



Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

VILLE MANSIKKA-AHO

Web-pohjainen huoltosovellus

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN
KOULUTUSOHJELMA
2020

Tekijä(t) Mansikka-aho, Ville	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Kesäkuu 2020
	Sivumäärä 24	Julkaisun kieli Suomi
Julkaisun nimi Web-pohjainen huoltosovellus		
Tutkinto-ohjelma Sähkö- ja automaatiotekniikan koulutusohjelma		
<p>Opinnäytetyön toimeksiantajana toimi Ferob Oy, joka tekee mm. robottien huoltoja ja näihin huoltoihin liittyy huoltoraportti. Huoltoraportti on PDF-dokumentti, josta selviää mitä huoltotoimenpiteitä huoltokohteeseen on tehty. Opinnäytetyön tavoitteena oli tehdä huoltoraporttisoluvus Ferob Oy:lle. Toimeksiantajalla oli tarve ohjelmalle, jonka avulla huoltoraportti voidaan tehdä ja tallentaa nopeasti kentällä. Lisäksi tarve oli mm. hakutoiminnolle, jolla voitiin rajata näytettäviä raportteja. Lopputuloksena sovelluksesta tuli melko perinteinen verkkopohjainen tietokantaa hyödyntävä ohjelma. Sovellus sisältää melkein jokaisesta verkkosovelluksesta löytyvät osat, kuten kirjautumis- ja pääsivun. Ohjelman jatkokehitys pitäisi olla vaivatonta, sillä tietokannan ja sovelluksen rakenne on suunniteltu helposti jatkettavaksi.</p>		
Ohjelmointi, automaatiotekniikka		

Author(s) Mansikka-aho, Ville	Type of Publication Bachelor's thesis	Date June 2020
	Number of pages 24	Language of publication: Finnish
Title of publication Web-based maintenance application		
Degree program in Electrical and Automation Engineering: Bachelor of Engineering		
Abstract The thesis was commissioned by Ferob Oy, which does e.g. robotic maintenance and these maintenance is accompanied by a maintenance report. The maintenance report is a PDF document that shows e.g. what maintenance measures have been taken at the service site. The aim of the thesis was to make a maintenance report application for Ferob Oy. The client had a need for a program that would allow a maintenance report to be made and saved quickly in the field. In addition, the need was e.g. a search function that could narrow down the reports displayed. As a result, the application became a fairly traditional web-based database program. The application includes components found in almost every web application, such as the login and main page. Further development of the program should be effortless, as the structure of the database and the application is designed to be easy to continue.		
Keywords: Programming, automation technology		

SISÄLLYS

1 JOHDANTO	6
1.1 Työn tausta ja tavoite	6
1.2 Työn rajaus.....	6
1.3 Työn toimeksiantaja	7
1.4 Huoltoraportti.....	7
2 KÄYTETYT TEKNIIKAT	8
2.1 Frontend ja backend	8
2.1.1 HTML, CSS ja Bootstrap.....	8
2.1.2 JavaScript, jQuery ja Ajax	9
2.1.3 Olio-ohjelmointi.....	10
2.1.4 PHP	10
2.1.5 MySQL.....	11
2.2 Käytetyt ohjelmat	11
2.2.1 Umlet, MySQL Workbench ja Visual Studio Code.....	11
2.3 Käyttöympäristö.....	12
3 KÄYTTÖLIITTYMÄ	12
3.1 Käyttöliittymän suunnittelu.....	12
3.1.1 Hyvä käyttöliittymä.....	12
3.1.2 Rakenne ja ominaisuudet	12
3.1.3 Käyttöliittymän osat	13
3.1.4 Huoltoraportti.....	13
3.1.5 Kirjautuminen ja Uloskirjautuminen	13
3.1.6 Pääikkuna	14
3.1.7 Uusi huoltoraportti	14
3.1.8 Haku-toiminto ja Hae raportteja toiminto.....	15
4 TOTEUTUS	15
4.1 Hakemistorakenne.....	16
4.2 Kirjautumis-ikkunan toteutus.....	16
4.3 Pääsivu-ikkunan toteutus	17
4.4 Uusi huoltoraportti	18
4.5 Hae raportteja ja Hakutoiminnon toteutus	19
4.6 Näytä, muuta ja poista.....	19
5 TIETOKANTA	20
5.1 Hyvän tietokannan ominaisuuksia	20
5.1.1 Tietokannan suunnittelu.....	21

5.2 SQL-injektio ja sen estäminen	21
5.3 Viiteavaimet	22
6 PDF	22
6.1 PDF-tiedoston luominen	22
6.2 FPDF luokan periyttäminen	23
7 TESTAUS	23
8 YHTEENVETO, LOPPUTULOS, JATKOKEHITYS	24
LÄHTEET	

SANASTO JA KÄSITTEET

luokka	olio-ohjelmoinnin peruskäsite, määrittelee olion rakenteen
olio	luokasta tehty ilmentymä
viiteavain	tietokannan viittaus toiseen tauluun
perintä	olio-ohjelmointi tapa, jolla laajennetaan olemassa olevaa luokkaa
olio-ohjelmointi	ohjelmointitapa tai -metodi, jolla jäsennetään ja jaetaan koodi luokiksi
URL	Uniform Resource Locator, verkkosivuihin viitataan URL:n avulla
id	on yksilöllinen tunniste
relaatiotietokanta	relaatiomalliin perustuva tietokanta, sisältää tauluja
metodi	luokan toiminto

1 JOHDANTO

Työn toimeksiantajalla oli tarve saada sovellus helpottamaan huoltoraportin tekemistä kentällä eli huoltokohteessa. Opinnäytetyön tuloksena toteutettiin ko. sovellus. Tämä antoi mahdollisuuden oppia uutta, kuten verkkosovelluksen tekemiseen tarvittavia taitoja.

