



SAVONIA

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

ELMERI+ JÄRJESTELMÄ

Työturvallisuusauditoinnin digitalisointi

TEKIJÄ: Jussi Nivamo

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Jussi Nivamo	
Työn nimi ELMERI+ työturvallisuusauditoinnin digitalisointi	
Päiväys 9.12.2014	Sivumäärä/Liitteet 26
Ohjaaja(t) Lehtori Sami Lahti / Lehtori Jussi Koistinen	
Toimeksiantaja/Yhteistyökumppani(t) Lujabetoni Oy	
<p>Tiivistelmä</p> <p>Tämä työ tehtiin projektityönä Lujabetoni Oy:lle suunnittelemalla ja toteuttamalla kolme eri ohjelmaa kattava ohjelmistokokonaisuus. Ohjelmistokokonaisuuden tarkoituksena oli modernisoida Lujabetonin tuotantolaitoksilla jo pitkään käytössä ollut ELMERI+-työturvallisuusauditointi tuomalla se tableteilla käytettäväksi sovellukseksi.</p> <p>Tuotettavia ohjelmia vaadittiin yksi Android-tableteilla käytettäväksi työturvallisuusauditointikierrosten tekemiseen, yksi ohjelma palvelinrajapinnaksi mobiililaitteen ja sisäverkon tietokantojen välille sekä yksi ohjelma sisäverkossa tapahtuville mittausten tarkastelulle ja raporttien luomiselle.</p> <p>Työssä ohjelmoitiin Java- ja C#-kielillä, sekä suunniteltiin käyttöliittymiä Android- ja Windows työpöytä-ympäristöön. Java-ohjelmointi sekä Android-käyttöliittymä tehtiin Eclipse-kehitysympäristössä. C#-ohjelmointi ja Windows työpöytä-käyttöliittymät tehtiin Visual Studio 2012-kehitysympäristössä. Työn suunnittelussa ja toteutuksessa kesti noin kolme kuukautta.</p> <p>Projektin tuloksena oli tämä raportti, sekä teolliseen käyttöön soveltuva palvelukokonaisuus jota on tarkoitus käyttää tulevaisuudessa kaikissa Lujabetoni Oy:n tuotantolaitoksissa ELMERI+-mittausten tekemiseen.</p>	
Avainsanat Android, työturvallisuus, elmeri+	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Jussi Nivamo			
Title of Thesis Digitalization of ELMERI+ Work Safety Audition Standard			
Date	9 december 2014	Pages/Appendices	20
Supervisor(s) Mr. Sami Lahti, Lecturer / Mr. Jussi Koistinen, Lecturer			
Client Organisation / Partners Lujabetoni Ltd			
<p>Abstract</p> <p>This project was carried out by designing and producing a service consisting of three separate computer programs for Lujabetoni Ltd. The purpose of the service was to modernize the ELMERI+ work safety audit standard currently in use in Lujabetoni's manufacturing establishments by giving the ability to perform safety audits via mobile application.</p> <p>Three programs were required for the operation of the service. The first one was the mobile application for performing the actual audits, the second was a server interface program for bridging communications between the mobile application and intranet databases, and the third required program was intended for viewing and generating reports of past audits.</p> <p>The programming was done in Java and C# and the user interfaces were designed for Android and Windows desktop environments. Java programming and the Android user interfaces were produced in the Eclipse integrated development environment. The design of the Windows desktop user interface and all C# programming were done in the Visual Studio 2012 integrated development environment. The processes of system design and system production were conducted sequentially and the whole project took approximately three months to complete.</p> <p>As a result of the project the new ELMERI+ mobile work safety audit service is going to be implemented in all Lujabetoni's production facilities.</p>			
Keywords Android, work safety, elmeri+			

SISÄLTÖ

1	JOHDANTO	5
2	KÄYTETYT TEKNIIKAT JA TYÖKALUT	6
2.1	Activity (Android)	6
2.2	Android	6
2.3	ASP.NET	6
2.4	C#-ohjelmointikieli	6
2.5	Eclipse-ohjelmointiympäristö.....	6
2.6	HTTPS-protokolla	6
2.7	IIS – Internet Information Services-palvelinympäristö	7
2.8	Java-ohjelmointikieli.....	7
2.9	Microsoft SQL Server Management Studio 2012	7
2.10	Microsoft Visual Studio	7
2.11	Microsoft Windows Server	7
2.12	SQL.....	8
2.13	XML - Extensible Markup Language	8
3	ELMERI+ MITTAUS.....	9
4	RISKIPAIKKAILMOITUS.....	9
5	PROJEKTIN MÄÄRITTELY JA SUUNNITTELU.....	11
5.1	Android-sovellus	11
5.2	ASP.NET-Palvelinsovellus.....	12
5.3	TJR-sovellus	12
6	PALVELUN TOTEUTUS	13
6.1	Android-sovellus	13
6.2	TJR-sovellus	19
6.3	ASP.NET-Palvelinsovellus.....	22
7	TESTAUS	24
8	YHTEENVETO.....	25
	LÄHTEET	26

1 JOHDANTO

Tässä opinnäytetyössä selvitetään, kuinka ELMERI+-työturvallisuusmittauksen digitalisointi suunniteltiin sekä toteutettiin. ELMERI+ on opinnäytetyön tilaajana toimivalla Lujabetoni Oy:ssä yli vuosikymmenen ajan käytössä ollut yhtiön sisäinen betonielementtejä valmistavien tehtaiden työturvallisuusauditointimenetelmä. (Lujabetoni 2014.)

ELMERI+-mittauksia on tehty toistaiseksi vain paperille tulostetulle taulukkopohjalle, johon mittaus-tulokset on merkitty kynällä. Mittaustulosten kirjaamisen jälkeen mittausdokumentti on kirjoitettu puhtaaksi tietokoneella taulukkolaskentaohjelmassa ja arkistoitu digitaalisesti. Vanha paperille tehtävä auditointi haluttiin muuttaa Android-tablettilaitteilla käytettäväksi ja kerätyt tiedot haluttiin keskitää tietokantaan, koska järjestelmästä oli hankala saada monipuolisia raportteja ja tilastotietoja.

Tableteilla käytettävän ohjelman tueksi täytyi tuottaa palvelinohjelmisto ottamaan vastaan ja tallentamaan mittauksia, sekä kommunikoidaan muuten Android-sovelluksen kanssa. Lisäksi tarvittiin yrityksen sisäverkossa toimiva työpöytäkäyttöön soveltuva ohjelma, jolla tehtyjä ELMERI+-mittauksia on mahdollista tarkastella ja tulostaa niistä raportteja.

Uuden Android-tableteilla toimiva ELMERI+ mittausjärjestelmä otettiin käyttöön syksyllä 2014.

2 KÄYTETYT TEKNIIKAT JA TYÖKALUT

2.1 Activity (Android)

Java-kielellä toteutetut Android-sovellukset perustuvat pääsääntöisesti activityihin, jotka aukeavat yksi kerrallaan laitteen ruudulle. Sivulla navigoimisen hierarkia on samantapainen kuin html-sivuilla internetissä. Merkittävänä erona verkkosivuihin on että ohjelman luoja ja käyttöjärjestelmä määrittelevät tarkasti, miltä sivuilta on mahdollista siirtyä millekin sivuille. (Google 2014a.)

2.2 Android

Android on pääasiassa kannettaville tietokonelaitteille suunniteltu Linux-pohjainen käyttöjärjestelmä. (Google 2014b.)

2.3 ASP.NET

Active Server Page (ASP) on Microsoftin kehittämä .NET-tuoteperheeseen kuuluva viitekehysjärjestelmä, joka on suunniteltu erityisesti verkkopalveluiden toteuttamiseen. ASP.NET sisältää sarjan työkaluja ja valmiita luokkakirjastoja dynaamisten verkkosivujen ja palvelinympäristöjen rakentamista varten. (Moghadampour 2011, 15.)

2.4 C#-ohjelmointikieli

Microsoftin kehittämä C# (C-Sharp) on vanhemmasta C++-kielestä sekä modernimmasta Java-kielestä piirteitä ja syntaksia lainaava ohjelmointikieli. C# on tyyliltään vahvasti tyypitetty olio-ohjelmointikieli, jonka muistinhallinta on automatisoitu suoritusympäristön sisäisen roskienkeräysjärjestelmän (Garbage Collector) avulla. (Moghadampour 2011, 12.)

