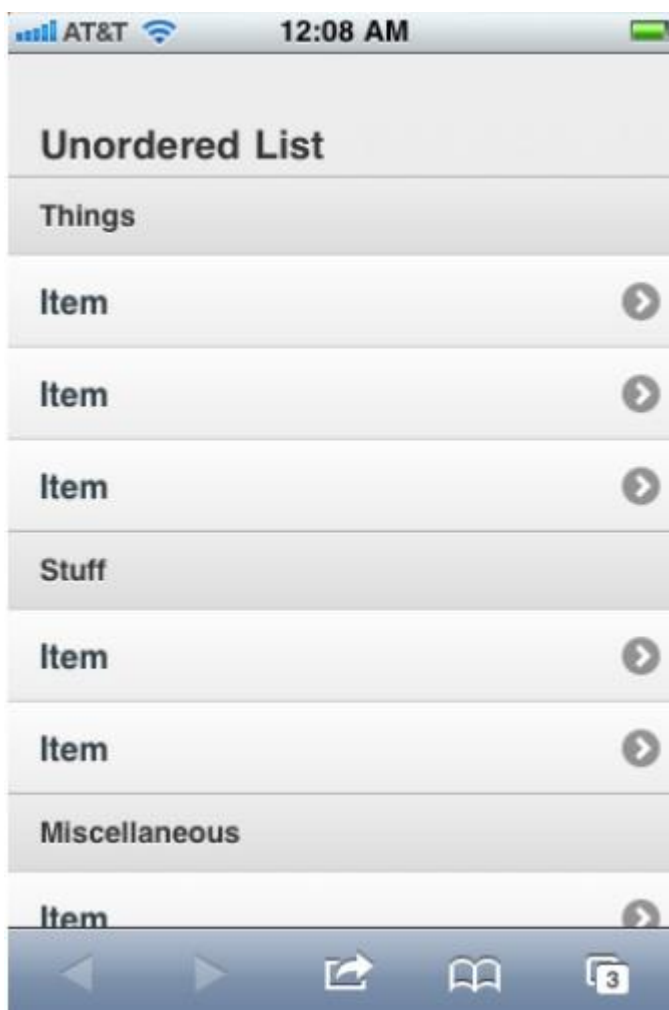
```

<ul data-role="listview">
<li data-role="divider">Things</li>
<li><a href="#">Item</a></li>
<li><a href="#">Item</a></li>
<li><a href="#">Item</a></li>
<li data-role="divider">Stuff</li>
<li><a href="#">Item</a></li>
<li data-role="divider">Miscellaneous</li>
<li><a href="#">Item</a></li>
<li><a href="#">Item</a></li>
<li><a href="#">Item</a></li>
</ul>
<h3>Ordered List</h3>
<ol data-role="listview">
<li data-role="divider">Group</li>
<li><a href="#">Item</a></li>
<li><a href="#">Item</a></li>
<li data-role="divider">Group</li>
<li><a href="#">Item</a></li>
<li data-role="divider">Group</li>
<li><a href="#">Item</a></li>
<li><a href="#">Item</a></li>
<li><a href="#">Item</a></li>
<li><a href="#">Item</a></li>
</ol>

```



Data-role="listview" attribuutti kertoo, että kyseessä on lista. Data-role="divider" auttaa luokittelemaan listan sisältöä.



Kuva 1. Listview (Reid 2011, 27)

Linkkien yhteydessä käytettävä attribuutti on data-role="button". Mikäli kyseisiä nappeja halutaan ryhmitellä, niin hyödynnetään data-attribuuttia data-role="controlgroup". Data-type="horizontal"- attribuutti määrittää, että ryhmän jäsenet listataan vaakatasossa. Data-inline="true"-attribuuttilla määritetään, ettei haluta ryhmän vievän koko tilaa vaakatasossa. Sen sijaan viedään vain nappien tarvitsema tila. Data-icon attribuutilla määritetään napin ikoni. Data-iconpos="left"-attribuutilla kerrotaan, että napin ikoni halutaan tekstin vasemmalle puolelle.

```
<h3>Button Icons</h3>
```

```
<div data-role="controlgroup" data-type="horizontal" data-
inline="true">
```

```

<a href="#" data-role="button" data-icon="home" data-
iconpos="left">Home</a>
<a href="#" data-role="button" data-icon="gear"
data-iconpos="left">Configure</a>
<a href="#" data-role="button" data-icon="search"
data-iconpos="left">Search</a>
</div>

```



Kuva 2. Listview (Reid 2011, 48)