1.1 Työn tausta ja tavoite

Työn toimeksiantaja Ferob Oy tekee mm. asentamiensa robottisolujen huoltoja. Huoltoja on kahta eri päätyyppiä: vuosi- ja korjaushuolto. Näistä huolloista tehdään huoltoraportti, joka hyödyntää ja palvelee toimittajaa sekä loppuasiakasta. Ongelmana ennen sovellusta oli huoltoraportin tekeminen kentällä. Aiemmin huollon suorittaja kirjoitti huoltoraportin tiedot vanhaan pohjaan ja vasta toimistolla varsinaiseen raporttiin liitettiin huollon tiedot kuten huoltotoimenpiteet. Tämä nosti virheiden esiintymistodennäköisyyttä sekä esiintymistiheyttä. Lisäksi oli tarve saada valmis raportti tehtyä nopeasti jo huoltokohteessa, jotta toimistolla raportti saataisiin helposti tarkastettua ja lähetettyä asiakkaalle.

Työn tavoitteena oli tehdä toimiva verkkopohjainen sovellus, jonka avulla toimeksiantaja saa tehtyä huoltoraportin ja tallennuttua sen tiedot tietokantaan. Huoltoraportti on yleensä yhden sivun mittainen pdf-tiedosto, joka sisältää huollon tiedot. Sovelluksen tuli vähentää raportin tekemiseen kuluvaa aikaa.

1.2 Työn rajaus

Työn rajaus on tärkeä osa työmäärän hallitsemisessa ja rakenteen selkeyttämisessä. Työn rajaus aloitettiin alkukeskustelulla toimeksiantajan kanssa. Alkukeskustelussa rajattiin pääpiirteittäin, mitä sovelluksen tulee tehdä ja mitkä ovat sen minimivaatimukset. Tämän jälkeen alkukeskustelun nojalla rajattiin käytettävät tekniikat. Alussa

pidettiin mahdollisena tehdä paikallinen työpöytäsovellus, mutta nopeasti päädyttiin verkkopohjaiseen ratkaisuun. Paikallisessa sovelluksessa ongelmana on se, että sovellus pitää asentaa jokaisen käyttäjän tietokoneeseen erikseen. Verkkosovellus ei puolestaan vaadi käyttäjältä ylimääräisiä ohjelmien asennuksia. Verkkosovelluksen hyvänä puolena on se, että se mahdollistaa tietojen tallentamisen tietokantaan. Tietojen tallennus tietokantaan mahdollistaa erilaiset monipuoliset hakutoiminnot. Olemassa olevien tietojen, kuten asiakas- ja robottisolurekisterin, hyödyntäminen poistaa jo olemassa olevan tiedon uudelleensyöttötarvetta. Lisärajaukset syntyivät käytännössä testaamalla, kuten raportin ulkoasuun sekä sovelluksen käyttöliittymään liittyvät rajaukset.

1.3 Työn toimeksiantaja

Työn toimeksiantajana toimi Ferob Oy. Ferob Oy on Ulvilassa sijaitseva automaatioalan yritys. Ferob Oy:n perusti vuonna 2011 Niko Moilanen ja Juha Karttunen. Pääasiallisesti Ferob Oy toimii teollisuusrobottien parissa ja tarjoaa mm. robottien huolto- ja korjauspalveluita.

1.4 Huoltoraportti

Huoltoraportin tehtävänä on hyödyntää ja palvella sekä toimittajaa että loppuasiakasta. Huoltoraportilla on muun muassa seuraavia tavoitteita:

- 1) Toimenpiteiden dokumentointi ja arkistointi. Tietokanta mahdollistaa monipuoliset hakutoiminnot.
- 2) Helppo, nopea ja vaivaton huoltoraportin generointi, joka säästää aikaa ja vaivaa. Olemassa olevien tietojen, kuten asiakas- ja robottisolurekisterin, hyödyntäminen poistaa jo olemassa olevan tiedon uudelleensyöttötarvetta. Helppouden takia raportti ei myöskään jää tekemättä tai ole puutteellinen. Huoltoraportti toimii pohjatietona myyntilaskun muodostamiseen.
- 3) Huoltoraporttien avulla voidaan seurata määrätyn robottisolun huoltohistoriaa ja suunnitella ennakoivia huoltotoimenpiteitä
(Niko Moilanen sähköposti 30.3.2020).

Huoltoraportin rakenne ja sisältö tuli hyvin pitkälti annettuna toimeksiantajalta, sillä heillä oli luonnollisesti paras käsitys siitä, mitä huoltoraporttiin tulisi sisällyttää.

2 KÄYTETYT TEKNIIKAT

2.1 Frontend ja backend

Frontend on yleisnimitys selainpuolen teknologioista, kuten HTML ja JavaScript. Selainpuolen tekniikalla tarkoitetaan selaimessa eli käyttäjän tietokoneessa, tabletissa tai kännykässä ajettavaa ohjelmaa. Selaimen tehtävänä on mm. lähettää tai vastaanottaa kuvia ja muita WWW-sivuilta löytyviä tietoja. Backend on yleisnimitys palvelinpuolen teknologioista, kuten PHP.

