

Maatalouslomittajille tarkoitettu sähköinen työnhallintatyökalu käyttäen NFC-tekniikkaa

Mikko Pyysalo

Opinnäytetyö
Toukokuu 2013
Tietotekniikan
koulutusohjelma
Ohjelmistotekniikka

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

MIKKO PYYSALO:

Maatalouslomittajille tarkoitettu sähköinen työhallinta työkalu käyttäen NFC-tekniikkaa.

Opinnäytetyö 38 sivua, joista liitteitä 12 sivua
Toukokuu 2013

Opinnäytetyön tarkoituksena oli toteuttaa maatalouslomittajille sähköinen työhallinta työkalu. Työssä suunniteltiin ja toteutettiin mobiilisovellus sekä palvelimella sijaitseva ohjelma ja ohjelmalle tietokanta.

Työ oli kaksiosainen. Toinen oppilas toteutti työssä vaadittavan käyttöliittymän Nokia N9 puhelimelle ja minun osuuteni oli toteuttaa työlle sille vaadittavan palvelinpään ohjelman ja tietokannan.

Työn idea oli luoda maatalousmittajille sähköinen työhallinta työkalu, jolla voitaisiin demonstroida maatalouslomittajille ja maatilayrittäjille nykyaikaista työhallinta menetelmää. Opinnäytetyössä käytettiin PHP-, MySQL- ja XML-ohjelmointikieliä.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
ICT Engineering
Software Engineering

MIKKO PYYSSALO:

Electric work control tool for stand-ins using NFC-technology

Bachelor's thesis 38 pages, appendices 12 pages
May 2013

The purpose of this thesis was to create electric work control tool for stand-ins using NFC-technology. In this work we have planned and implemented mobile application, program what is located in server and database for program.

This thesis was separated in two-pieces. Another student implemented work required interface for Nokia N9 phone and my half from thesis was to create server and database for program.

The idea of the work is to create electric work control tool which can be demonstrate stand-ins and farmers what is modern work control method. In this thesis program have been used PHP-, MySQL- and XML-programming languages.

Key words: php, mysql, xml

SISÄLLYS

1	JOHDANTO.....	6
2	LOMITTAJA.....	7
3	OHJELMISTOT	8
	3.1 Mobiiliapplikaatio.....	8
	3.2 PHP	9
	3.3 MySQL	9
4	SUUNNITTELU	10
	4.1 Tietokannan suunnittelu.....	10
	4.2 PHP-ohjelman suunnittelu	11
5	PHP-OHJELMA.....	12
	5.1 Ohjelman erittely	12
	5.2 Ohjelmaan sisäänkirjautuminen.....	12
	5.3 Ohjelmasta uloskirjautuminen	15
6	TIETOKANTA	17
	6.1 Tietokannan luonti	17
	6.2 Tietokannan taulukot	17
	6.3 Tietokannan laajentaminen	21
7	TESTAUS	22
8	TIETOTURVA.....	23
9	YHTEENVETO	24
	LÄHTEET.....	25
	LIITTEET	26
	Liite 1. PHP-pääohjelman lähdekoodi.....	26
	Liite 2. Tietokannan lähdekoodi.....	30
	Liite 3. Valmiin mobiilisovelluksen tilannekuvat.	35

LYHENTEET JA TERMIT

PHP	Hypertext Preprocessor, PHP on komentosarjakieli.
MySQL	Relaatiotietokantaohjelmisto.
XML	Extensible Markup Language, merkintäkieli tai standardi.
NFC	Near Field Communication.
QT	Kehitysympäristö.

1 JOHDANTO

Tarkoituksena oli luoda testiohjelma mobiililaitteelle jolla voitiin demonstroida maatalouslomittajille sähköisen työnhallinta työkalun tuomat edut verrattuna heidän nykyiseen paperiseen työn hallintamenetelmään verrattuna. Työ toimi pelkästään esittely ohjelmana maataloudessa työskenteleville henkilöille. Työn tarkoitus ei ollut suoraan luoda heille uutta menetelmää, mutta työn pohjalta voidaan toteuttaa samaa tekniikkaa käyttävän ohjelman joka voi korvata heidän nykyisen paperisen menetelmän.

Tässä työssä kerrotaan kuinka koko ohjelma luotiin ja suunniteltiin aivan alusta sekä kerrotaan tarkasti miten työn vaatima serveri ja tietokanta on toteutettu ja miten ne toimivat. Mitä erilaisia ohjelmointi kieliä tarvittiin testi ohjelman toteutukseen jotta se toimi halutulla tavalla.

Työ toteutettiin monessa eri vaiheessa. Ensimmäinen vaihe oli työn vastaanotto ja samalla suunnittelun aloitus. Seuraavaksi alkoi toteutus suunnittelun pohjalta ja lopuksi oli ohjelman testaus ja esitleminen.

Jotta tämän opinnäytetyön lukija voisi ymmärtää, miksi ohjelma tehtiin on tässä työssä esitelty maatalouslomittajan perustyöpäivää. Mikä on lomittaja ja mitä kaikkea hän tekee sekä mitä kaikkea tapahtuu ennen kun lomittaja vastaanottaa työnsä.

2 LOMITTAJA

”Maatalouslomittaja on kunnan palkkaama työntekijä, joka hoitaa maatilaa sillä välin kun maatalousyrittäjä on sairaus- tai vuosilomalla. Lain mukaan lomittajan tehtävänä on huolehtia maatilalla karjalouteen kuuluvista ja muista maatalousyrittäjien hoitoon liittyvistä välttämättömistä tehtävistä. Maatalouslomittaja on usein moni osaaja. Päivittäiseen työhön kuuluu jatkuva eläinten havainnointi ja tarkkailu, ja lomittajan tulee osata tunnistaa eläimen terveydentilan muutokset ja kiimakierrot. Myös erilaisten maatalouskoneiden kuten traktorin käyttö erilaisine lisälaitteineen on arkipäivää maatalouslomittajien työssä. –wikipedia [3]”

Maatalousyrittäjä luo lomittajan työtehtävälistan. Ensiksi hän luo tehtävälistan, jonka jälkeen hän lähettää sen kunnalle ja kunta palkkaa lomittajan. Jos maatalousyrittäjän lomittajan tarve on kiireellinen, on kunnalla myös päivystyksessä lomittajia jotka voivat tulla paikalle tietyn määräajan sisällä.