2.5 Eclipse-ohjelmointiympäristö

Eclipse on Eclipse foundationin kehittämä kehitysympäristö, johon on mahdollista integroida lisäominaisuuksia sekä liittymiä ulkopuolisiin järjestelmiin. Yksi tämän työn kannalta keskeinen lisäominaisuus oli Eclipseen liitettävä Android Development Tools-plugin (ADT), jonka avulla on mahdollista tuottaa Android-ohjelmia Java-kielellä sekä emuloida Android-laitteita tietokoneella testikäytössä. (Eclipse Foundation 2014.)

2.6 HTTPS-protokolla

Alun perin verkkosivujen siirtämiseen suunniteltu HyperText Transfer Protocol (HTTP) kehitettiin TCP/IP-protokollan päälle luotettavaksi tekniikaksi hakea palvelimilta verkkosivuja loppukäyttäjän tietokoneella esitettäväksi. HyperText Transfer Protocol Secure (HTTPS) on HTTP-protokollan muunnos, joka käyttää Secure Sockets Layer (SSL)-tekniikkaa protokollan sisällä liikkuvan tiedon salaamiseen. (IETF 1996.)

2.7 IIS – Internet Information Services-palvelinympäristö

IIS on Microsoftin kehittämä palvelinohjelmistokokonaisuus, joka on tarkoitettu käytettäväksi Windows-pohjaisissa palvelimissa. IIS:ään on integroitu oletusarvoisesti yleisimmät verkkopalvelinten käytössä olevat palveluprotokollat. Esimerkiksi File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) sekä verkkosivujen esittämiseen liittyvät HTTP- ja HTTPS-protokollat. Verkkosivujen esittämisen lisäksi HTTP(S)-protokollilla voidaan liikuttaa tehokkaasti ja luotettavasti tietoa palvelimen ja asiakaslaitteen välillä. (Microsoft 2014a.)

2.8 Java-ohjelmointikieli

Java on alustariippumaton käyttöjärjestelmän yhteyteen asennettavalla virtuaalikoneella suoritettava olio-pohjainen ohjelmointikieli. Java käännetään yleensä vasta suoritettaessa just-in-time-kääntäjien avulla etukäteen Java-kääntäjällä tuotetusta bytecode-”välikielestä”, mikä tekee kielestä monissa tilanteissa raskaamman suorittaa. Etuna Javan käytössä on koodin alustariippumattomuus. (Kolehmainen 2006.)

2.9 Microsoft SQL Server Management Studio 2012

SQL Server Management Studio (SSMS) on Microsoftin kehittämä tietokantojen hallintajärjestelmä. SSMS:ssä on sisäänrakennettuna tuki yleisimmin käytetyille SQL-kyselykielille (T-SQL ja ANSI SQL), ja siinä on normaalien tietokantojen ylläpito- sekä hallintaominaisuuksien lisäksi analysointityökaluja vaativaan yrityskäyttöön. (Microsoft 2014b.)

2.10 Microsoft Visual Studio

Visual Studio on Microsoftin kehittämä integroitu kehitysympäristö. Ympäristö tukee oletuksena useita yleisiä ohjelmointikieliä, kuten C#, C, C++, sekä siihen on sisäänrakennettuna liittymät keskitettyjen versionhallintajärjestelmien käyttöön. Yhteenliittymät muihin Microsoftin järjestelmiin on suunniteltu helppokäyttöisiksi ja Visual Studiolla on myös mahdollista luoda Object-Relational Mapping-tyyppisiä tietokantayhteyksiä. (Moghadampour 2011, 11, 360.)

2.11 Microsoft Windows Server

Microsoft Corporationin tekemä palvelinkäyttöön tarkoitettu käyttöjärjestelmien tuoteperhe. Windows Server-tuotteissa on useita sisäänrakennettuja ominaisuuksia ja palveluita yleisimpien palvelinratkaisujen toteuttamiseksi. (Microsoft 2014c.)

2.12 SQL

Structured Query Language (SQL) on alun perin IBM:n kehittämä standardisoitu kyselykieli relaatio-tietokantojen luomista, ylläpitoa ja muokkaamista varten. SQL-kielestä on tehty useita eri variaatioita eri tarkoituksiin. Esimerkiksi Android-laitteisiin on sisäänrakennettu ominaisuuksiltaan riisuttu SQLite sekä Microsoftin palvelinsovelluksissa käytettävä Transact SQL (T-SQL). Kieli on käyttöalustasta riippumaton. Yleisesti hyväksytyjen hyvien SQL-käytäntöjen mukaisesti tietokannoissa oleva tieto on normalisoitu niin että kantaan tallennettavaan tietoon liittyvät käsitteet on jaettu mahdollisimman laajalti omiin tauluihinsa. (Lahtonen 2003, 2-8.)

2.13 XML - Extensible Markup Language

XML on standardisoitu tiedon tallennusmenetelmä joka pyrkii säilyttämään tiedon luettavana yhtä aikaa niin ihmisille kuin tietokoneellekin. XML:ää käytetään paljon tiedon siirtämisessä tietoverkkojen yli sekä erilaisissa tallennusratkaisuissa. (Walkama ja Laakkonen 2004, 1 – 3.)

3 ELMERI+ MITTAUS

Työturvallisuusmittaus tehdään säännöllisin väliajoin kaikilla Lujabetonin tehtailla. Mittauksessa kierretään tutkittava alue ja tehdään merkintöjä paperiseen mittauspohjaan (Kuvio 1) sanoko äikän maikka mitään pitääkö olla kuva vai kuvio???. Mittauspohjaan tehdyt merkinnät ovat

- mittauksen suorittajan nimi tai nimimerkki
- muut mittaukseen osallistuvat henkilöt
- mittauksen ajankohta
- paikka jossa mittaus suoritetaan
- mittauksen aikana havaitut työturvallisuusriskit
- 15 eri osa-alueeseen jakautuvat mittaustulokset.

Mittaustulokset ovat työntekijöiden työoloista tehtyjä havaintoja. Havaintoja kirjatessa arvioidaan työskentelevien ihmisten työskentelymenetelmiä sekä työoloja 15 eri osa-alueen näkökulmasta. Yhtä työpistettä tarkastellessa tutkittavan osa-alueen ollessa kunnossa lisätään yksi *kunnossa*-merkintä osa-alueen merkitsevälle riville mittauspohjaan. Vastaavasti jos työskentelyssä tai työtiloissa havaitaan puutteita tai riskejä, lisätään *ei kunnossa*-sarakeeseen merkintä.

Paperille tehty mittaus on kopioitu tietokoneelle taulukkolaskentapohjalle, minkä jälkeen paperi sekä tietokoneelle kopioidut tiedot on arkistoitu.

4 RISKIPAikkAILMOITUS

Riskipaikkailmoitus on Lujabetonin sisäinen konsepti työturvallisuusriskien ilmoittamisesta. Ilmoitukset on toistaiseksi tehty vain Lujabetonin sisäverkon toimintajärjestelmässä (TJR) sijaitsevalla riskipaikkailmoitus-sovelluksella. Riskipaikkailmoitukseen merkitään missä työturvallisuusriski on havaittu, kuka riskin havaitsi, kuvaus riskistä ja riskin vakavuuden taso.

Riskipaikkailmoitukset kirjautuvat Lujabetonin tietokantaan niin että riskin havaintopaikasta vastavalle esimiehelle lähtee riskin vakavuuden perusteella määräytyvän määräajan päästä automatisoitu sähköposti, jossa pyydetään alueen vastuuhenkilöä kuittausta ilmoituksesta. Näin varmistetaan työturvallisuusriskeihin puuttuminen esimiesten toimesta.