2.1.1 HTML, CSS ja Bootstrap

HTML on lyhenne sanoista Hypertext Markup Language. HTML standardoitu kuvauskieli, jolla internetsivut ovat tehty. HTML:llä voidaan kuvata mm. tekstin rakenne (HTML. n.d.). Tyypillinen HTML-sivu on rakenteeltaan seuraavanlainen:

```
<html>
<head></head>
<body></body>
</html>
```

HTML on rakenteinen kuvauskieli, jossa käytetään sisäkkäisiä tageja sivun rakenteen merkkäämiseen. HTML-sivu koostuu head-osuudesta, jossa on mm. metatietoa sivusta, tyylitiedoston linkitys ja selainskriptin linkitys. Body-osuus on varsinaisesti käyttäjälle näkyvä osuus. Paljon käytettyjä HTML-tageja ovat mm. table, div ja p. Cascading Style Sheets (CSS) on kuvauskieli, jonka avulla verkkosivun ulkoasua voidaan muuttaa. CSS:llä voidaan vaihtaa esim. värejä ja fontteja (HTML & CSS. n.d.). CSS:ää voidaan käyttää kolmella eri tavalla. Voidaan käyttää ulkoista CSS-tiedostoa (erillinen CSS-tiedosto), elementin sisäistä tyyliä tai style-elementin sisään laitettua CSS-koodia. Ulkoisen CSS-tiedoston voi linkittää HTML sivun head-osuuteen seuraavasti. Tässä ulkoisen CSS-tiedoston nimi on mystyle.css.

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```


Elementtiin voidaan suoraan kirjoittaa CSS tyylimäärittelyjä seuraavasti. Tässä h1 ot-sikon fontin väriksi laitetaan sininen ja teksti tasataan keskelle.

```
<h1 style="color:blue;text-align:center;">This is a heading</h1>
```

Voidaan myös käyttää HTML-tagia nimeltään style seuraavasti. Tässä asetetaan sivun taustaväriksi vihreä.

```
<style>
body {
  background-color: green;
}
</style>
```

Esim. Muutetaan koko HTML-dokumentin taustaväriksi sinisen.

```
body {
  background-color: blue;
}
```

Bootstrap on ilmainen frontend kirjasto, joka nopeuttaa ja helpottaa web-sivujen tekemistä. Bootstrap sisältää HTML ja CSS malleja mm. painikkeille ja lomakkeille. Bootstrap mahdollistaa responsiivisen verkkosivun tekemisen. Responsiivisuus tarkoittaa sitä, että sivu näyttää samalta eri päätelaitteilla (Bootstrap. n.d).

2.1.2 JavaScript, jQuery ja Ajax

JavaScript on selainskripti. Selainskripti on ohjelmakoodia, joka on tarkoitettu selaimen ajettavaksi. Selainskriptit ovat yksinkertaisia ohjelmointikieliä, jotka tarjoavat paljon valmiita funktioita. Näiden funktioiden avulla voidaan mm. muuttaa selaimen toimintaa (Korpela, J., Linjama, T. 2004, 718). Esim. Muutetaan p elementin, jonka id on p1 fontin väriksi sininen:

```
<p id="p1">Hello World!</p>
<script>
document.getElementById("p1").style.color = "blue";</script>
</script>
```

jQuery on nopea ja monipuolinen JavaScript-kirjasto. jQuery:llä voidaan tehdä mm. animaatioita ja tapahtumien käsittelyä (jQuery. n.d.). Esim. Muutetaan p elementin, jonka id on p1 fontin väriksi punainen:

```
<p id="p1">Hello World!</p>
<script>
$("#p1").css("color", "red");
</script>
```

Ajax on lyhenne sanoista Asynchronous JavaScript and XML. Ajax ei ole varsinainen ohjelmointikieli, vaan sillä viitataan joukkoon web-kehitys tekniikoita. Ajaxia käytetään paljon frontend-kehityksessä, kuten JavaScript-koodeissa. Ajaxin avulla voidaan esim. lähettää tai vastaanottaa dataa palvelimelta tai tietokannasta häiritsemättä käyttäjää (Ajax programming. n.d.). Esim. Näytetään testi.php sisältö elementissä, jonka id on sivu. Tässä käytetään jQuery:n load-metodia.

```
$("#sivu").load("testi.php");
```

2.1.3 Olio-ohjelmointi

Olio-ohjelmointi on ongelman jakamista pieniksi kokonaisuuksiksi. Näitä kokonaisuuksia kutsutaan luokiksi. Luokasta tehtyä ilmentymää kutsutaan olioksi. Luokkaa voidaan kuvailla olion pohjapiirustukseksi. Olio-ohjelmointi helpottaa koodin uudelleenkäyttöä ja selkeyttää ohjelman rakennetta.

2.1.4 PHP

PHP on web-kehityksessä yleisesti käytetty skriptikieli. Yleensä PHP:tä käytetään palvelinpuolella (PHP. n.d.). PHP-koodia voidaan lisätä HTML:n sekaan seuraavasti

```
<?php
echo "Hello world!";
?>
```

Tämä tulostaa näytölle Hello world! tekstin. PHP:hen voi lisätä ulkoisia tiedostoja kahdella komennolla require ja include. Näiden komentojen ainoa ero on se, että virheen sattua require komento aiheuttaa virheen(error) ja pysäyttää skriptin. Include puolestaan aiheuttaa ainoastaan varoituksen(warning) ja skriptin ajo jatkuu normaalisti. Esim. tiedosto.ini:n lisääminen. Tässä tiedosto.ini sijaitsee samassa kansiossa, missä itse ajettava PHP-tiedostokin. Tämä rivi lisää PHP-tiedostoon kaikki tiedosto.ini:ssä olevat rivit.

```
<?php
include tiedosto.ini
?>
```

MySQL-tietokantaan PHP:llä saa yhteyden mm. käyttämällä mysqli-ajuria seuraavasti:

```
<?php
$mysqli = mysqli_connect('palvelin','käyttäjä','salasana','tietokanta');
?>
```

Tässä `mysqli_connect` nimiselle funktiolle annetaan parametreina mm. palvelimen osoite. HTTP POST-metodilla lähetetty data voidaan vastaanottaa PHP:llä seuraavasti:

```
if(isset($_POST['nimi']))
{
    echo "Nimi vastaanotettu";
}
```

Tässä tarkistetaan, että saadaan POST-metodilla lähetetty nimi kenttä. Jos nimi kenttä saadaan, tulostetaan näytölle teksti ”Nimi vastaanotettu”.