Ohjelman tarkoitus on nopeuttaa ja helpottaa maatalousyrittäjän ja kunnan välistä paperilla käytävää keskustelua. Lopullisessa ohjelman versiossa olisi jonkinlainen sähköinen työkalu, millä maatalousyrittäjä ja kunta voivat jakaa tietojansa ja tarpeitansa. Esimerkiksi sopimuksia tai dokumentteja.

Mobiilisovelluksen tarkoitus olisi tuoda helpotuksia maatalouslomittajan peruspäivään. Sovelluksella ei ole tarkoitus suoraan korvata lomittajaa, vaan tarjota sähköisiä ongelmien ratkaisuja työkaluja. Esimerkiksi ohjelmassa on työtehtävän sijainti. Jos lomittaja on ensimmäistä kertaa maatilalla, niin hän ei välttämättä tiedä missä kaikki on.

3 OHJELMISTOT

Seuraavaksi käydään läpi, mitä ohjelmia työn tekemiseen käytettiin. Ohjelmat olivat helppo valita koska työ toteutettiin Nokia N9:lle, sen NFC valmiuden takia. Koska puhelimen malli oli Nokia, niin päätimme käyttää mobiiliohjelman luomiseen sille tarkoitettua kehitystyökalua Nokia Qt:a.

Ohjelmointikielten valinta oli helppoa, koska PHP oli entuudestaan jo tuttu ohjelmointikieli. Seuraavaksi valitsimme palvelinohjelmiston, koska tarvitsimme työssä jatkuvasti päällä olevaa palvelinta. Päätimme, että palvelimen ja domainin vuokraaminen olisi helpoin vaihtoehto. Valitsimme Hostingpalvelut Oy palvelimemme tarjoajaksi. Vuokrasimme palvelimen Hostingpalveluilta tietokantaohjelmaksi tuli MySQL.

3.1 Mobiiliapplikaatio

Mobiiliohjelman luominen alkoi ensin lataamalla QT:n asennuspaketti ohjelman kotisivuilta. Ohjelman kotisivuilla valittiin ensimmäiseksi ohjelmalle käyttöjärjestelmä, minkä päällä ohjelma tulee toimimaan. Tämän jälkeen lataajan piti täyttää pienimuotoinen kaavake. Kaavakkeen täytön jälkeen sivusto lähettää latauslinkin ja lisenssiavaimen kaavakkeen täyttäjän valitsemaan sähköpostiosoitteeseen.

Tämän jälkeen ohjelman asentaja voi asentaa QT:n. Ohjelman asennus tapahtui todella suoraviivaisesti vain seuraamalla asennusohjeita. Asennuksen aikana ei tarvinnut määritellä mitään erikoisasetuksia.

Asennuksen jälkeen ohjelma oli käyttövalmis ja mobiilipuolen ohjelman rakentaminen voitiin aloittaa.

Työparini toteutti mobiiliapplikaation, joten en kerro tässä työssä miten itse mobiilisovellus on tarkalleen toteutettu. Mutta kuvissa 1-8 (liite 3) on esitelty miltä mobiiliapplikaatio näytti valmiina.

3.2 PHP

Ohjelman palvelimen päässä toimivan ohjelman toteuttamiseen valitsimme PHP-ohjelmointikielen, joka on suoritusvaiheessa tulkittava ohjelmointikieli. Koska vuokrasimme palvelimen, käytimme palvelun tarjoajien valmiiksi asentamia ohjelmia.

PHP- tiedostojen luonti onnistui todella helposti notepad++:lla. Itse tiedostot vain ladattiin palvelun tarjoajan käyttämälle tiedostojenhallinta järjestelmälle, josta palvelin osasi kääntää PHP- tiedostot asiakkaan haluamalle sivulle.

PHP on avointa lähdekoodia käyttävä ohjelmointikieli, joten sen käyttö on ilmaista ja mitään lisenssimaksuja ei tarvitse maksaa. PHP mahdollistaa ohjelman XML-tekniikkaa käyttävien tiedostojen luomisen sekä poistamisen.

PHP- ohjelmointikielen valintaan vaikutti todella paljon siihen sisäänrakennetut helppokäyttöiset tietokantayhteydet, joilla tietojen hakeminen tietokannasta tapahtuu todella yksinkertaisesti. Kielen valintaan myös vaikutti todella paljon ohjelmointikielen helppous ja aikaisempi kokemus PHP- ohjelmoinnista.

3.3 MySQL

Ohjelmassa tarvittavien tietojen tallentamiseen käytettiin MySQL- tietokantaa, joka on myös avoimeen lähdekoodiin perustuva tietokantaohjelmisto. MySQL on relaatiotietokantaohjelmisto, jota kehittää MySQL AB.

Valitsimme MySQL tietokantaohjelmaksi sen yhteensopivuuden PHP-kielen kanssa sekä siksi, että MySQL on entuudestaan tuttu ja olimme jo aikaisemmin luoneet monia erilaisia tietokantoja.

Tietojen lisääminen tietokantaan sekä tietokannan hallinta tapahtui Hostinpalveluiden tarjoaman phpMyAdmin työkalun kautta.

4 SUUNNITTELU

Niin kuin kaikki hyvät ja onnistuneet työt alkavat kunnollisella suunnittelulla, niin myös tämän ohjelman toteutus alkoi. Ennen suunnittelun alkua oli tietysti työn vastaan ottaminen sekä myös suullinen perehdyttäminen maatalouslomittajien perinteiseen työpäivään. Perehdyttämisessä käytiin läpi kaikki eri vaiheet, mitä lomittaja tekee perinteisen työpäivän aikana sekä myös mitä kaikkea tapahtuu ennen lomittajan saapumista työpaikalleen.

Perehdyttämisen tarkoituksena oli luoda mielikuva ohjelman toteuttajille, jotta osaisimme suunnitella oikeanlaisen ohjelman. Joka auttaisi ja nopeuttaisi lomittajan tavallista työpäivää.

Kun perehdyttäminen oli ohi ja ymmärsimme mitä kaikkea lomittajan perustyöpäivänä tapahtuu ja mitä kaikkea hän tarvitsisi työpäivän aikana. Vasta tämän jälkeen aloitimme suunnittelemaan ohjelmaa, jotta ohjelma voisi tarjota mahdollisimman paljon helpotuksia lomittajan työpäivään.