Lomakkeiden luontiin jQuery Mobilesta löytyy tarvittavat syöttökentät eli inputit. Alapuolella näkyy, checkbox- sekä radiobutton-inputtien esitystapa. Radiobuttoneissa tulee huomioida, että kaikki kolme radiobutton-inputtia sisältävät saman nimen. Eli valinta tehdään tällöin ranskan,italian tai kreikan välillä.

```

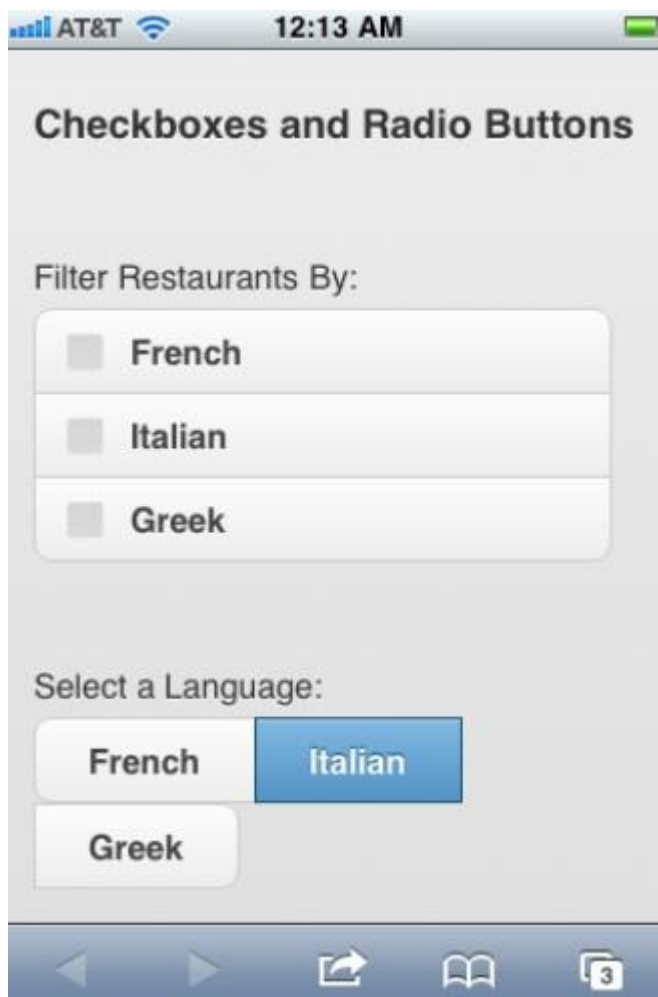
<h3>Checkboxes and Radio Buttons</h3>
<form id="myform" action="formprocessor.php" method="post">
<div data-role="fieldcontain">
<fieldset data-role="controlgroup">
<legend>Filter Restaurants By:</legend>
<input type="checkbox" name="checkbox-french" id="checkbox-french">
<label for="checkbox-french">French</label>
<input type="checkbox" name="checkbox-italian" id="checkbox-
italian">

```

```

<label for="checkbox-italian">Italian</label>
<input type="checkbox" name="checkbox-greek" id="checkbox-greek">
<label for="checkbox-greek">Greek</label>
</fieldset>
<div data-role="fieldcontain">
<fieldset data-role="controlgroup" data-type="horizontal">
<legend>Select a Language:</legend>
<input type="radio" name="radio-language" id="radio-french">
<label for="radio-french">French</label>
<input type="radio" name="radio-language" id="radio-italian"
checked="checked">
<label for="radio-italian">Italian</label>
<input type="radio" name="radio-language" id="radio-greek">
<label for="radio-greek">Greek</label>
</fieldset>
</div>
</form>

```



Kuva 3. Listview (Reid 2011, 51)

## 2.4 Metodit

Sovelluksen toimivuuden takaamiseksi ohjelmisto hyödyntää jQueryn tarjoamia metodeja. Näillä pyritään reagoimaan käyttäjän tekemiin toimintoihin ja laukaistaan erilaisia toimintoja javascriptin kautta.