2.1.5 MySQL

MySQL on maailman suosituin avoimen lähdekoodin tietokanta. MySQL on tehokas, luotettava ja helppokäyttöinen tietokanta. MySQL:ää käyttää mm. Twitter ja Facebook (About MySQL. n.d.). Esim. MySQL-komento, joka hakee kaikki rivit ja kaikki sarakkeet taulusta nimeltä taulu on seuraavanlainen:

```
SELECT * FROM taulu
```

Datarivin lisääminen onnistuu seuraavalla rakenteella:

```
INSERT INTO taulun_nimi(sarake1,sarake2,...) VALUES (arvo1, arvo2,...)
```

Taulun tietojen päivittämiseen voidaan käyttää seuraavaa komentoa:

```
UPDATE taulun_nimi
SET sarake1=arvo1, sarake2=arvo2,...
WHERE jokin_sarake=jokin_arvo
```

Datan poistaminen voidaan tehdä seuraavasti:

```
DELETE FROM taulun_nimi WHERE ehto;
```

2.2 Käytetyt ohjelmat

2.2.1 Umlet, MySQL Workbench ja Visual Studio Code

Umlet on ilmainen avoimen lähdekoodin UML-työkalu, jolla on helppo tehdä esim. ER-kaavio, jossa on taulujen väliset relaatiot esitetty. Ohjelman hyvänä puolena on sen yksinkertaisuus ja helppokäyttöisyys. MySQL Workbench on työkalu MySQL-tietokannan käyttöön. Visual Studio Code on paljon käytetty kevyt ja muokattava tekstieditori.

2.3 Käyttöympäristö

Toimeksiantajalla on käytössä webbihotelli, joka tukee suoraan PHP:tä ja MySQL:ää. Tämän takia webbihotellin palveluita ei tarvinnut muuttaa. Vaihtoehtona olisi ollut käyttää omaa palvelinta, mutta tämä olisi vaatinut huomattavasti enemmän osaamista palvelimien arkkitehtuurista ja ylläpidosta. Webbihotelli tarjoaa hyviä palveluita, kuten asiakaspalvelun.

3 KÄYTTÖLIITTYMÄ

3.1 Käyttöliittymän suunnittelu

3.1.1 Hyvä käyttöliittymä

Käyttöliittymän tarkoituksena on toimia rajapintana käyttäjän ja sovelluksen välillä. Käyttöliittymän tulee olla mm. yksinkertainen, selkeä ja helposti omaksuttava. Lisäksi on tärkeää ottaa huomioon käyttäjät suunnitellessa käyttöliittymää. Huoltoraporttisolvelluksen käyttöliittymän muokkaaminen tapahtui suoraan käyttäjäpalautteen perusteella. Käyttäjäpalautteen avulla tein muutokset käyttöliittymään. Lähtökohtaisesti minulla oli melko vapaat kädet tehdä mieleiseni käyttöliittymä.

3.1.2 Rakenne ja ominaisuudet

Aloitin käyttöliittymän suunnittelun pohtimalla tarvittavia näyttöjä. Suunnittelin ensin pääpiirteittäin mitkä näytöt ohjelmaan tarvitaan, kuten pää- ja kirjautumisikkunat. Tämän jälkeen jaoin pääikkunan rakenteen osiin. Näiden pohjalta syntyivät näyttökuvat, joiden perusteella saatoinkin tehdä varsinaisen käyttöliittymän toteutuksen. Aivan tarkkaan kaikkia yksityiskohtia ei suunnitteluosuudessa huomannut ottaa huomioon, joten suunnitteluosuuteen piti palata myöhemmin. Tämä johtui osaksi myös siitä, että sovellukseen tarvittavat ominaisuudet muuttuivat hieman ajan kuluessa.

3.1.3 Käyttöliittymän osat

Huoltoraporttisovelluksen käyttöliittymä sisältää perinteiset verkkosovelluksen osat kuten kirjautuminen, uloskirjautuminen ja pääsivu. Huoltoraporttisovelluksen toimintoina ovat Uusi ikkuna, Statistiikka ja Hae raportteja toiminnot. Näihin osiin päädyin melko nopeasti ottamalla mallia valmiista sovelluksista ja konsultoimalla toimeksiantajaa. Lähes jokainen verkkosovellus koostuu edellä mainituista osista, joten käyttöliittymän rakenne on yleisesti käytetty ja hyväksi todettu. Huoltoraportin rakenne on kuvan 1 mukainen. Tarkistuskohtien tarkempi rakenne on yrityssalaisuus.



Kuva 1: Huoltoraportin rakenne

3.1.5 Kirjautuminen ja Uloskirjautuminen

Kirjautumis-ikkuna koostuu kahdesta syöttökentästä ja painikkeesta Kirjautu. Käyttäjä syöttää käyttäjätunnuksen ja salasanan ja kirjautuu sovellukseen painamalla Kirjautu-painiketta (Kuva 2).



Kuva 2. Kirjautumis-ikkuna suunnitelma

3.1.6 Pääikkuna

Pääikkunassa (Kuva 3) näkyy statistiikka-sivu kirjautumisen jälkeen. Statistiikka-sivulle pääse uudelleen statistiikka-painiketta painamalla. Statistiikka-sivulla näytetään tietoja kuten asiakkaiden lukumäärä. Painamalla Uusi huolto raportti, aukeaa lomake huolto raportin tekemiseen. Hae raportteja kohdasta, aukeaa raporttilistaus. Kirjaudu ulos painikkeesta kirjaututaan ohjelmasta ulos ja palataan kirjautumisikkunaan.