Havainnoija: JNIV

Tehdas/halli:

Pvm: 1.10.2013

Havaintoalue: Halli 1

Havaintokohteet	Osa- indeksi	Kunnossa	yht.	Ei kunnossa	yht.	Ei havaintoa	yht.
Työskentely	40 %		2		3		
1. Riskinotto, suojaimet, vaatetus		2		3			
Ergonomia	80 %		4		1		
2. Fyysinen kuormitus		3		1			
3. Työpisteen ja välineiden ergonomisuus		1		0			
Kone- ja laiteturvallisuus	63 %		5		3		
4. Koneiden kunto ja suojalaitteet		4		1			
5. Koneiden hallintalaitteet ja merkinnät		1		2			
Liikumisturvallisuus	100 %		5				
6. Kulkutiet/lattiat rakenne, putoamissuojaus		3		0			
7. Poistumistiet		2		0			
Järjestys	78 %		7		2		
8. Kulkuteiden ja lattioiden järjestys		2		0			
9. Pöydät, päällykset ja hyllyt		4		1			
10. Jäteastiat		1		1			
11. Tupakointipaikka				0			
Työympäristötekijät	75 %		6		2		
12. Melu		3		0			
13. Valaistus		2		0			
14. Lämpöolosuhteet		1		2			
15. Ilman puhtaus ja käsiteltävät aineet				0			
		yhteensä:	29	yhteensä:	11	yhteensä:	0
ELMERI-indeksi =		$\frac{\text{kunnossa}}{\text{kunnossa + ei kunnossa}} \times 100 \% =$	$\frac{29}{40} =$	73	$\%$		
Muistiinpanot havaintokohteittain:							
1	Hallin lämpötila liian korkea työskentelylle						
2	Työntekijöiden ergonomiaan kiinnitettävä huomiota						
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							

KUVIO 1. Vanha ELMERI+-mittauspohja.

5 PROJEKTIN MÄÄRITTELY JA SUUNNITTELU

Järjestelmä jakautuu kolmeen eri sovellukseen. Android-tabletille kehitettävään ELMERI+-mittaussovellukseen, yrityksen tietoverkosta internetiin auki olevaan palvelinsovellukseen sekä yrityksen sisäverkon toimintajärjestelmään (TJR) liittyvään Windows työpöytäsovellukseen.

Android-sovellus päätettiin toteuttaa Eclipse-kehitysympäristössä Java-kielellä. Konsernissa ei ollut etukäteen mitään työkaluja Android-sovellusten toteuttamista varten eikä kaupallisten sovellusten hankintaa koettu tarpeelliseksi sillä Eclipse on ilmaisohjelmaksi riittävän monipuolinen ja helppokäyttöinen tätä projektia varten.

Palvelinsovellus suunniteltiin toimimaan Windows Server 2012-palvelinkäyttöjärjestelmällä, sillä konsernilla oli jo aiempaa kokemusta Windows Server-tuoteperheen aiemmista versioista. Uusi palvelinsovellus oli myös tilaisuus siirtyä koekäyttämään uutta palvelinkäyttöjärjestelmää. Konsernin aiempia palvelinsovelluksia oli toteutettu Microsoftin IIS-verkkopalvelinalustalla Windows Serverissä, joten ylläpidon helpottamiseksi päätettiin tässäkin projektissa käyttää IIS:ää. Palvelinsovellus toteutettiin käyttäen C#-kieltä ja ASP.NET 4.0-tekniikkaa.

TJR-sovellus suunniteltiin toimimaan Windows-työasemilla sillä Windows on konsernin sisäisen virtualisointipalvelimen oletuskäyttöjärjestelmä jonka kautta kaikkia konsernin ohjelmia käytetään. Sovellus kehitettiin C#-kielellä käyttäen Windows Forms-ohjelmistokomponenttia käyttöliittymän rakentamiseen.

5.1 Android-sovellus

Android-sovelluksen tarkoituksena on olla käytännössä vanha ELMERI+-mittauspohja, tabletilta käytettävänä digitalisoituna versiona. Ohjelmalla on kyettävä tekemään vanhaa kuviossa 1 esiintyvää mittauspohjaa vastaavia työturvallisuusmittauksia missä tahansa Lujabetonin tuotantoyksikössä. Koska ohjelmaa käytetään pääasiassa tehdasympäristössä, täytyy mittaukset voida tehdä yhteydetöissä tilassa. Tehdyt mittaukset kuitenkin lähetetään internetiin yli kun laite kytketään langattomaan lähiverkkoon.

ELMERI+-mittausten lisäksi ylimääräisenä ominaisuutena ohjelmaan haluttiin mahdollisuus liittää välittömistä vaaratekijöistä kertovia riskipaikkailmoituksia. Riskipaikkailmoituksista puhutaan ohjelmassa toimenpiteinä sillä kirjatut ilmoitukset tulostetaan TJR-ohjelman avulla ja asetetaan tuotantotiloihin esille korjattavina työturvallisuustoimenpiteinä.

Vaikka riskipaikkailmoituksia pystyttiin jo aiemmin tekemään yrityksen sisäverkossa käytettävällä riskipaikkailmoitus-ohjelmalla, ja niihin oli mahdollisuus liittää digitaalikameralla otettuja valokuvia, oli käytännössä valokuvien liittäminen niin hankalaa ettei ominaisuutta oltu käytetty aiemmin ollenkaan. Aika pitkä lause Android-ohjelman riskipaikkailmoitus-työkaluun haluttiin mahdollisuus liittää valokuvia riskipaikkailmoituksiin.

5.2 ASP.NET-Palvelinsovellus

Palvelinsovellus toimii yhtiön sisäverkossa toimivien tietokantojen ja mobiilisovelluksen välisenä rajapintana. Palvelimesta on auki yhteys yhtiön sisäverkkoon, missä se odottaa passiivisesti yhteyttä mobiilisovellukselta. Palvelin hyväksyy pelkästään salatut HTTPS-yhteydet, jotka autentikoidaan sertifioidulla paikallisella nimipalvelimella. Lähetettäessä mittauksia ja riskipaikkailmoituksia mobiililaitteelta palvelimelle palvelin tallentaa vastaanotetut mittaustiedot sekä mittausten yhteydessä otetut valokuvat yhtiön tietokantaan.

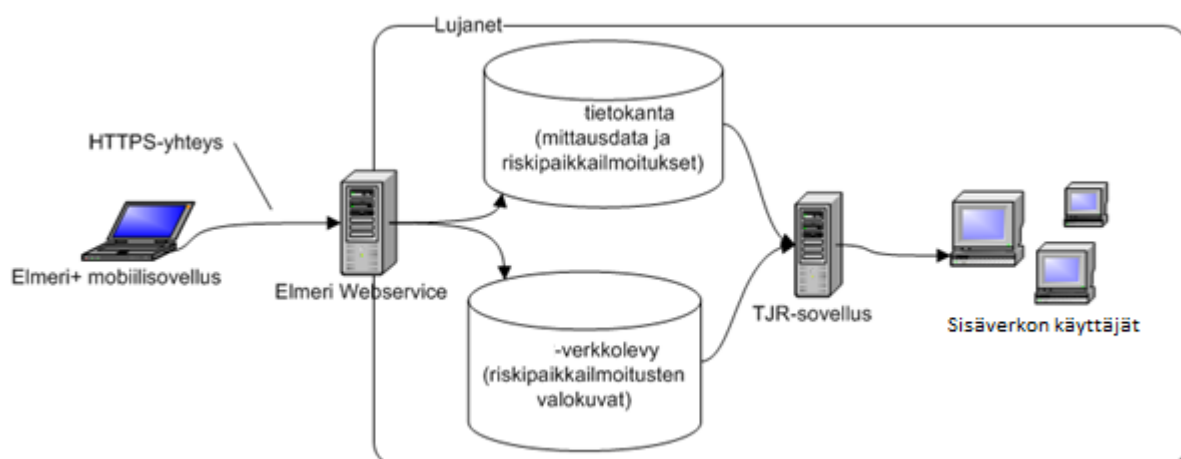
5.3 TJR-sovellus

Yhtiön sisäverkossa toimivan TJR-toimintajärjestelmään liittyvä Windows Formsilla toteutettu työkalu, jolla Android-sovelluksella lähetettyjä mittauksia on mahdollista tarkastella. TJR-sovelluksen kautta on mahdollista tuottaa kuvion 1 muotoisia raportteja vanhaa ELMERI+-mittauspohjaa käyttäen. Raportteja on mahdollista tuottaa yksittäisistä mittauskerroista, koontiraportteja useasta mittauskerrosta sekä vuosiraportteja joko tehtaan vastuualue-, tehdas- tai tuotantoyksikkökohtaisesti.

6 PALVELUN TOTEUTUS

Kokonaisuutena ohjelmistoista rakentui palvelu (Kuvio 2), jolla tehtailla valtuutetut henkilöt voivat tehdä ELMERI+-mittauksia ja lähettää niitä verkon yli palvelimelle, joka tallentaa ne yhtiön tietokantoihin. Järjestelmän loppukäyttäjät voivat lopulta tarkastella tallennetuista mittauksista luotuja raportteja helposti tulostettavassa MS Excel-taulukon muodossa (Kuvio 1).