### 2.4.1 click()- metodi

Kyseinen metodi asetetaan mihin tahansa HTML-elementtiin. Metodi suoritetaan, jos kohdetta klikataan. Metodi suoritetaan vasta, kun painike vapautetaan./3/

HTML koodi, johon sidotaan tapahtuman seuraaja metodi:

```
<div id="kohde">
  Paina tästä
</div>
<div id="toinen">
  Tälle ei ole triggeriä
</div>
```

Nyt tehdään javascriptiin click- metodi, joka suorittaa tulosteen ruudulle:

```
$('#kohde').click(function() {
  alert('kohde divin sisältöä painettiin.');
```

### 2.4.2 blur()- metodi

Blur metodia käytetään hyödyntämään tilannetta, jossa kohde menettää "fokuksen" eli kohde ei ole enää valittuna. Mikäli kohteesta siirrytään pois suoritetaan elementtiin sidottu blur- metodi. Tässä sovelluksessa metodia hyödynnetään, kun tekstikenttä menettää "fokuksen"./4/

HTML-merkintä:

```
<form>
<input id="kohde" type="text" value="testil" />
</form>
```

Nyt tehdään javascriptiin blur- metodi, joka suorittaa tulosteen ruudulle:

```
$('#kohde').blur(function() {
alert('blur metodi() kutsuttu');
});
```

### 2.4.3 val()- metodi

Metodia käytetään, kun halutaan saada lomakkeen elementin arvo talteen. Tässä sovelluksessa metodia käytetään tallettamaan syöttökenttien sisältöä muuttujiin./5/

Esimerkki val()-metodin käytöstä:

```
<!DOCTYPE html>
<html>
<head>
<style>
button { margin:4px; cursor:pointer; }
input { margin:4px; color:blue; }
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
</head>
<body>
<div>
<button>Aseta</button>
<button>tekstikentän</button>

<button>Sisältö</button>
</div>
<input type="text" value="Paina jotain napeista asettaaksesi teksti tähän" />
<script>
$("button").click(function () {
var text = $(this).text();
$("input").val(text);
});
</script>
</body>
</html>
```

Tässä esimerkissä HTML-merkkkaus, tyylitiedosto sekä javascript on määritetty samaan tiedostoon. Tätä tapaa ei suositella, mutta on myös mahdollinen. Mikäli käyttäjä painaa jotain napeista asetetaan tekstikenttään napin sisältämä arvo eli teksti, mikä napissa näkyy.

#### 2.4.4 focus()- metodi

Metodi suoritetaan, jos elementti saa fokuksen eli tulee aktiiviseksi. Metodia voidaan myös käyttää alapuolen esimerkin mukaisesti./6/

```
<form>
<input id="kohde" type="text" value="testi" />
</form>
<div id="kohde2">
Trigger the handler
</div>
```

**Fokus-metodi sidottuna inputtiin:**

```
$('#target').focus(function() {
  alert('focus() metodi kutsuttu');
});
```

**Kohde2 klikattaessa focus siirretään tekstikenttään:**

```
$('#kohde2').click(function() {
  $('#kohde').focus();
});
```

#### 2.4.5 show()- ja hide()- metodit

Metodit toimivat toistensa vastakohtina. Show()- metodilla voidaan näyttää piilossaolevia elementtejä ja hide- metodilla piilottaa näkyvissä olevia.

**Painettaessa elementtiä, jonka id on ”triggerdivshow” näytetään kuva id:n sisältö:**

```
<div id="triggerdivshow">
  Push
</div>
$( "#triggerdivshow" ).click(function() {
  $( "#kuva" ).show( "slow", function() {

  });
});
```

**Painettaessa elementtiä, jonka id on ”triggerdivhide” piilotetaan kuva id:n sisältö:**

**HTML- osio:**

```
<div id="triggerdivhide">
  Push
</div>

```

**jQuery- osio:**

```
$( "#triggerdivhide" ).click(function() {
  $( "#kuva" ).hide( "slow", function() {
  });
});
```

```
});
```

#### 2.4.6 replaceWith()- metodi

Metodin avulla elementin sisältöä voidaan muuttaa uudeksi poistamalla DOM rakenteesta ja asettamalla tilalle uutta sisältöä. /7/

Korvataan divin sisältö uudella sisällöllä replaceWith metodilla:

**HTML-osio:**

```
<div id="subquestion1">
<span class="subquestion"> Subquestion : kysymyksen nimi</span>
</div>
```

**jQuery-osio:**

```
$('#subquestion1' ).replaceWith("<textarea id ='subquestion#id'
placeholder='kysymyksen nimi'></ textarea>");
```

**Tulos:**

```
<div id="subquestion1">
<textarea id ='subquestion#id' placeholder='kysymyksen nimi'>
</ textarea>
</div>
```

## 3 PHP

### 3.1 Yleistä

PHP(*Hypertext Preprocessor*) on laajalti käytetty avoimeen lähdekoodiin perustuva skriptauskieli, joka soveltuu erityisesti web kehitykseen. Päätaavoite kielellä on mahdollistaa web kehittäjien tehdä dynaamisesti generoituja sivuja nopeasti./8/

### 3.2 SQL-lauseet

Tietokantafunktioita voidaan käyttää joko proseduraalisesti tai oliopohjaisella tyyllillä. Tässä opinnäytteessä keskitytään proseduraaliseen tyyliin.