Huolto raporttisovellus		Kirjaudu ulos
Huolto raportit	statistiikkaa	
Uusi huolto raportti	asiakkaiden lkm	
Hae raportteja	Robottisolujen lkm	
statistiikkaa	Robottien määrä	
	Kirjausten määrä	
	Viimeisimmät 5 raporttia	
	Tulevat ennakkohuollot	

Kuva 3. Pääikkuna suunnitelma

3.1.7 Uusi huolto raportti

Uusi huolto raportti -sivu on lomake, jonka rakenne näkyy Kuvassa 4, jonka tietojen avulla voidaan huolto raportti tehdä.

Uusi huolto raportti
Valitse raportin tyyppi
Valitse asiakas
Valitse solu
Tallenna raportti

Kuva 4. Uusi huolto raportti

3.1.8 Haku-toiminto ja Hae raportteja toiminto

Hae raportteja toiminto on kuvan 5 mukainen. Hae raportteja näyttää huoltoraportit taulukossa. Toimintoina ovat näytä, muuta ja poista. Näytä toiminto avaa ko. raportin pdf-muotoisen tiedoston. Muuta toiminto avaa muutostoimintoikkunan, jonka avulla käyttäjä voi muokata ko. raporttia. Poista toiminto poistaa halutun raportin.

Haku toiminto on osa Hae raportteja -sivua. Kuten kuvasta 5 näkyy, niin käyttäjä voi suodattaa hakulistausta päivämäärän, asiakkaan tai solun perusteella. Käyttäjä valitsee suodattimen, jonka jälkeen Suodata-painiketta painamalla näytetään suodatettu listaus. Poista suodattimet toiminto poistaa hakusuodattimet käytöstä eli sivu listaa kaikki raportit.

Huoltoraportit

↓

RaportinID	SuoritusPVM	Asiakas	Solu	<input type="button" value="Näytä"/>	<input type="button" value="Muuta"/>	<input type="button" value="Poista"/>
------------	-------------	---------	------	--------------------------------------	--------------------------------------	---------------------------------------

Kuva 5. Haku toiminto suunnitelma

4 TOTEUTUS

Käyttöliittymän tekemisessä on käytetty valmista Bootstrap-mallia nimeltään SB Admin. Kyseessä on ilmainen, vapaan lähdekoodin omaava MIT-lisenssillä oleva Bootstrap malli. Valitsin kyseisen pohjan, koska siihen ei tarvinnut tehdä paljon muutoksia ja se vähensi käyttöliittymän tekemiseen käytettyä aikaa. Tarkastukset kuten, käyttäjän syötteen tarkastus on tehty JavaScriptillä.

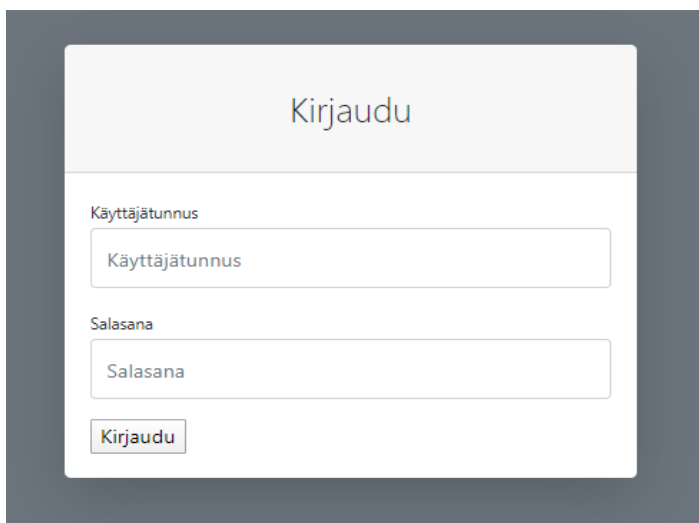
4.1 Hakemistorakenne

Sovelluksen hakemistorakenne on mahdollisimman yksinkertainen ja selkeä. Ideana on laittaa esim. kuvat omaan hakemistoonsa. Lisäksi skriptit ja tyylisivut on laitettu omiin kansioihinsa. Nimeämisessä on yritetty käyttää selkeitä nimiä, kuten tietokanta-luokka voisi olla nimeltään tietokanta.php. Kuvassa 6 näkyy mahdollinen hakemistorakenne.



Kuva 6. Hakemistorakenne

4.2 Kirjautumis-ikkunan toteutus



Kuva 7. Kirjautumisikkuna

Kirjautumis-ikkuna on kuvan 7 näköinen HTML-sivu, jossa on käytetty tyylisivuja (CSS). Kirjautumis-ikkuna koostuu sisäkkäisistä DIV-elementeistä sekä HTML-lomakkeesta, jonka käyttäjä täyttää. JavaScriptillä tarkistetaan, että käyttäjä syöttää molemmat syöttökentät. Jos käyttäjä ei anna molempia kenttiä, ilmoitetaan käyttäjälle ponnahdusikkunalla, että käyttäjän tulee syöttää molemmat kentät. Kirjaudu painike kutsuu ohjelmaa, joka tarkistaa, että käyttäjällä on oikeudet käyttää sovellusta. Jos

riittävät oikeudet löytyvät, niin ohjataan käyttäjä pääsivulle. Jos ei riittäviä oikeuksia löydy, niin ilmoitetaan käyttäjälle oikeuksien riittämättömyydestä.