4.1 Tietokannan suunnittelu

Koska työparini suunnitteli mobiilisovelluksen, en voinut heti aloittaa suunnittelemaan tietokantaa, koska en tiennyt minkälaisen tietokannan työparini tarvitsee. Vasta kun mobiilipuoli oli suunniteltu ja osittain toteutettu tiesin sitten vasta mitä kaikkea tietoa pitää tietokannassa olla.

Tietokannan suunnittelussa toimi myös pohjana lomittajien tämän hetkinen paperinen kaavake. Jonka pohjalta pystyin luomaan tietokannan rakenteen joka piti kaiken lomittajien tarvitsemat tiedot.

Koska kaksi ihmistä teki työtä samaan aikaan piti tietokantaa muokata koko ajan erilaisten vaatimusten pohjalta.

4.2 PHP-ohjelman suunnittelu

Suunnittelun pääkohtana toimi mobiilisovellus ja mitä tietoja ohjelma haluaisi tietokannasta. Vasta kun mobiilisovellus oli ohjelma kutsuja myöten valmis ja tietokanta oli tarpeiden mukainen. Vasta tämän jälkeen aloitimme PHP-ohjelman suunnittelemisen.

Suunnitelmassa päädyimme, että PHP-ohjelma tarkistaisi ensin kutsun tullessa onko laitteella oikeutta tietokantaan. Jos laitteella ei ollut oikeutta kantaan, niin ohjelma ei tekisi mitään. Mutta jos laitteella on oikeus tietokantaa niin PHP-ohjelma hakisi tarvittavat tiedot tietokannasta ja loisi 2 erillistä XML-tekniikkaa käyttävää tiedostoa.

Tiedostojen luonnin jälkeen mobiililaite lataisi tiedostot puhelimen omaan muistiin yhteyden katkeamisen varalta. Kun mobiililaite on ladannut tiedostot, niin PHP-ohjelma poisti luodut tiedostot.

5 PHP-OHJELMA

Seuraavassa osassa esitellään koko PHP-ohjelman toiminta kertomalla niistä lähdekoodin (liite 1) kanssa. Ohjelma on jaettu kahteen eri osaan. Ensimmäinen osa on, kun lomittaja sisään kirjautuu puhelimella palvelimelle NFC-lukijan kautta. Toinen osa on, kun lomittaja kirjaa itsensä NFC-lukijan kautta ulos.

5.1 Ohjelman erittely

Ohjelma on jaettu neljään eri tiedostoon. Ensimmäinen osa, joka toimii palvelimen domainin indeksinä, on vain muutama komento. Ensimmäinen ohjelman rivi on kutsun vastaan ottaminen, joka tulee mobiiliohjelman kautta.

Ohjelman toinen osa sisältää kaikki tietokanta kutsut, sekä kaikki funktiokutsut.

Ohjelman kolmas osa sisältää tietokannan yhdistämiseen tarvittavat tiedot, jotka ovat serverinimi, käyttäjätunnus ja salasana.

Ohjelman neljäs osa sisältää kaikki funktiot, joita tarvitaan XML-tiedostojen luomiseen sekä poistamiseen.

5.2 Ohjelmaan sisäänkirjautuminen

Ohjelma toimii seuraavalla tavalla. Kun lomittaja tulee maatilalle ja laittaa puhelimensa NFC-lukijan päälle, niin mobiililaitte kutsuu ohjelmaa ja se myös samalla lähettää laitteen tunnistetiedon. Kun palvelinpään ohjelma vastaan ottaa kutsun se tarkistaa ensimmäiseksi, onko tietokannassa tietoa kyseiselle puhelimelle.

```

4
5  if(isset($_POST['name'])){
6      $comfirmnid = 0;
7  }
8  if($_POST['name'] != ""){
9      $tmp = mysql_query("SELECT contract_idcontract, contract_phone FROM contract_has_standin",$connect);
10
11     while( $row = mysql_fetch_assoc($tmp)){
12         $comfirmnumber = $row["contract_phone"];
13         if( $comfirmnumber == $_POST['name']){
14             $conid = $row["contract_idcontract"];
15             $comfirmnid = 1;
16         }
17     }
18 }

```

Jos puhelin ei lähetä omaa tunnistetta, tunniste puuttuu tai kyseiselle puhelimelle ei ole tallennettu mitään tietokantaan, niin ohjelma ei luo mitään uusia tiedostoja.

Jos ohjelma löytää kyseisen puhelimen tiedot tietokannasta, niin ohjelma alkaa suorittaa lisää kutsuja tietokantaan, jotta kaikki tarpeelliset tiedot saadaan tietokannasta ohjelmaan.

Ohjelma tallentaa kaikki tietokannasta tulleet tiedot erilaisiin väliaikaisiin muuttujiin, joista ne sitten jatko puretaan käyttäen erilaisia silmukoita.

```

24
25 //finding all information nessesary information from contract
26 $tmp2 = mysql_query("SELECT standin_form_iddetails, farm_id, instructions_location, manual_location from contract where idcontract = ". $conid ." ",$connect);
27
28 //taking information to attributes for xml
29 while($row2 = mysql_fetch_assoc($tmp2)){
30     $standingid = $row2["standin_form_iddetails"];
31     $farmid = $row2["farm_id"];
32     $instruction = $row2["instructions_location"];
33     $manual = $row2["manual_location"];
34 }
35
36 //finding all nessesary information from farm and permanent_employees
37
38 $tmp3 = mysql_query("Select farm.name as fname, permanent_employees.name as pname ,
39 permanent_employees.phone_number as pphone from farm inner join permanent_employees where ".$farmid." = permanent_employees.farm_id and ".$farmid." = farm.id ");
40 while($row3 = mysql_fetch_assoc($tmp3)){
41     $farmname = $row3["fname"];
42     $pname = $row3["pname"];
43     $pphone = $row3["pphone"];
44 }

```

Kun kaikki tarvittavat tiedot on saatu ja ne on jaettu omiin osiinsa, niin ohjelma suorittaa funktiokutsun, jolla luodaan ja nimetään ensimmäinen XML-tiedosto. Tämän jälkeen luodaan tiedostolle XML-dokumentin vaatiman otsikon ja ensimmäisen tiedon kuvausrivin.

```

17
18 function openfile(){
19     $fh2 = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml","w");
20     $xml2 = "<?xml version='1.0'?>";
21     $xml2 .= "<missions>";
22     fwrite($fh2,$xml2);
23     fclose($fh2);
24 }