#### 3.2.1 SELECT- lause

Funktiota käytetään, kun halutaan valita tietoa tietokannasta. Tieto talletetaan usein muuttujaan taulukko muodossa. Alapuolella esimerkki rakenteesta.

```
SELECT column_name(s)
FROM table_name
```

Esimerkki SELECT-lauseen käytöstä. Funktio hakee spacegroups-tilusta kaikki rivit ja asettaa ne taulukkomuuttujaan sisälto. Talletuksessa käytetään avuksi tietokantafunktiota `mysqli_fetch_array`, jolla voidaan päättää minkälaiseen indexiin tieto talletetaan. Vaihtoehtoina ovat ASSOC, NUM ja BOTH. ASSOC tallettaa tiedon kannasta haetun sarakkeen nimen mukaan. NUM tallettaa numeron mukaan ja BOTH molempien.

```
function spacegroupsDB($yhteys) {
    $sql_lause = "SELECT * FROM spacegroups ORDER BY space-
groups_order";
    $tulokset = mysqli_query($yhteys, $sql_lause);
    if (mysqli_num_rows($tulokset) < 1) {
        return null;
    }
}
```



```

    } else {
        $sisalto = array();
        while($rivi=mysqli_fetch_array($tulokset, MYSQLI_ASSOC)) {
            array_push($sisalto,
                $rivi);
        }
        mysqli_free_result($tulokset);
        return $sisalto;
    }
}

```

### 3.2.2 INSERT INTO- lause

SQL-lauseella pystytään lisäämään kantaan tietoa. Rakenne on seuraavanlainen:

```

INSERT INTO table_name (column1, column2, column3)
VALUES (value1, value2, value3)

```

Alapuolella on esimerkki siitä, miten **SELECT** lausetta käytetään kartoitussovelluksessa. **SELECT**-lause hakee viimeisimmän lisätyn kyselypohjaryhmän järjestysnumeron kannasta ja tallettaa sen muuttujaan. Tämän jälkeen arvoa kasvatetaan yhdellä ja uusi kyselypohjaryhmä lisätään kantaan oikealla järjestysnumerolla.

```

function lisaaKyselypohjaRyhmaKantaan($yhteys,$kyselypohjaryhmanimi)
{
    $kyselypohjaryhmanimi=mysqli_real_escape_string($yhteys,$kyselypohjaryhmanimi);

    $sql_lause = "SELECT MAX(spacegroups_order) AS spacegroups_order FROM spacegroups";
    $tulokset = mysqli_query($yhteys, $sql_lause);
    $rivi = mysqli_fetch_array($tulokset, MYSQLI_ASSOC);
    mysqli_free_result($tulokset);
    $order = $rivi["spacegroups_order"] + 1;
}

```

```

$sql_lause = 'INSERT INTO spacegroups
(space_name,spacegroups_order) VALUES ("' .
$kyselypohjaryhmanimi . "', ' . $order . ' )';
$tulokset = mysqli_query($yhteys, $sql_lause);
}

```

### 3.2.3 UPDATE- lause

UPDATE-lausetta käytetään päivittämään olemassaolevia kenttiä tietokannasta. UPDATE-lause hyödyntää WHERE ehtoa osoittamaan halutut kentät joihin UPDATE-lause kohdistuu. Rakenne on seuraavaa:

```

UPDATE table_name
SET column1=value, column2=value2
WHERE some_column=some_value

```

Alapuolella kartoitustyökalun esimerkki UPDATE-lauseen käytöstä. Funktio päivittää oikean kentän nimen id:n avulla:

```

function updatespacegroup($yhteys,$spacegroupid, $nimi) {
    $spacegroupid =
    mysqli_real_escape_string($yhteys,$spacegroupid);
    $nimi = mysqli_real_escape_string($yhteys,$nimi);
    $sql_lause = 'UPDATE spacegroups SET space_name = "' .
    $nimi . '" WHERE id=' . $spacegroupid;
    $tulokset = mysqli_query($yhteys, $sql_lause);
    mysqli_free_result($tulokset);
}

```

### 3.2.4 DELETE- lause

DELETE-lausetta käytetään poistamaan tietokannan kenttiä. Myös DELETE lauseen yhteydessä käytetään WHERE ehtoa osoittamaan, mitkä kentät poistetaan. Esimerkki käytöstä:

```

DELETE FROM table_name
WHERE some_column = some_value

```

## 4 KÄYTTÖLIITTYMÄ

Käyttöliittymän kuvaus ja toteutus kuvattu asiakkaan raportissa, mikä pidetään salaisena.