4.3 Pääsivu-ikkunan toteutus

Huoltoraporttisovellus

Statistiikka

Asiakkaiden lukumäärä: 5
 Robottisolujen määrä: 13
 Robottien määrä: 3
 Kirjausten määrä: huoltoraportteja 6 / vikaraportteja 2 / yhteensä 8

Viimeisimmät 5 raporttia

RaportinID	Asiakas	Solu
050520-09:33:50	UPM	Solu2
050520-12:50:22	UPM	Solu1
050520-12:50:53	UPM	Solu1
300420-13:14:31	UPM	Solu1
300420-13:12:14	UPM	Solu1

Tulevat ennakkohuollot: 4 kpl

pvm	Asiakas	Solu
30.10.2020	UPM	Solu1
30.10.2020	UPM	Solu1
30.10.2020	UPM	Solu1
01.05.2021	UPM	Solu1

Kuva 8. Pääsivu

Pääsivu on Kuvan 8 näköinen sivu, joka sisältää toiminnot: Uusi huoltoraportti, Hae raportteja, Käyttöohje ja Kirjaudu Ulos. Pääsivulla tarkistetaan, että käyttäjä on kirjautunut sisään. Jos käyttäjä ei ole kirjautunut sisään, ohjataan käyttäjä kirjautumissivulle takaisin. Pääsivun vasemmassa laidassa on toiminnot Uusi huoltoraportti ja Hae raportteja. Oikeassa yläkulmasta aukeaa lista, josta voi valita ohjekirjan tai uloskirjautumisen. Uloskirjautuminen painike kutsuu ohjelmaa, joka suorittaa uloskirjautumisen. Uusi huoltoraportti -painike avaa lomakkeen, jolla voidaan tehdä huoltoraportti. Aukaistaessa pääsivu näytetään statistiikka sivu.

Statistiikka sivulle pääsee painamalla Statistiikka-painiketta. Statistiikka sivu sisältää tietoja kuten asiakkaiden lukumäärä, solujen lukumäärä ja robottien lukumäärä. Tämä on toteutettu hakemalla tietokannasta tarvittavat tiedot. Esim. Tietokannan taulun rivien lukumäärä voidaan hakea seuraavalla MySQL-lauseella.

```
SELECT COUNT(Table) as TableIkm FROM Table;
```

Tässä haetaan Table-nimisen taulun rivien lukumäärä ja talletetaan arvo Tablelkm nimiseen aliakseen. Alias komento tehdään MySQL:n as-määritteellä. Tämä tarkoittaa sitä, että Tablelkm sisältää rivien lukumäärän.

4.4 Uusi huoltoraportti

The screenshot shows a web application interface for creating a new maintenance report. The page title is 'Uusi huoltoraportti'. On the left, there is a navigation menu with 'Pääsivu', 'Huoltoraportit', 'Uusi huoltoraportti', and 'Hae raportteja'. The main content area contains the following form elements:

- Valitse raportin tyyppi:** A dropdown menu with 'Vikahuoltoraportti' selected.
- Valitse asiakas:** A dropdown menu with 'UPM' selected.
- Valitse solu:** A dropdown menu with 'Solu1' selected.
- Valitse raporttiin tulevat laitteet:** A table with columns 'Laitetyyppi', 'Malli', 'Sarjanumero', and 'Mukaan raporttiin'.

Laitetyyppi	Malli	Sarjanumero	Mukaan raporttiin
Robotti	KUKA KR 16 R1610	656565	<input type="checkbox"/>
Robotti	KR 20 R1610	54534534534	<input type="checkbox"/>
Ohjais	Ohjaislaite	12345321	<input type="checkbox"/>
Tarttuja	Tarttujalaitte	443322	<input type="checkbox"/>
- Lisätietoja:** A large text area for additional information.
- Käytetyt Osat:** A text area for used parts.
- Testiajo suoritettu:** A list of checkboxes for 'Käsijolla', 'Automaattijolla', and 'Tuotantokäytössä'.
- Päivämäärä:** An input field with '07.05.2020' entered.
- Työn suorittaja(t):** An input field with 'admin' entered.
- Työaika:** An input field for working hours.
- Tallenna raportti:** A button to save the report.

Kuva 9. Uusi huoltoraportti

Uusi huoltoraportti on kuvan 9 näköinen. Valitsemalla asiakas soluvalintalista täyttyy tietokannan arvojen perusteella. Tämä tarkoittaa sitä, että tietokannasta haetaan kyseiseen asiakkaaseen liittyvät solut ja täytetään solu-valintalista. Valitsemalla solu aukeaa laitelistaus, minkä avulla käyttäjä voi päättää mitä laitteita raporttiin otetaan mukaan. JavaScriptillä on tarkastettu, että käyttäjä antaa pakolliset kentät. Pakollisia kenttiä ovat asiakas ja solu. Lisäksi on tarkastettu, että työaika on positiivinen kokonaisluku. JavaScriptillä voidaan tarkistaa, onko annettu arvo numeerinen isNaN-funktion avulla. Tallenna raportti -painike kutsuu JavaScript-funktiota, joka välittää lomakkeen tiedot Ajax-kutsun avulla varsinaisen tallennuksen tekeväälle PHP-ohjelmalle.

4.5 Hae raportteja ja Hakutoiminnon toteutus

Hae raportteja toiminto näyttää HTML-taulukkona kaikki huoltoraportit. Tämä on toteutettu hakemalla tietokannasta huoltoraportin tiedot ja näyttämällä tiedot käyttäjälle. PHP:llä voidaan käydä tietokannan taulun tietoja läpi silmukassa mm. seuraavalla rakenteella:

```
$mysqli = new mysqli("localhost","my_user","my_password","my_db");
$sql = "SELECT * from testitaulu";
if ($result = $mysqli -> query($sql)) {
    while ($row = $result -> fetch_row()) {
        echo $row[0];
    }
}
```

Tässä tulostetaan testitaulun ensimmäisen sarakkeen tiedot.

Hakutoiminto toimii seuraavalla periaatteella: Ensin muodostetaan oikean muotoinen MySQL-komento valittujen suodattimien perusteella. Tämän jälkeen suoritetaan ko. haku tietokannasta ja tehdään HTML taulukko, joka sisältää haun tulokset. Tämä taulu näytetään pääsivulla hakutoiminnon alapuolella kuten kuvassa 10. Kuvassa on käytetty testidataa ja suodatettu listaa asiakas- ja solusuodattimilla.