```

Seuraavaksi ohjelma suorittaa kutsun tietokantaan, jolla se hakee XML-tiedostoa varten lisää tarvittavaa tietoa. Kun tiedot on haettu ohjelma suorittaa uuden funktiokutsun, jolla kutsutaan XML-tiedoston rakentajaa. Rakentaja funktio rakentaa XML-tiedostoa tietokanta kutsun purkusilmukan sisällä.

```

44 }
45 openfile();
46 $i = 1;
47 //finding all necessary information from machinery
48 $tmp4 = mysql_query("Select * from machinery where standin_form_iddetails = ".$standingid." ");
49 while($row4 = mysql_fetch_assoc($tmp4){
50     $taskid = $row4["taskid"];
51     $taskname = $row4["taskname"];
52     xmltmp2($taskid,$taskname);
53     $tmp5 = mysql_query("select name,instructions,video from machines where taskid =". $taskid ." ");
54     while($row4 = mysql_fetch_assoc($tmp5) {
55         $macname = $row4["name"];
56         $macinstructions = $row4["instructions"];
57         $macvideo = $row4["video"];
58         xmltmp3($macname, $macinstructions, $macvideo, $i);
59         $i++;
60     }
61     xmltmp4();
62     $i = 1;
63 }

```

XML-tiedoston rakentajaa on kutsuttu purkusilmukan sisällä sen takia, jotta XML-tiedoston tietojen luominen oikeaan järjestykseen olisi mahdollista.

Seuraavaksi ohjelma kutsuu ensimmäisen XML-tiedoston sulku funktiota, jolla suljetaan tiedoston luonti ja lisätään tiedoston loppuun sulkeva kuvausrivi. Tämän jälkeen ohjelma vielä kutsuu toista XML-tiedoston luonti funktiota, jolla luodaan toinen uusi XML-tiedosto.

```

3 function xmltmpl($a, $b, $c, $d, $e){
4   $fh = fopen("/home/lompsane/public_html/xmlfiles/authbart farm.xml","w");
5   $xml = "<?xml version='1.0'?>";
6   $xml .= "<auth>";
7   $xml .= "<access>1</access>";
8   $xml .= "<farmname>".$a."</farmname>";
9   $xml .= "<bossname>".$b."</bossname>";
10  $xml .= "<bossnumber>".$c."</bossnumber>";
11  $xml .= "<instructions>".$d."</instructions>";
12  $xml .= "<manual>".$e."</manual>";
13  $xml .= "</auth>";
14  fwrite($fh,$xml);
15  fclose($fh);
16 }
17
18 function closefile(){
19  $fh5 = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml","a+");
20  $xml6 = "</missions>";
21  fwrite($fh5,$xml6);
22  fclose($fh5);
23 }

```

Lopuksi ohjelma katkaisee yhteyden tietokantaan ja asettaa tarkistus lipun nolllaksi seuraavaa ohjelman kutsua varten.

5.3 Ohjelmasta uloskirjautuminen

Lopuksi kun lomittaja on hoitanut työnsä ja kirjautuu ulos. Hän asettaa puhelimen NFC-lukijan päälle ja puhelin lähettää palvelinpään ohjelmalle uloskirjautumiskutsun.

Kun ohjelma saa uloskirjautumiskutsun, ohjelma kutsuu tiedostojen puhdistus funktiota. Puhdistus funktio avaa ensin ensimmäisen XML-tiedoston ja tyhjentää tiedoston. Tämän jälkeen funktio avaa toisen XML-tiedoston ja tyhjentää myös sen.

```
65
66 function filecleaner() {
67     $clean = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml", "w");
68     $filler = "";
69     fwrite($clean, $filler);
70     fclose($clean);
71
72     $clean = fopen("home/lompsane/public_html/xmlfiles/authbart farm.xml", "w");
73     fwrite($clean, $filler);
74     fclose($clean);
75 }
76
77 function xmltmp5(){
78     $file = fopen("home/lompsane/public_html/xmlfiles/logoutbart farm.xml", "w");
79     $xml = "<?xml version='1.0'?>";
80     $xml = "<logout>";
81     $xml = "<access>l</access>";
82     $xml = "</logout>";
83     fwrite($file, $xml);
84     fclose($file);
85 }
```

Tämän jälkeen ohjelma kutsuu vielä yhtä XML-tiedoston teko funktiota, joka luo uuden XML-tiedoston. Tiedostoon kirjoitetaan uloskirjautumislipun sen hetkinen tilanne.

6 TIETOKANTA

Seuraavaksi esitellään miten tietokanta on luotu ja mitä kaikkea se pitää sisällään.

6.1 Tietokannan luonti

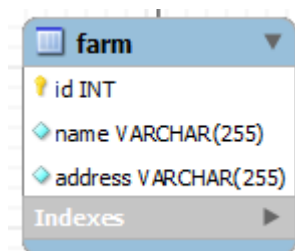
Koska tiesimme, että tietokannasta tulee todella iso ja se tulee pitämään sisällään paljon erilaista tietoa. Niin päätin, että tietokannan luominen olisi helpompaa jollain siihen tarkoitettuun ohjelmalla. Joten valitsimme tietokannan luontiohjelmaksi MySQL Workbench-ohjelman, jolla suunnittelimme ja loimme koko tietokannan. Kun tietokanta oli valmis, se oli helppo siirtää palvelimelle.

Koska työ muuttui mobiilipuolen ohjelman puolella muutaman kerran, jouduimme luomaan monia erilaisia versioita tietokannasta. Kun viimeinen tietokanta oli luotu, se oli versio numeroltaan 6.

6.2 Tietokannan taulukot

Tietokanta pitää sisällään 7 erilaista taulukkoa, jotka kaikki ovat jollain tavalla riippuvaisia toisistaan.

Kaikkein tärkein taulukko on itse maatilataulukko. Taulukko pitää sisällään maatilalan id tunnusteen, maatilalan nimen ja osoitteen. Taulukko on sen takia tärkein, koska maatilalan tunnusteen kautta pystytään yhdistämään jokaiselle maatilalle sille ominaiset tiedot.