Tietokantana yrityksen sisäverkossa toimii Microsoft SQL Server 2005, johon palvelun käyttämät taulut luotiin. Tietokannan käsittely- ja ylläpitosovelluksena käytettiin Microsoft SQL Server Manageria.

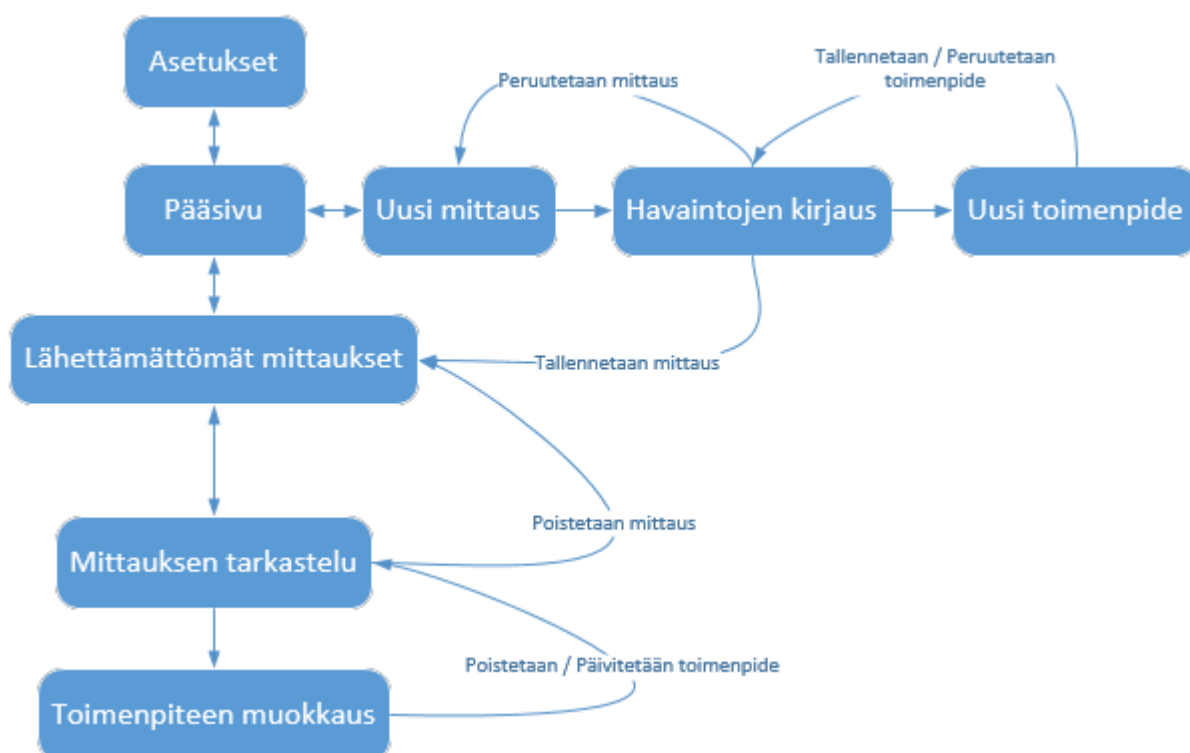


KUVIO 2. Järjestelmän yleiskuva.

6.1 Android-sovellus

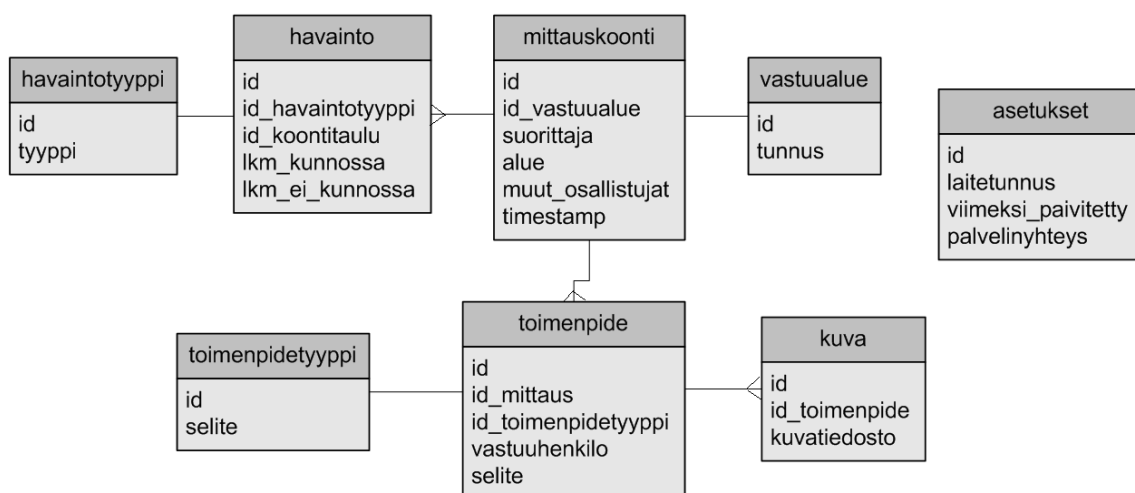
ELMERI+-mittausten tekemistä varten tarvittiin ohjelmaan selkeät eri osa-alueet uusien mittausten tekemiselle, lähettämättömien mittausten tarkastelulle sekä mittaustapahtumien aikana havaittujen työturvallisuusriskien kirjaamiselle riskipaikkailmoituksena. Jokainen sivu on kokonainen Activity, joiden ulkoasu on määritelty erikseen XML layout-tiedostoissa. Activityjen välillä navigoimiseen oli ohjelman käyttökokemuksen yksinkertaistamiseksi asetettava rajoituksia ja sääntöjä, jotta käyttäjä ei vahingossa navigoi Activityyn, josta ei pääse takaisin edelliseen vaikka käyttötapaus kyseisessä tilanteessa sitä vaatisi.

Android-ohjelmalla tehdyt mittaukset tallentuvat väliaikaisesti laitteen sisäiseen SQLite-tietokantaan (Kuvio 4) odottamaan lähetystä verkon yli palvelimelle. Keskeinen taulu tietokannassa on *mittauskoonti*, johon liittyvät kaikki yksittäiseen mittaukseen liittyvät taulut. Havainto-taulu vastaa yhtä kuviossa 1 esiintyvän mittauksen osa-alueen riviä, missä sarakkeet *lkm_kunnossa* sekä *lkm_ei_kunnossa* ovat suoritetun mittauksen yksittäisen osa-alueen positiiviset ja negatiiviset havainnot. Taulut *havaintotyyppi* sekä *toimenpidetyyppi* ovat tyyppitauluja tauluille *havainto* sekä *toimenpide*.



KUVIO 3. Android-ohjelman sisäinen Activity-hierarkia.

SQLite ei tietokanta-arkkitehtuurina tue päivämääriä tallentavia tietotyyppejä, joten mittausajan kertova *mittauskoonti*-taulun sarake *timestamp* säilyttää vain tekstiä. Ohjelman toteutuksessa otettiin huomioon että sarakkeeseen tallennetaan aikaleima vain tarkasti muotoiltuna tekstinä, josta se tarvittaessa puretaan ohjelmallisesti taas päivämääriä käsitteleväksi tietotyyppiksi Java-kielessä.