Huoltoraportit

Hakusuodattimet

alkupvm pp.kk.vvvv | loppupvm pp.kk.vvvv

Valitse asiakas

UPM

Valitse solu

Solu1

Suodata Poista suodattimet

Raportin ID	Suoritus pvm	Asiakas	Solu	Tyyppi	Työaika[h]	Tekijä(t)	
270420-09:57:57	02.05.2020	UPM	Solu1	Vuosihuoltoraportti	12345	Vima, Tima	Näytä Muuta Poista
270420-09:19:23	02.05.2020	UPM	Solu1	Vuosihuoltoraportti		admin	Näytä Muuta Poista
270420-08:14:15	30.04.2020	UPM	Solu1	Vikahuoltoraportti	123456	Ville	Näytä Muuta Poista
270420-09:18:42	27.04.2020	UPM	Solu1	Vuosihuoltoraportti		admin	Näytä Muuta Poista
270420-08:49:44	27.04.2020	UPM	Solu1	Vikahuoltoraportti	1	Seku, Johu	Näytä Muuta Poista

Kuva 10. Haku toiminto

4.6 Näytä, muuta ja poista

Näytä toiminto avaa rivillä olevan huoltoraportin. Muuta-toiminto avaa muutoslomakkeen (kuva 11), jonka täyttämällä käyttäjä voi muuttaa raportin tietoja. Päivitä-painikkeella muutokset talletetaan sekä tietokantaan että pdf muotoiseen dokumenttiin. Älä päivitä ja palaa takaisin -painike ei tee mitään muutoksia ja ohjelma palaa

raporttilistaus-sivulle. Poista toiminto poistaa ko. rivillä olevan raportin. Uloskirjautuminen voidaan tehdä poistamalla käytössä oleva istunto ja ohjaamalla sovellus takaisin kirjautumissivulle.

Muuta raportin 270420-10:36:34 tietoja ja paina Päivitä painiketta

Lisätietoja

Käytetytosat

- Testiajo suoritettu käsiajolla:
- Testiajo suoritettu automaattilla:
- Testiajo suoritettu tuotantokäytössä:

Työnsuorittaja1

Pvm

Työaika

Kuva 11. Muuta raporttia

5 TIETOKANTA

5.1 Hyvän tietokannan ominaisuuksia

Hyvän tietokannan keskeisiä ominaisuuksia ovat

- kattavuus
- selkeys ja ymmärrettävyys
- muutosjoustavuus

- yleiskäyttöisyys
- eheys
- ohjelmointimukavuus
- suorituskyky

(Hovi, A., Huotari, J. & Lahdenmäki T. 2005, 21)

Tietysti kaikki ehdot harvoin toteutuvat samanaikaisesti, mutta yritin pitää mielessä yllä mainitut ominaisuudet suunnitellessani tietokantaa.

5.1.1 Tietokannan suunnittelu

Tietokannan rakenteen suunnittelu lähti käsiteanalyysistä eli miettimällä, mitä tietoa ja minkä tyyppistä tietoa halutaan tallentaa tietokantaan. Tästä mietinnästä seurasi tietokannan taulujen karkea rakenne. Tämän jälkeen pohdin taulujen välisiä relaatioita. Tämä osio oli aika pitkälti yritys/erehdys tekniikalla tehty. Rakenteellisesti ideana on se, että asiakkaaseen liittyy soluja. Näihin soluihin liittyy tietty määrä laitteita. Tämä on melko luonnollinen rakenne, sillä se vastaa käytäntöä. Robottisolussa on tietty määrä laitteita kuten robotteja ja robottisolu on jossakin asiakkaan tiloissa. Näiden perusteella suunnittelin tarvittavat viiteavaimet. Viiteavain laitetaan ns. lapsitauluun. Lapsitaulussa on siis viiteavain isäntätauluun.

5.2 SQL-injektio ja sen estäminen

SQL-injektiohyökkäys on tietokantaan kohdistuva hyökkäys, joka mahdollistaa haitallisten SQL-lauseiden suorittamisen. SQL-injektiohyökkäykset ovat yksi vanhimmista, yleisimmistä ja vaarallisimmista verkkosovellusten haavoittuvuuksista (SQL.injektio.n.d.). SQL-injektio voidaan yrittää estää mm. käyttämällä mysqli-ajuria seuraavasti:

```
$lause = $mysqli -> prepare("INSERT INTO taulu(nimi, osoite) VALUES (?, ?)");
$stmt -> bind_param("ss", $nimi, $osoite);
$nimi = "ville";
$email = "osoite";
$stmt -> execute();
```

SQL-injektio on ongelma niissä tapauksissa, jossa käyttäjä voi antaa syötteitä vapaasti. Esim. HTM-lomake, jossa on input tyyppisiä tekstisyötekenttiä.

5.3 Viiteavaimet

Luotaessa taulua, voidaan viiteavain antaa create-komennossa seuraavanlaisesti

```
CREATE TABLE TAULU (
    id int NOT NULL PRIMARY KEY,
    nimi VARCHAR(100),
    dataid int,
    FOREIGN KEY (dataid) REFERENCES Taulu2(id)
);
```

Tässä dataid on viiteavain, joka viittaa Taulu2:sen id kenttään.