## 5 TIETOKANTA

Tietokannan kuvaus ja toteutus kuvattu asiakkaan raportissa, mikä pidetään salaisena.

## 6 YHTEENVETO

Sovelluksen toteuttaminen sujui hyvin, vaikka omat kokemukseni jQuery Mobilesta olivat lähtökohtaisesti vähäiset. jQuery Mobilesta oli saatavilla paljon tietoa ja jQuery Mobilen omista manuaaleista oli helppo katsoa apua.

Sovellusta suunniteltaessa päädyttiin erilaisiin kannan malleihin, joista mielestäni viimeisimmästä versiosta saatiin optimoitu ja selkeä. Ongelmallisinta sovelluksen suunnittelussa oli tietokannan luominen. Itse en ollut koskaan tehnyt näin laajaa tietokantaa. Tietokannan eheyden kannalta ratkaisevaksi osoittautuikin huolellisesti tehty käsitelmä. Ensimmäinen oma versio kannasta osoittautui muuten hyväksi, mutta sisälsi liikaa tauluja, eikä se ollut eheä. Ongelma saatiin kuitenkin korjattua.

Tällä hetkellä sovellus täyttää kaikki toiminnallisuudet, joita siltä aluksi lähdettiin hakemaan. Sovellus toimii eri mobiilialustoilla ja sillä saa tehtyä sulavasti esteettömyyskartoituksia. Sovellus sisältää pääkäyttäjälle ja peruskäyttäjälle erilaiset toiminnallisuudet.

Tulevaisuudessa sovellus on tarkoitus ottaa käyttöön kenttätyöskentelyyn eli sillä on tarkoitus tehdä mobiililaitteilla esteettömyyskartoituksia paikasta riippumatta. Sovellusta on jo testattu ja se on toiminut hyvin. Mielestäni hyvä jatkokehitysidea sovellukselle voisi olla 'offline'-versio eli sovellus toimisi myös ilman Internet-yhteyttä. Tällöin sovelluksesta tarvitsisi tehdä erilainen versio ellei jopa useampi, mutta sovelluksessa voitaisiin hyödyntää puhelimen omia ominaisuuksia, kuten esimerkiksi kameraa.

## LÄHTEET

1. Satakunnan ammattikorkeakoulun www-sivut. Viitattu 11.4.2014.  
[http://www.samk.fi/tutkimusryhm%C3%A4t/esteett%C3%B6myys\\_ja\\_saavuttavuus](http://www.samk.fi/tutkimusryhm%C3%A4t/esteett%C3%B6myys_ja_saavuttavuus)
2. The jQuery Foundation. 2014. Viitattu 24.3.2014.  
<http://api.jquerymobile.com/>
3. Reid, J. 2011. jQuery Mobile. Sebastopol, California: O'Reilly Media. Viitattu 24.3.2014. <http://www.it-ebooks.info/book/344/>
4. The PHP Group. 2014. Viitattu 24.3.2014.  
<http://www.php.net/manual/en/preface.php>
5. Alshanetsky, I. 2005. SQL injection. Teoksessa PHP Security. Viitattu 24.3.2014. <http://dev.mysql.com/tech-resources/articles/guide-to-php-security-ch3.pdf>
6. The jQuery Foundation. 2014. jQuery mobile documents. Viitattu 24.3.2014.  
<http://demos.jquerymobile.com/1.0a1/experiments/api-viewer/docs/click/index.html><http://demos.jquerymobile.com/1.0a1/experiments/api-viewer/docs/blur/index.html>
7. The jQuery Foundation. 2014. jQuery mobile documents. Viitattu 24.3.2014.  
<http://demos.jquerymobile.com/1.0a2/experiments/api-viewer/docs/val/>
8. The jQuery Foundation. 2014. jQuery mobile documents. Viitattu 24.3.2014.  
<http://demos.jquerymobile.com/1.0a1/experiments/api-viewer/docs/focus/>
9. The jQuery Foundation. 2014. jQuery mobile documents. Viitattu 24.3.2014.  
<http://demos.jquerymobile.com/1.0a1/experiments/api-viewer/docs/replaceWith/index.html>