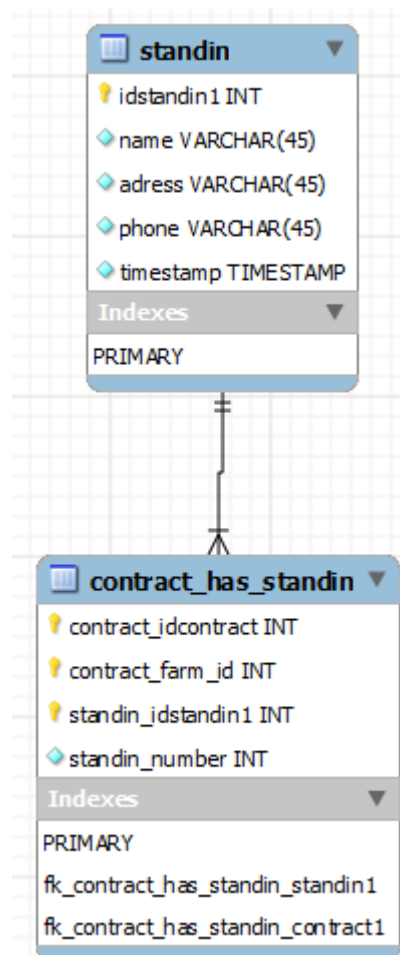


Tietokannan toiseksi tärkein taulukko on itse sopimustaulukko, joka pitää sisällään kaikki tiedot mitä heidän nykyinen paperinen järjestelmä pitää sisällään. Taulukosta selviää kaikki mitä kunnan pitää tietää, jotta he osaavat palkata oikeanlaisen lomittajan.

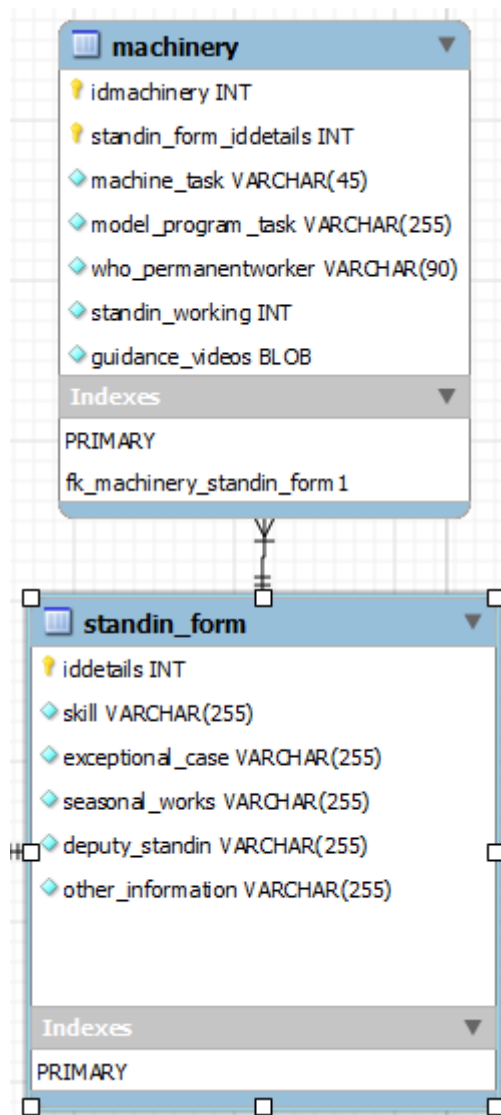
contract	
idcontract	INT
farm_id	INT
standin_form_iddetails	INT
task_percent1	INT
task_percent2	INT
attachment1	INT
attachment2	INT
begindate1	DATE
begindate2	DATE
enddate1	DATE
enddate2	DATE
hours	INT
works_with	INT
pworker	INT
pworker_name	VARCHAR(45)
deputy	INT
deputy_name	VARCHAR(45)
deputy_phone	VARCHAR(45)
TSS_time	TIME
TSS_created	DATE
same_vacationtime	TIME
svacation_created	DATE
work_measure	TIME
wmeasure_created	DATE
written_instructions	INT
instructions_location	VARCHAR(90)
quality_manual	INT
manual_location	VARCHAR(90)
phone_list	INT
list_location	VARCHAR(90)
social_facilities	INT
circumstance_date	DATE
circumstance_fixes	VARCHAR(255)

Indexes

Lomittajan tiedot löytyvät omasta taulukosta, joka on sidottu aputaulukkoon. Aputaulukko pitää sisällään kolme erilaista avainta. Aputaulukko pitää sisällään sopimus tunnisteiden, maatilan tunnisteiden ja lomittajan tunnisteiden. Tämän aputaulukon avulla tietokanta pystyy sitomaan oikean lomittajan oikeaan sopimukseen.



Lomittajan lomaketaulukko, joka on sidottu työkonetaulukkoon, pitävät sisällensä tietoa lomitustöiden tiedoista. Esimerkiksi mitä konetta kyseisessä työssä tarvitaan, minkä merkinen kone on, kuka tekee pääsääntöisesti kyseisen työn ja minkälaista osaamistasoa lomittajalta vaaditaan.



Työkone taulusta löytyy myös ohjevideoille oma sarakkeensa, jonne voi tallentaa minkälaisia ohjevideoita tahansa. Vaikka tässä työssä on mahdollisuus tallentaa ohjevideoita tietokantaan, niin ohjelmaa ei ole ohjelmoitu käsittelemään videoita millään tavalla.

6.3 Tietokannan laajentaminen

Kun tietokanta luotiin pyrimme siihen, että sitä pystyisi jatkokehittämään. Siksi monissa taulukoissa on sarakkeita, joita ei ole käytetty itse ohjelmassa millään tavalla. Kuten esimerkiksi ohjevideo sarake.

Taulukoiden muuttaminen onnistuu todella helposti joko ottamalla viimeisin versio auki MySQL Workbench:llä ja muokkaamalla tietokantaa haluamallansa tavalla. Muokkauksen jälkeen koko kannan päivittäminen palvelimelle onnistuu helposti palvelimella sijaitsevalla phpMyAdmin hallintatyökalulla.

Taulukoiden muuttaminen ja taulukoihin tietojen lisääminen onnistuu myös phpMyAdmin hallintatyökalulla todella helposti ilman, että joutuu käyttämään mitään muuta ohjelmaa.

7 TESTAUS

Ohjelma on testattu pääasiassa täyttämällä tietokannan kaikki taulukot erilaisilla kirjain- ja numeroyhdistelmillä ja ajamalla koko ohjelmaa, niin kuin käyttäjä käyttäisi ohjelmaa. Jos virheitä huomattiin, ne korjattiin välittömästi ja testattiin uudelleen niin kauan kuin virheet saatiin korjattua.

Ohjelman testauksesta ei ole tehty erillistä testauspöytäkirjaa.