6 PDF

6.1 PDF-tiedoston luominen

FPDF on PHP luokka, jonka avulla voidaan tehdä PDF-tiedostoja pelkästään käyttämällä puhdasta PHP-koodia. FPDF on ilmainen ja sitä voidaan käyttää vapaasti. FPDF:llä voidaan mm. tehdä automaattinen rivin vaihto, linkkejä ja ylä-/alatunniste. (FPDF. n.d.). FPDF:ää voidaan käyttää seuraavasti PHP:lla.

```
<?php
require('fpdf.php');
$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(40,10,'Hello World!');
$pdf->Output();
?>
```

Require tai include komennolla voidaan tuoda FPDF-kirjasto PHP-ohjelman käyttöön. Yllä oleva koodi tekee pdf-tiedoston, jossa lukee teksti ”Hello World!”. Lisäksi tekstin fontiksi on asetettu Arial, fontin kooksi 16 ja teksti on korostettu.

6.2 FPDF luokan periyttäminen

Huoltoraporttisovelluksessa käytetään FPDF-luokasta periytettyä luokkaa, jonka avulla huoltoraportti tehdään. Se hoitaa pdf:n tekemiseen liittyvät toiminnallisuudet. Tämä sen takia, että FPDF-luokka tarjoaa valmiiksi haluttuja toimintoja, kuten output-metodi. Output-metodin avulla voidaan tallentaa pdf levyille. PHP:llä voidaan perii FPDF luokka seuraavasti:

```
class OmaLuokka extends FPDF
```

Tämä tarkoittaa, että OmaLuokka perii FPDF:n attribuutit, muodostimet ja metodit.

Muodostin voidaan tehdä seuraavasti:

```
public $teksti;
public function __construct($iteksti)
{
    parent::__construct();
    $this->teksti = $iteksti;
}
}
```

Tässä on ensin kutsuttu yläluokan (FPDF) muodostinta. Tämän jälkeen on tallennettu luokan jäsenmuuttujaan parametrina saatu tieto.

Esim. Ylätunnisteen tekeminen

```
function Header()
{
    $this->Image($this->logo,10,10,30);
    $this->SetFont($this->font,'B',$this->fontsize);
    $this->Cell(80);
    $this->Cell(30,10,$this->otsikkoteksti,0,0,'C');
    $this->Cell(50);
    $this->SetFont($this->font,'',10);
    $this->Cell(30,10,"<" . $this->nimi . ">",0,0,'r');
    $this->Ln(20);
}
}
```

Tässä on ylätunnisteeseen laitettu logo vasempaan reunaan. Keskelle on laitettu otsikko ja oikealle nimi. Komennolla Cell voidaan kirjoittaa pdf:ään. Tässä on annettu parametrina tekstin sijainti (C=keskellä, r = oikealla). Samalla rakenteella on tehty muutkin tarvittavat metodit, kuten Footer(alatunniste).

7 TESTAUS

Testausta suoritettiin jo ohjelmaa tehdessä. Joka kerta kun tein muutoksia ohjelmaan suoritin nopean käyttöttestauksen. Tämä piti sisällään sovelluksen toimintojen oikeellisuuden testauksen. Lopputestaus suoritettiin toimeksiantajan toimesta. Ohjelma annettiin huollon tekijöille käyttöön, jotka testasivat ohjelmaa käytännössä.

Ohjelmaa pyrittiin testaamaan mahdollisimman kattavasti ja systemaattisesti, jotta siinä olevat ongelmat löytyisivät mahdollisimman nopeasti.

8 YHTEENVETO, LOPPUTULOS, JATKOKEHITYS

Jatkossa sovellusta voidaan täydentää mm. lisäämällä asiakkaalle, solulle ja laitteille omat lisäys-, muutos- ja poistotoiminnot. Nämä lisäykset ovat yksinkertaisia tehdä sillä tietokanta on jo valmiina.

Lopputuloksena syntyi toimiva ja käytössä oleva sovellus johon toimeksiantaja oli tyytyväinen. Ohjelmasta saatu palaute oli positiivista ja sovelluksen teosta jäi hyvä mieli. Projekti opetti monta asiaa tietokannasta ja webbiohjelmoinnista ja yleisesti ottaen projektinhallinnasta. Tietokannoista tuli opittua viiteavaimien käytöstä. Webbiohjelmoinnista eniten opein PHP-ohjelmoinnista, joka ei sinällään juurikaan eroa muista ohjelmointikielistä, joista on kokemuksia. Perusteet ovat melko lailla samat. Tärkein oppimani asia oli projektin aikataulutuksen tärkeys. Vaikka kuinka hyvin yrittäisi aikatauluttaa ja suunnitella projektia, aina tulee vastaan yllättäviä ja aikaa vieviä ongelmia.

LÄHTEET

HTML. n.d. Viitattu 14.4.2020. <https://fi.wikipedia.org/wiki/HTML>

HTML & CSS. n.d. Viitattu 14.4.2020. <https://www.w3.org/standards/web-design/htmlcss#whatcss>

Bootstrap. n.d. Viitattu 21.5.2020.
https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp

Korpela, J & Linjama, T 2004. XHTML-käsikirja. Jyväskylä: Docendo.

jQuery. n.d. Viitattu 21.5.2020. <https://jquery.com/>

Ajax programming. n.d. Viitattu 21.5.2020. <https://www.keycdn.com/support/ajax-programming>

PHP. n.d. Viitattu 21.5.2020. <https://en.wikipedia.org/wiki/PHP>

About MySQL. n.d. Viitattu 21.5.2020. <https://www.mysql.com/about/>

Hovi, A., Huotari, J. & Lahdenmäki T. 2005. Tietokantojen suunnittelu ja indeksointi. Porvoo: Docendo.

sql-injection. n.d. Viitattu 21.5.2020. <https://www.acunetix.com/websitesecurity/sql-injection/>

FPDF. n.d. Viitattu 27.4.2020. <http://www.fpdf.org/>

Moilanen, Niko. Opparin rajauksesta. Vastaanottaja:illevma85@gmail.com. Vastaanotettu 30.3.2020.