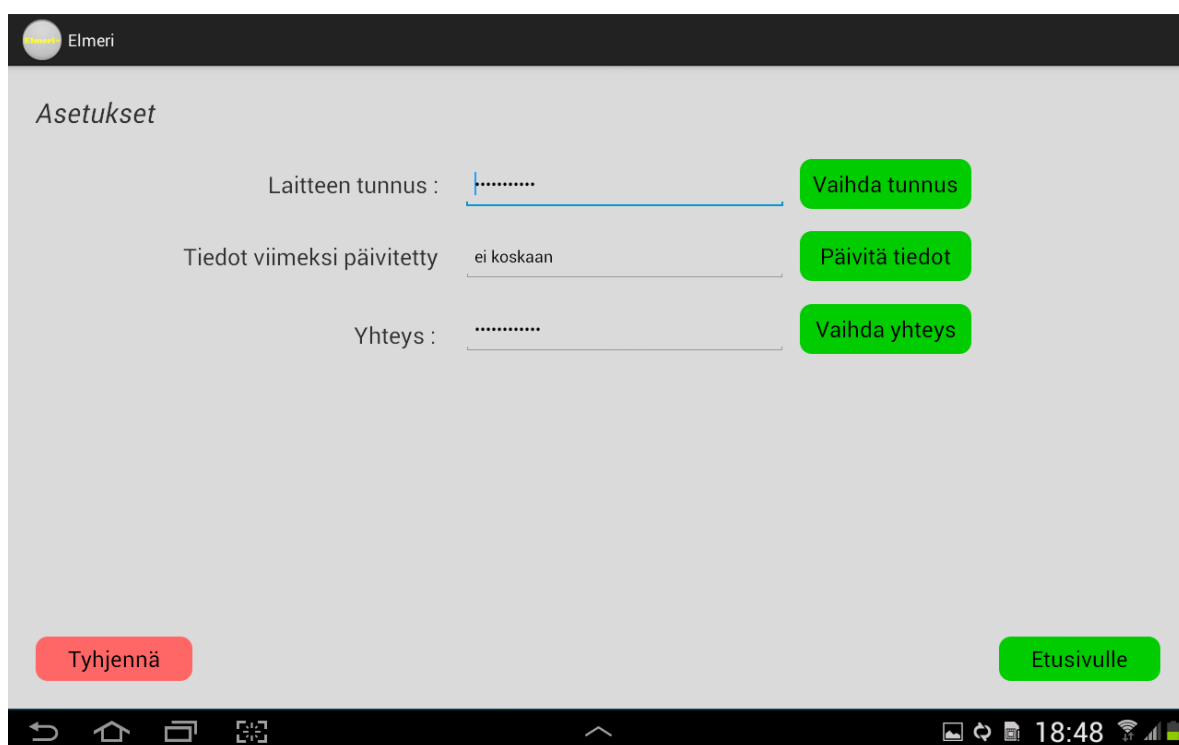
KUVIO 4. Android-sovelluksen sisäinen tietokanta.

Sovelluksen aloitussivulta on mahdollista navigoida tekemään uusia Elmeri+-mittauksia, tarkastelemaan palvelimelle lähettämättömiä mittaustuloksia, muuttamaan ohjelman toiminnan asetuksia, lukemaan käyttöohjeita tai sammuttamaan ohjelman. Sovellus asetettiin toimimaan pelkästään tabletin ollessa vaakatasossa, joten laitteen kääntäminen ei käänne kuvaa sovelluksessa.



KUVIO 5. Android-sovelluksen aloitussivu.

Asetukset-activityllä (Kuvio 6) voi asettaa salasanana käytettävän laitetunnuksen sekä palvelimen käyttämän yhteysosoitteen. *Päivitä tiedot*-napista ohjelma koostaa XML-muotoisen kyselyn uusien asetustietojen kysymistä varten ja pyrkii lähettämään HTTPS-yhteydellä kyselyn palvelimelle. Jos yhteys onnistuu ja palvelinsovellus hyväksyy kyselyn mukana lähetetyn salasanan, saapuu palvelimelta android-sovellukselle XML-muotoinen lista vastuualueista Lujabetonin tuotantolaitoksilla ja niihin liittyvät tunnukset sisäverkon tietokannassa. Lista parsitaan ohjelman sisäiseen SQLite-tietokantaan. Ohjelma ilmoittaa onnistuneesta tietojen päivityksestä tai virheilanteesta tulostaa virheilmoituksen.



KUVIO 6. Android-sovelluksen asetussivu.

Uusi mittaus-activityllä (Kuvio 7) syötetään tarvittavat tiedot uuden Elmeri+-mittauksen aloittamista varten. Ohjelma vaatii ennen uuden mittauksen aloittamista vähintään vastuualuenumeron, mitattavan alueen sekä mittauksen suorittajan. Muiden mittaukseen osallistuvien kirjaaminen on valinnais- ta. Vastuualuenumero-kentästä poistuttaessa ohjelma ilmoittaa käyttäjälle vastuualuenumeroa vastaavan vastuualueen nimen. Jos numeroa vastaava vastuualuetta ei löydy, ohjelma antaa virheil- moituksen eikä päästä etenemään mittauksen kirjaamisvaiheeseen *Aloita mittaus*-napin kautta.

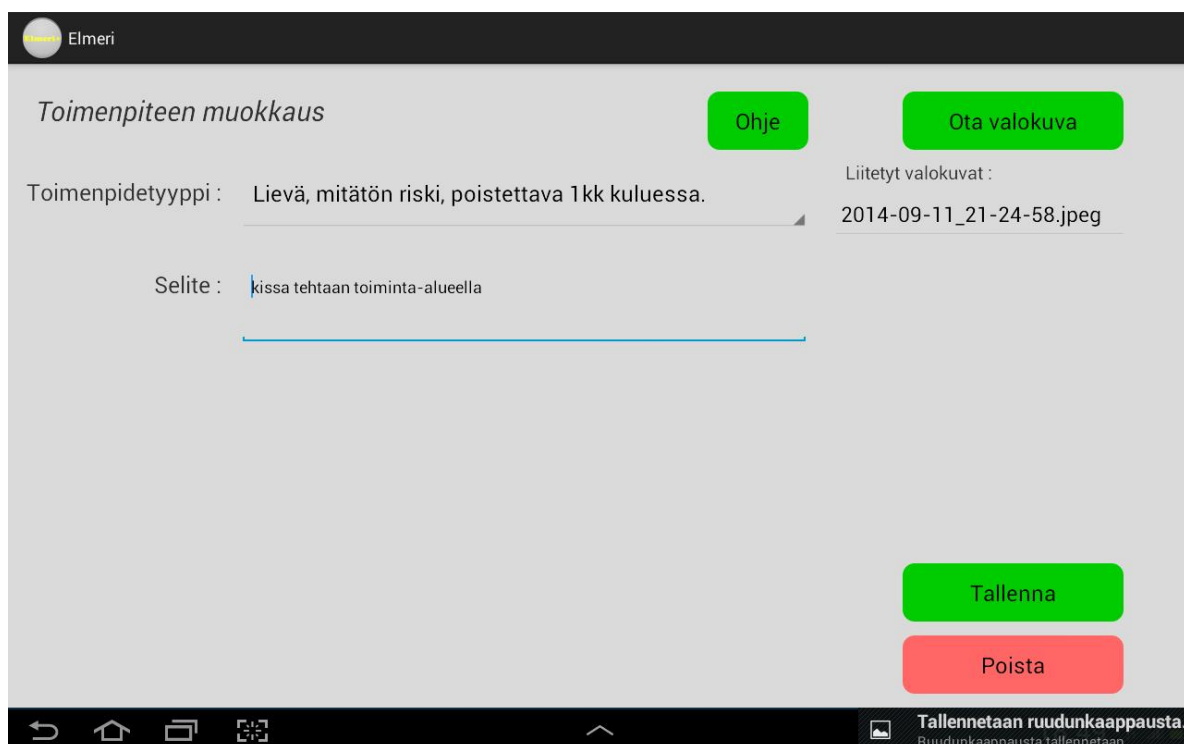
KUVIO 7. Uuden Elmeri+-mittauksen aloitus

Havaintojen kirjaukseen käytetyllä activityllä (Kuvio 8) voidaan kirjata *kunnossa* sekä *ei kunnossa*-tyyppisiä havaintoja osa-alueittain. Koska havaintoja on 15 eri osa-alueella ja käyttöliittymän näp- päimistä haluttiin isoja, on lista vertikaalisesti vetoliikkeellä vieritettävä. *Uusi toimenpide*-napista päästään kirjaamaan Elmeri+-mittaukseen liitettävä riskipaikkailmoitus/elmeri+-havainto (Kuvio 9). *Peruuta*-nappi poistaa meneillään olevan mittauksen kaikki kirjaukset, sekä toimenpiteet jotka mit- taukseen on ehditty siihen mennessä jo liittää. *Tallenna*-nappi varmistaa ensin käyttäjältä lopete- taanko mittauksen kirjaaminen.

Yhden positiivisen tai negatiivisen havainnon kirjatakseen käyttäjä painaa havainnoidun osa-alueen rivillä olevaa *kunnossa*- tai *ei kunnossa*-nappia jolloin napissa näkyvä havaintojen lukumäärä nousee yhdellä. Vahingossa suoritettun painalluksen voi peruuttaa painamalla kyseistä nappia pitkään, jolloin luku laskee yhdellä ja tabletin tukiessa värinätoimintoja, laite värähtää hetken.