Työssä ei ollut suuria ongelmia tai suuritöisiä virheitä. Suurimmat syyt ohjelman virheille oli kirjoitusvirheet. Esimerkiksi erinimiset muuttujat eri ohjelmissa. Kun yritti kutsua muuttujaa tai funktiota toisen ohjelman yli niin muuttujaa tai funktiota ei löytynyt.

8 TIETOTURVA

Ohjelmassa ei pahemmin käytetty erilaisia suojausmenetelmiä. Koska ohjelma toteutettiin nopealla aikataululla ja emme kerinneet panostaa ohjelman tietoturvaan niin paljoa kuin lopullinen versio ohjelmasta olisi vaatinut.

Ohjelmassa on käytetty perussuojausmenetelmiä. Esimerkiksi tietokannan serverinimi, käyttäjätunnus ja salasana sijaitsevat eri paikassa, missä ohjelman indeksi tiedosto sijaitsee. Koska jos palvelin kaatuu syystä tai toisesta, niin tietokannan tunnukset ja salasanat eivät tule esille.

Ohjelmassa on myös pienimuotoinen puhelimen tunnistamismenetelmä jolla pyrittiin, että oikea puhelin pääsee sisäänkirjautumaan ohjelmaan ja löytämään oikeat tiedot tietokannasta.

Suunnittelimme, että olisimme salanneet kaikki tiedot mitä tietokantaan tallennetaan, jotta jos vika tai jossain muussa tilanteessa kaikki tiedot eivät olisi suoraan luettavassa muodossa tietokannassa.

9 YHTEENVETO

Työn tavoitteena oli toteuttaa demoversio lomittajille tarkoitettua sähköisestä työn hallinta työkalusta. Työn aiheen saimme koululta ja en itse päässyt koskaan tapaamaan maatalanlomittajia, jotka olisivat saaneet kertoa oman mielipiteensä ohjelman toimivuudesta.

En itse myöskään päässyt paikalle kun työparini esitteli ohjelman opettajille. Palautteen perusteella ohjelmamme toimi todella hyvin ja he olivat tyytyväisiä työn tulokseen. Joten onnistuimme tavoitteessa ja saimme ohjelman valmiiksi ajallansa.

Ohjelman suunnitteluvaiheessa oli paljon erilaisia ongelmia. En tiedä johtuivatko erilaiset näkemykset ohjelmasta ohjelman suunnitteluvaiheessa, kielimuurin takia vai mikä oli syy. Mutta kaikki ongelmat saatiin ratkottua.

Työpari oli vaihto-opiskelija ja hänen äidinkieltensä ei ollut suomi, joten jouduimme käymään kaikki keskustelut englannin kielellä. Myös kaikki neuvottelut joissa oli opettajia mukana, käytiin englanniksi.

LÄHTEET

Wikipedia. PHP. Artikkel.

<http://fi.wikipedia.org/wiki/PHP>

Wikipedia. Tietokanta. Artikkel.

<http://fi.wikipedia.org/wiki/Tietokanta>

Wikipedia. Maatalouslomittaja. Artikkel.

<http://fi.wikipedia.org/wiki/Maatalouslomittaja>

QT-kotisivu. Päivitetty 27.5.2013. Luettu 27.5.2013.

<http://qt.digia.com/>

LIITTEET

Liite 1. PHP-pääohjelman lähdekoodi.

Liite 1 1(4)

```

<?php
include("../conf.php");
include("xml.php");

if(isset($_POST['name'])){
$confirmid = 0;
if($_POST['name'] != ""){
$tmp = mysql_query("SELECT contract_idcontract, contract_phone FROM contract_has_standin",$connect);

while($row = mysql_fetch_assoc($tmp)){
$confirmnumber = $row["contract_phone"];
if($confirmnumber == $_POST['name']){
$conid = $row["contract_idcontract"];
$confirmid = 1;
}
if($_POST['name'] == "logout".$confirmnumber){
filecleaner();
xmltmp5();
}

}}
if($confirmid == 1){
xmltmp6();

//finding all information nessessary information from contract
$tmp2 = mysql_query("SELECT standin_form_iddetails, farm_id, instructions_location, manual_location from contract where
idcontract = ".$conid."",$connect);

//taking information to attributes for xml
while($row2 = mysql_fetch_assoc($tmp2)){
$standingid = $row2["standin_form_iddetails"];
$farmid = $row2["farm_id"];
$instruction = $row2["instructions_location"];
>manual = $row2["manual_location"];
}

//finding all nessessary information from farm and permanent_employees

$tmp3 = mysql_query("Select farm.name as fname, permanent_employees.name as pname , permanent_employees.phone_number
as pphone from farm inner join permanent_employees where ".$farmid." = permanent_employees.farm_id and ".$farmid." = farm.id
");
while($row3 = mysql_fetch_assoc($tmp3)){
$farmname = $row3["fname"];
$pname = $row3["pname"];
$pphone = $row3["pphone"];
}

```

Liite 1 2(4)

```

openfile();
$i = 1;
//finding all nessessary information from machinery
$tmp4 = mysql_query("Select * from machinery where standin_form_iddetails = ".$standingid." ");
while($row4 = mysql_fetch_assoc($tmp4)){
    $taskid = $row4["taskid"];
    $taskname = $row4["taskname"];
    xmltmp2($taskid,$taskname);
    $tmp5 = mysql_query("select name,instructions,video from machines where taskid =". $taskid ." ");
    while($row4 = mysql_fetch_assoc($tmp5)){
        $macname = $row4["name"];
        $macinstructions = $row4["instructions"];
        $macvideo = $row4["video"];
        xmltmp3($macname, $macinstructions, $macvideo, $i);
        $i++;
    }
    xmltmp4();
    $i = 1;
}
closefile();
xmltmp1($farmname,$penname,$pephone,$sinstruction,$smanual);
mysql_close($connect);
$confirmnid = 0;
}
}
?>

<?php
    $connect = mysql_connect("localhost","lombsane_root","");
    mysql_select_db("lombsane_database");
?>

<?php
    $_POST['name'];
    include("confirm.php");
?>

```

Liite 1 3(4)

```
<?php
```

```
function xmltmp1($a, $b, $c, $d, $e){
    $fh = fopen("/home/lompsane/public_html/xmlfiles/authbart farm.xml","w");
    $xml = "<?xml version='1.0'?>";
    $xml .= "<auth>";
    $xml .= "<access>1</access>";
    $xml .= "<farmname>".$a."</farmname>";
    $xml .= "<bossname>".$b."</bossname>";
    $xml .= "<bossnumber>".$c."</bossnumber>";
    $xml .= "<instructions>".$d."</instructions>";
    $xml .= "<manual>".$e."</manual>";
    $xml .= "</auth>";
    fwrite($fh,$xml);
    fclose($fh);
}
```

```
function openfile(){
    $fh2 = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml","w");
    $xml2 = "<?xml version='1.0'?>";
    $xml2 .= "<missions>";
    fwrite($fh2,$xml2);
    fclose($fh2);
}
```

```
function xmltmp2($taskid , $taskname){
    $fh3 = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml","a+");
    $xml3 = "<mission>";
    $xml3 .= "<id>".$taskid."</id>";
    $xml3 .= "<name>". $taskname . "</name>";
    $xml3 .= "<machines>";
    fwrite($fh3, $xml3);
    fclose($fh3);
}
```

```
function xmltmp3($macname, $macinstructions, $macvideo, $i){
    $fh33 = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml","a+");
    $xml4 = "<machine>";
    $xml4 .= "<id>".$i."</id>";
    $xml4 .= "<name>". $macname . "</name>";
    $xml4 .= "<instructions>";
    $xml4 .= "<instruction>";
    $xml4 .= "<id>1</id>";
}
```

```
$xml4 .= "<movie>" . $macvideo. "</movie>";
$xml4 .= "<name>" . $macinstructions. "</name>";
```

Liite 1 4(4)

```
$xml4 .= "</instruction></instructions></machine>";
fwrite($fh33,$xml4);
fclose($fh33); }
```

```
function xmltmp4(){
$fh4 = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml","a+");
$xml5 = "</machines></mission>";
fwrite($fh4,$xml5);
fclose($fh4); }
```

```
function closefile(){
$fh5 = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml","a+");
$xml6 = "</missions>";
fwrite($fh5,$xml6);
fclose($fh5); }
```

```
function filecleaner() {
    $clean = fopen("/home/lompsane/public_html/xmlfiles/missionbart farm.xml","w");
    $filler = "";
    fwrite($clean, $filler);
    fclose($clean);

    $clean = fopen("home/lompsane/public_html/xmlfiles/authbart farm.xml","w");
    fwrite($clean,$filler);
    fclose($clean); }
```

```
function xmltmp5(){
    $file = fopen("home/lompsane/public_html/xmlfiles/logoutbart farm.xml","w");
    $xml = "<?xml version='1.0'?>";
    $xml = "<logout>";
    $xml = "<access>1</access>";
    $xml = "</logout>";
    fwrite($file,$xml);
    fclose($file);
}
```

```
function xmltmp6(){
    $file = fopen("home/lompsane/public_html/xmlfiles/logoutbart farm.xml","w");
    $filler = "";
    fwrite($file,$filler);
    fclose($file); }
```

```
?>
```

Liite 2. Tietokannan lähdekoodi.

Liite 2 1(5)

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `lompsane_database` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ;
SHOW WARNINGS;
USE `lompsane_database` ;

-----
-- Table `table1`
-----

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `table1` (
  `idtable1` INT NOT NULL ,
  PRIMARY KEY (`idtable1`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `farm`
-----

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `farm` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(255) NOT NULL ,
  `address` VARCHAR(255) NOT NULL ,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `permanent_employees`
-----

DROP TABLE IF EXISTS `permanent_employees` ;

```

Liite 2 2(5)

SHOW WARNINGS;

```
CREATE TABLE IF NOT EXISTS `permanent_employees` (
  `idpermanentemployees` INT NOT NULL AUTO_INCREMENT ,
  `farm_id` INT NOT NULL ,
  `name` VARCHAR(255) NOT NULL ,
  `phone_number` VARCHAR(45) NOT NULL ,
  `task_percent` INT NOT NULL DEFAULT 0 ,
  PRIMARY KEY (`idpermanentemployees`, `farm_id`)
ENGINE = InnoDB;
```

SHOW WARNINGS;

```
-----
-- Table `standin_form`
-----
DROP TABLE IF EXISTS `standin_form` ;
```

SHOW WARNINGS;

```
CREATE TABLE IF NOT EXISTS `standin_form` (
  `iddetails` INT NOT NULL AUTO_INCREMENT ,
  `skill` VARCHAR(255) NOT NULL ,
  `exceptional_case` VARCHAR(255) NOT NULL ,
  `seasonal_works` VARCHAR(255) NOT NULL ,
  `deputy_standin` VARCHAR(255) NOT NULL ,
  `other_information` VARCHAR(255) NOT NULL ,
  PRIMARY KEY (`iddetails`)
ENGINE = InnoDB;
```

SHOW WARNINGS;

```
-----
-- Table `contract`
-----
DROP TABLE IF EXISTS `contract` ;
```

SHOW WARNINGS;

```
CREATE TABLE IF NOT EXISTS `contract` (
  `idcontract` INT NOT NULL AUTO_INCREMENT ,
  `farm_id` INT NOT NULL ,
  `standin_form_iddetails` INT NOT NULL ,
  `task_percent1` INT NOT NULL DEFAULT 0 ,
  `task_percent2` INT NOT NULL DEFAULT 0 ,
  `attachment1` INT NOT NULL DEFAULT 1 COMMENT 'attachment 0 permanent 1 temporary' ,
  `attachment2` INT NOT NULL DEFAULT 1 COMMENT 'attachment 0 permanent 1 temporary' ,
```

Liite 2 3(5)

```

`begindate1` DATE NOT NULL ,
`begindate2` DATE NOT NULL ,
`enddate1` DATE NOT NULL ,
`enddate2` DATE NOT NULL ,
`hours` INT NOT NULL ,
`works_with` INT NOT NULL DEFAULT 0 COMMENT '0 alone 1 with another standin 2 with permanent worker'
,
`pworker` INT NOT NULL COMMENT '0 doesn\'t work with permanent worker 1 will work with permanent worker'
,
`pworker_name` VARCHAR(45) NOT NULL ,
`deputy` INT NOT NULL DEFAULT 0 COMMENT '0 doesn\'t work with deputy 1 will work with deputy' ,
`deputy_name` VARCHAR(45) NOT NULL ,
`deputy_phone` VARCHAR(45) NOT NULL ,
`TSS_time` TIME NOT NULL ,
`TSS_created` DATE NOT NULL ,
`same_vacationtime` TIME NOT NULL ,
`svacation_created` DATE NOT NULL ,
`work_measure` TIME NOT NULL ,
`wmeasure_created` DATE NOT NULL ,
`written_instructions` INT NOT NULL DEFAULT 0 COMMENT '0 yes 1 no' ,
`instructions_location` VARCHAR(90) NOT NULL ,
`quality_manual` INT NOT NULL DEFAULT 0 COMMENT '0 yes 1 no' ,
`manual_location` VARCHAR(90) NOT NULL ,
`phone_list` INT NOT NULL DEFAULT 0 COMMENT '0 yes 1 no' ,
`list_location` VARCHAR(90) NOT NULL ,
`social_facilities` INT NOT NULL DEFAULT 0 COMMENT '0 yes 1 no' ,
`circumstance_date` DATE NOT NULL ,
`circumstance_fixes` VARCHAR(255) NOT NULL ,
PRIMARY KEY (`idcontract`, `farm_id`, `standin_form_iddetails`)
ENGINE = InnoDB;

```