KUVIO 8. Elmeri+-mittauksen havaintojen kirjaussivu



KUVIO 9. Elmeri+-mittaukseen liitettävän toimenpiteen kirjaus

Valmiiksi kirjattuja mittauksia on mahdollista selata ja lähettää *Lähetämättömät mittaukset*-sivulta (Kuvio 10). Yksittäistä mittausta pääsee tarkastelemaan kyseisellä rivillä sijaitsevalla *Tarkastele*-napilla jolloin mittaus aukaistaan *Mittauksen tarkastelu*-sivulle (Kuvio 11).

Mittauksia lähettäessä *lähetä mittaukset*-napilla sovellus rakentaa laitteen tietokantaan tallennetuista mittauksista, mittauksiin liittyvistä toimenpiteistä sekä toimenpiteisiin liittyvistä valokuvista XML-

muotoisen dokumentin, jota ohjelma pyrkii lähettämään HTTPS-yhteydellä *Asetukset*-ikkunassa määriteltyyn verkko-osoitteeseen. Jos palvelin lähettää takaisin merkinnän onnistuneesta lähetyksestä ja tallennuksesta, mittaukset, niihin liittyvät toimenpiteet ja toimenpiteisiin liittyvät valokuvat poistetaan laitteen tietokannasta ja laitteen muistista.



KUVIO 10. Lähtämättömät Elmeri+-mittaukset Android-sovelluksessa

Mittauksen tarkastelu-activityssa nähdään yhteenveto tallennetusta mutta vielä palvelimelle lähettämättömästä mittauksesta. Activity hakee ohjelman sisäisestä tietokannasta mittauksen tiedot, *kunnossa* ja *ei kunnossa* merkinnät sekä laskee niiden välisen suhteen ja ELMERI+-indeksin. Mittaukseen liitettyjä toimenpiteitä on mahdollista vielä mennä tarkastelemaan, muokkaamaan ja poistamaan mutta uusia toimenpiteitä ei voi enää mittaukseen lisätä.

Mittauksen tarkastelu

Poista mittaus Takaisin

Mittauksen tiedot :

Suorittaja : jniv
Mittausaika : 2014-09-11
Vastuualue : SJR Hormielementit
Alue : 8
Muut osallistujat :

Mittaukseen liitetyt toimenpiteet :

kissa tehtaan toimin...

Havaintokohde	Osaaindeksi	Kunnossa	Ei kunnossa
Riskinotto, suojaimet, vaatetus	100 %	3	0
Fyysinen kuormitus	100 %	2	0
Työpisteen ja välineiden ergonomisuus	33 %	1	2
Koneiden kunto, ja suojalaitteet	50 %	2	2
Koneiden Hallintalaitteet ja merkinnät	0 %	0	1
Kulkuteiden / lattioiden rakenne, putoamissuojaus	Ei arvoa	0	0
Poistumistiet	67 %	2	1
Kulkuteiden ja lattioiden järjestys	67 %	4	2
Pöydät, päällyset ja hyllyt	100 %	2	0
Jäteastiat	100 %	1	0
Tupakointipaikka	33 %	1	2
Melu	67 %	2	1
Valaistus	33 %	3	6
Lämpöolosuhteet	100 %	3	0
Ilman puhtaus ja käsiteltävät aineet	50 %	4	4
ELMERI+ indeksi	59 %	30	21

KUVIO 11. Suoritetun mittauksen tarkastelu Android-sovelluksessa.

6.2 TJR-sovellus

Toimintajärjestelmän kautta käytettävällä sovelluksella (Kuvio 6) on mahdollista etsiä mittauksia hakukriteereillä *tuotantoyksikkö*, *vastuualue*, *mittausalue*, *suorittaja* sekä *mittauspäivämäärä*. Haetuista mittauksista kerrotaan suorittaja, mittauspäivä, vastuualue, jolla mittaus on suoritettu, mittaukseen liitettyjen riskipaikkailmoitusten lukumäärä sekä mittausarvoista laskettu ELMERI+-indeksi.

Tuplaklikkaamalla mittauksen riviä, aukeaa kyselyikkuna josta voi joko avata tulostettavan version mittauksen tuloksista ja tiedoista, tai tarkastella kyseisen mittauksen aikana tehtyjä riskipaikkailmoituksia. Riskipaikkailmoitusten tarkastelua varten avautuu jo pitkään konsernissa käytössä ollut TJR-riskipaikkasovellus, eikä siihen tehty muutoksia tässä projektissa.

ELMERI+

Hae mittaukset Tyhjennä kriteerit Uusi mittaus Luo raportti Leikepöydälle Sulje

Vastuualueet SJR teräsbetoni SJR tasoristeyselementit

Yksiköt BTY ELE VBY

Mittausalunumero

Mittauksen suorittaja

Mittausaikaväli Mistä 1.1.2014 Mihin 30.6.2014

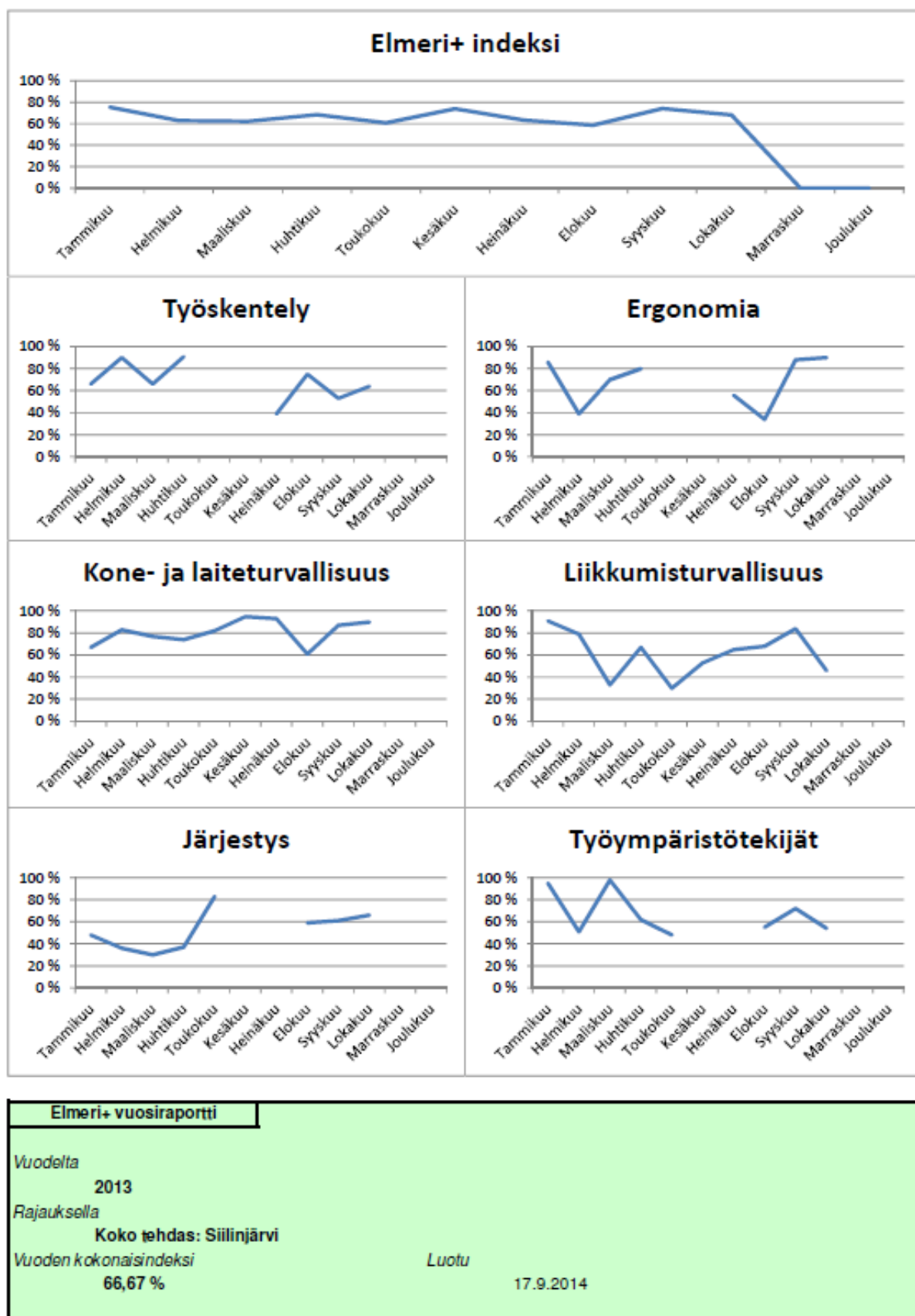
Vuosiraportti Yksikkö * Tehdas KUO Vastuualue KUO valmisbetoni Vuosi 1994