```
SHOW WARNINGS;
```

```
-----
-- Table `standin`
-----
```

```
DROP TABLE IF EXISTS `standin` ;
```

```
SHOW WARNINGS;
```

```
CREATE TABLE IF NOT EXISTS `standin` (
  `idstandin` INT NOT NULL AUTO_INCREMENT ,
  `name` VARCHAR(45) NOT NULL ,
  `adress` VARCHAR(45) NOT NULL ,
  `phone` VARCHAR(45) NOT NULL ,

```


Liite 2 4(5)

```
`timestamp` TIMESTAMP NOT NULL ,
  PRIMARY KEY (`idstandin`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `machinery`
-----
DROP TABLE IF EXISTS `machinery` ;
```

```
SHOW WARNINGS;
```

```
CREATE TABLE IF NOT EXISTS `machinery` (
  `idmachinery` INT NOT NULL AUTO_INCREMENT ,
  `standin_form_iddetails` INT NOT NULL ,
  `machine_task` VARCHAR(45) NOT NULL ,
  `model_program_task` VARCHAR(255) NOT NULL ,
  `who_permanentworker` VARCHAR(90) NOT NULL ,
  `standin_working` INT NOT NULL DEFAULT 0 COMMENT '0 empty 1 need to know how 2 no need to know
how 3 no use' ,
  `guidance_videos` BLOB NOT NULL ,
  PRIMARY KEY (`idmachinery`, `standin_form_iddetails`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `contract_has_standin`
-----
DROP TABLE IF EXISTS `contract_has_standin` ;
```

```
SHOW WARNINGS;
```

```
CREATE TABLE IF NOT EXISTS `contract_has_standin` (
  `contract_idcontract` INT NOT NULL ,
  `contract_farm_id` INT NOT NULL ,
  `standin_idstandin1` INT NOT NULL ,
  `standin_number` INT NOT NULL DEFAULT 1 COMMENT '1 first worker 2 second worker' ,
  PRIMARY KEY (`contract_idcontract`, `contract_farm_id`, `standin_idstandin1`))
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-----
-- Table `standin_timetable`
```

Liite 2 5(5)

```
-----  
DROP TABLE IF EXISTS `standin_timetable` ;  
  
SHOW WARNINGS;  
CREATE TABLE IF NOT EXISTS `standin_timetable` (  
  `idstandin_timetable` INT NOT NULL AUTO_INCREMENT ,  
  `standin_idstandin1` INT NOT NULL ,  
  `date` DATE NOT NULL ,  
  `starttime` TIME NOT NULL ,  
  `endtime` TIME NOT NULL ,  
  PRIMARY KEY (`idstandin_timetable`, `standin_idstandin1`))  
ENGINE = InnoDB;  
  
SHOW WARNINGS;  
  
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Liite 3. Valmiin mobiilisovelluksen tilannekuvat.

Liite 3 1(4)



KUVA 1. Aloitusnäky



KUVA 2. Epäonnistunut sisäänkirjautuminen

Liite 3 1(4)



KUVA 3. Perusohjeet

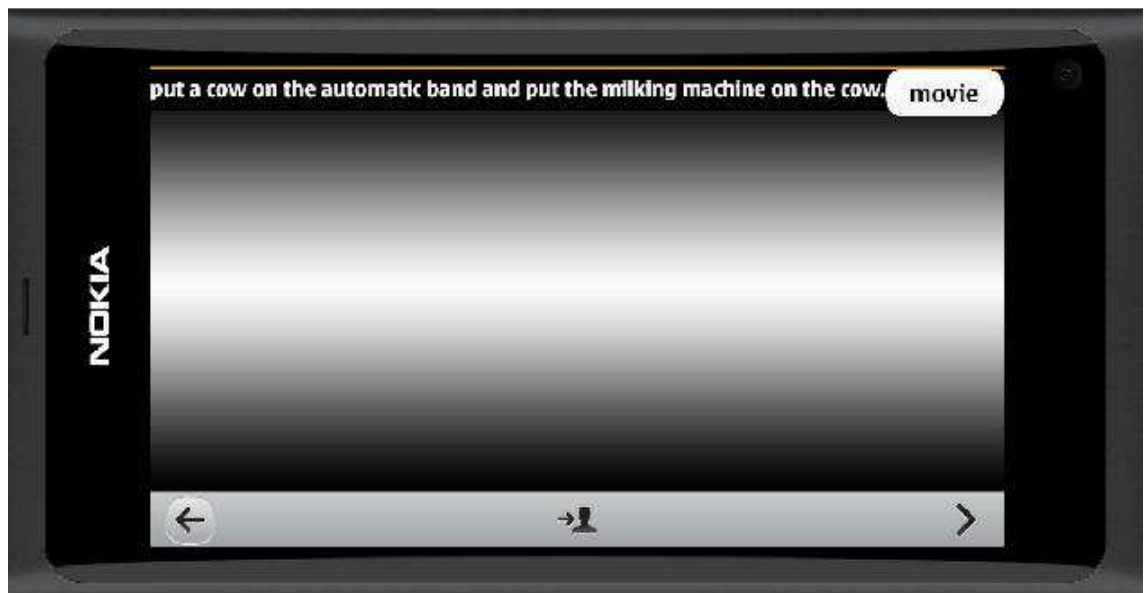


KUVA 4. Maatalouslomittajan työtehtävät

Liite 3 3(4)



KUVA 5. Työvälineet



KUVA 6. Työvälineiden ohjeet

Liite 3 4(4)



KUVA 7. Opetusvideo



KUVA 8. Uloskirjautuminen