Id	Suorittaja	Aikaleima	Vastuualue	Alue	Muut osallistujat	Toimenpiteet	Indeksi
2	Jyrki Katainen	05.01.2014	SJR tasoristeyselementit	3		0	62.50
32	Hannu Salama	20.03.2014	SJR tasoristeyselementit	2	Juhannustanssit	0	64.29

KUVIO 12. Elmeri+ TJR-sovellus

Kaikista haetuista mittauksista on mahdollista luoda koontiraportti, joka käyttää samaa pohjaa kuin kuvion 1 ELMERI+-raporttipohja. Koontiraporttiin lasketaan yhteen kaikkien hakukriteerit täyttävien mittausten arvot. Vuosiraportti-osiosta voi tuottaa valmiin taulukkolaskentapohjan täyttävän taulukon, johon on laskettu kaikkien mittausten kuukausittaiset keskiarvot joko tuotantoyksiköittäin, tehtaittain tai tehtaiden vastuualueittain tietyltä vuodelta (Kuva 7). Tiedot viedään taulukkolaskentapohjaan, joka hakee tietokannasta ensin valmiiden pohjien varaston sisäverkko-osoitteen, minkä jälkeen noutaa sieltä tarvittavan pohjan raportille ja vie pohjaan tietokannasta kootut mittaustiedot.

Ohjelma hakee aina uudet tulosrivit välittömästi hakuehtojen muuttuessa, mikä aiheuttaa jatkuvia SQL-kyselyitä tietokantaan. Sovelluksen toteutuksessa ei kuitenkaan ollut tarpeen ottaa huomioon viiveitä tietokantahakujen aikana sillä ohjelma toimii ainoastaan yhtiön sisäverkossa, missä yhteydet ovat verkon normaalissa toiminnassa riittävän nopeita ihmissilmälle.



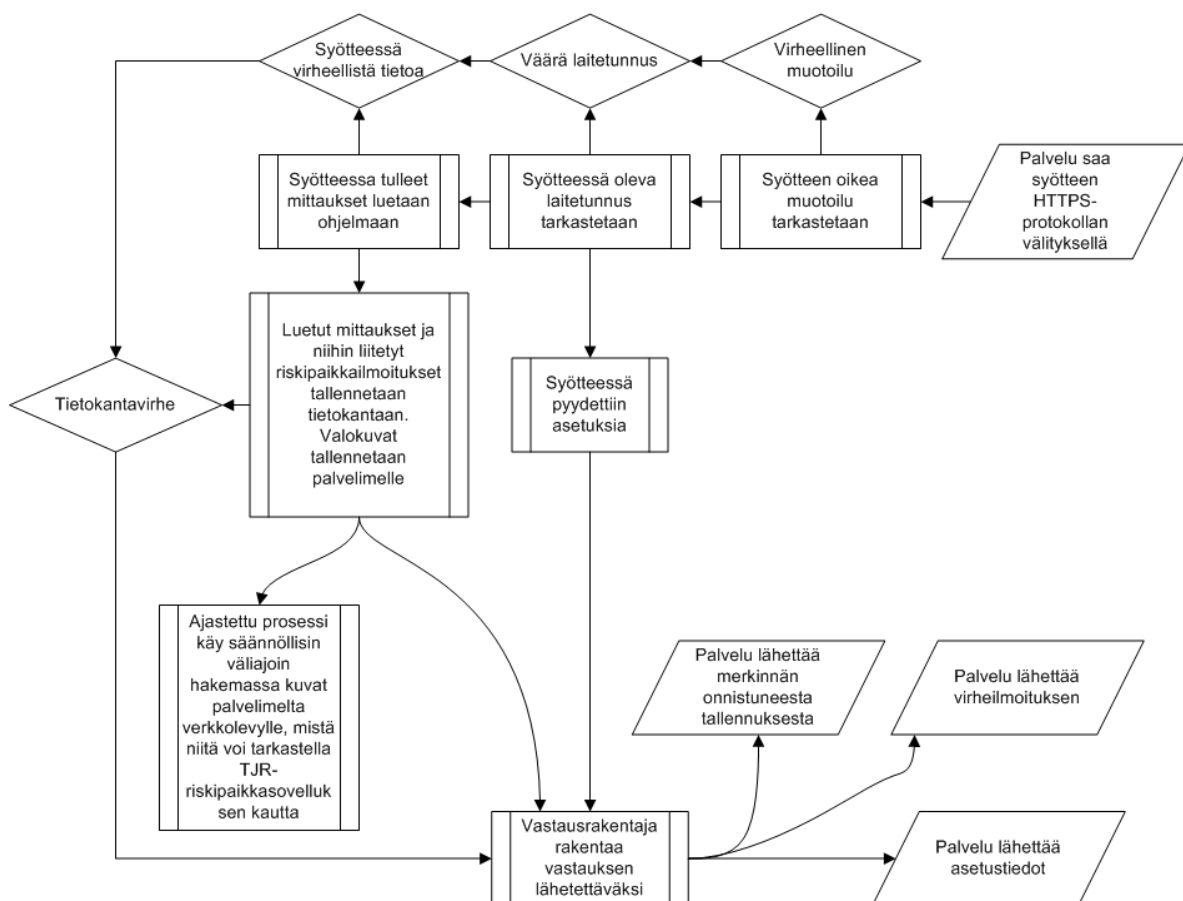
KUVIO 13. Vuosittainen raportti Elmeri+-mittauksista

6.3 ASP.NET-Palvelinsovellus

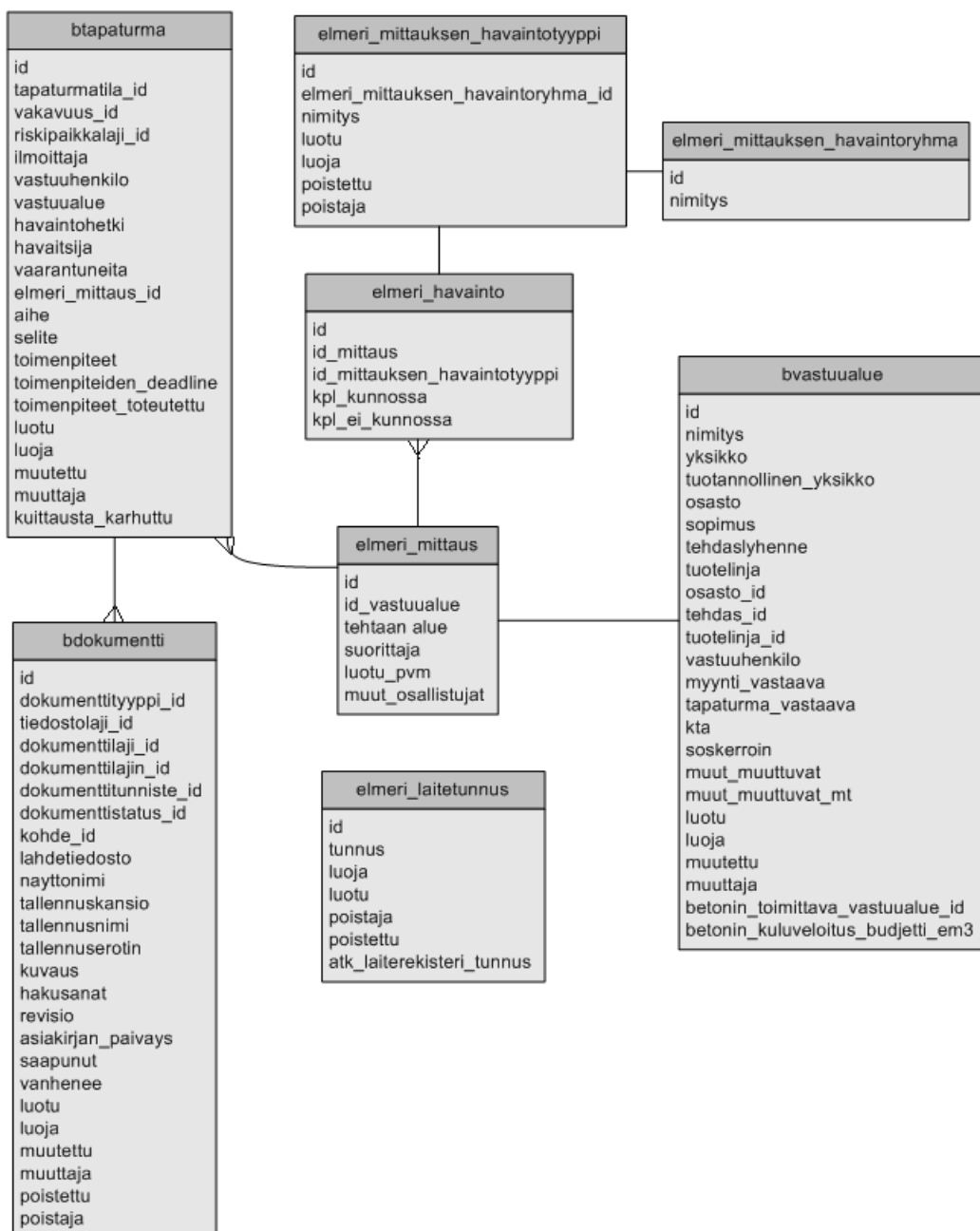
Android-sovellukselta saapuva syöte tulee palvelimelle salatun HTTPS-yhteysprotokollan välityksellä. Mobiililaitteen päässä syöte on koodattu XML-muotoon ja palvelin tarkastaa ensiksi, onko syöte XML-muotoinen. Tämän jälkeen tarkastetaan syötteen mukana tuleva laitetunnus, jonka mobiilisovelluksen käyttäjä on asettanut mobiilisovelluksen *Asetukset*-ikkunasta laitteelleen. Jos laitetunnus vastaa yhtiön tietokannasta löytyvää laitetunnusta, syötteen käsittelyä jatketaan. Laitetunnuksen tarkoituksena on toimia turvallisuusmekanismina tilanteissa, jossa laite katoaa tai se varastetaan. Tällöin laitetta vastaavan laitetunnuksen voi poistaa tietokannasta, eikä ohjelmalla voi enää lähettää mittauksia.

Tämän jälkeen tarkastetaan lähetetystä kyselystä, haluaako mobiililaitteen käyttäjä kyselyllään päivittää laitteen tietoja. Jos tietoja päivitetään, niin tietokannasta koostetaan lista XML-muotoon päivittävistä tiedoista ja lähetetään takaisin tallennettavaksi mobiililaitteen tietokantaan.

Jos mobiililaitteelta ollaan tallentamassa mittauksia, mittaukset luetaan ensin syötteestä olioiksi. Olioiksi lukemisen yhteydessä selviää että tietojen muoto on oikea ja syöte ei ole virheellinen. Tämän jälkeen mittaukset tallennetaan tietokantaan. Mittaustietojen kantaan tallentamisen seurauksena palvelin rakentaa XML-muotoisen vastauksen joka ilmoittaa mobiililaitteelle ilmoittaakseen että laitteella olevat tiedot voidaan poistaa. Mahdollisissa virhetilanteissa palvelin lähettää takaisin mobiililaitteelle virheilmoituksen joko virheestä tietokantayhteydessä tai virheellisestä syötteestä.



KUVIO 14. Palvelimen toiminnan vuokaavio.



KUVIO 15. Mittausten sekä toimenpiteiden rakenne sisäverkon tietokannassa.

7 TESTAUS

Palvelinsovellusta varten pystytettiin virtualisoitu Windows Server 2012-palvelinkone ja siihen palvelinsovelluksen käyttämä IIS sekä sen käyttämä ASP.NET-rajapinta. Työpöytäsovellus asennettiin yhtiön sisäverkossa toimivaan TJR-järjestelmään suoraan tuotantokäyttöä varten. Järjestelmää testattiin tekemällä 10-tuumaisilla Lenovo- sekä Samsung Galaxy S Tab-tableteilla testimittauksia. Testa- tessa laitteilla tehtiin vaihtelevia määriä testimittauksia ja riskipaikkailmoituksia, joihin liitettiin vaihtelevia määriä valokuvia. Mittausten siirtoa ja tallentumista tietokantaan seurattiin ja tallennettujen tietojen oikeellisuus tarkistettiin.

8 YHTEENVETO

Tavoitteena oli suunnitella ja toteuttaa toimiva it-palvelukokonaisuus Lujabetoni Oy:lle mallina vain palvelun kynällä ja paperilla käytettävä esikuva. Vaikka minulla oli aiempaa kokemusta ohjelmoinnista useilla eri kielillä ja tiesin teoriassa järjestelmän toteutusperiaatteet, ei minulla ollut useiden keskenään kommunikoivien useampia ohjelmia käsittävien kokonaisuuksien toteuttamisesta kokemusta. Tämä asetti minulle henkilökohtaisia haasteita, sillä jouduin opettelemaan suunnittelumalleja ja ohjelmistoarkkitehtuurin perusteita itsenäisesti.

Toimeksianto oli siinäkin mielessä haastava, että toimeksiantajan organisaatiossa ei ollut kukaan aiempaa kokemusta mobiilijärjestelmien toteuttamisesta, jolloin kukaan ei voinut neuvoa minua mobiilisovelluksen toteuttamisessa. Palvelinsovelluksen sekä TJR-sovelluksen toteuttamisessa sain tosin apua ja kannustusta työtovereiltani, mistä olen erittäin kiitollinen.

Projekti opetti minulle paljon ohjelmoinnin hyväksi havaituista käytännöistä, ohjelmistojen suunnittelusta sekä itsenäisestä työskentelystä alalla. Ymmärryksen varsinkin Javan ja Androidin sisäisistä toimintamekanismeista kehittyi ja uskon siitä olevan hyötyä tulevaisuudessa vähintäänkin välillisesti. Myös palaute toimeksiantajalta oli kannustavaa sekä rakentavaa niin projektin edetessä kuin sen valmistuessa.

Työ valmistui ennalta sovittuun päivämäärään mennessä. Jatkokehitystä ajatellen palvelun kehittämiskohteet tulevaisuudessa ovat toimintojen tehokkuuden ja ohjelmien vakauden päivittämisessä paremmalle tasolle.

Jos toteuttaisin saman projektin nyt projektin aikana opitulla tietotaidolla, toteuttaisin ohjelmistojen väliset rajapinnat sekä olioiden relaatiot loogisemmin ja selkeämmin. Tietokantayhteydet varsinkin Android-sovelluksessa olisi pitänyt myös suunnitella tarpeeksi pitkälle etukäteen muokattavuuden ja ylläpidettävyyden kannalta.

LÄHTEET

ECLIPSE FOUNDATION 2014 [verkkójulkaisu] [Viitattu 2014-10-24] Saatavissa: <http://eclipse.org/org/>

GOOGLE JA OPEN HANDSET ALLIANCE 2014a [verkkójulkaisu] [Viitattu 2014-10-05] Saatavissa: <https://developer.android.com/reference/android/app/Activity.html>

GOOGLE JA OPEN HANDSET ALLIANCE 2014b [verkkójulkaisu] [Viitattu 2014-11-18] Saatavissa: <https://developer.android.com/legal.html>

IETF 1996. SSL-protokollan käyttö TCP-protokollan päällä [verkkójulkaisu] [Viitattu 2014-12-01] Saatavissa: <https://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00>

KOLEHMAINEN, Kauko. 2006. JAVA-ohjelmointi – Algoritmit ja mallit. Jyväskylä: Gummeruksen Kirjapaino Oy.

LAHTONEN, Tommi 2003 SQL. Jyväskylä: Docendo.

LUJABETONI OY [verkkójulkaisu] [Viitattu 2014-12-01] Saatavissa: <http://www.lujabetoni.fi/>

MICROSOFT 2014a Internet Information Services. [verkkójulkaisu] [Viitattu 2014-12-01] Saatavissa: <http://msdn.microsoft.com/en-us/library/ee532514%28v=vs.90%29.aspx>

MICROSOFT 2014b SQL Server Management Studio. [verkkójulkaisu] [Viitattu 2014-11-23] Saatavissa: <http://msdn.microsoft.com/en-us/library/hh213248.aspx>

MICROSOFT 2014c Windows Server 2012 R2. [verkkójulkaisu] [Viitattu 2014-12-02] Saatavissa: <http://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2/default.aspx>

MOGHADAMPOUR, Ghodrat. 2011. C# Windows- ja tietokantaohjelmointi. Jyväskylä: Docendo.

WALKAMA, Pekka ja LAAKKONEN, Aapo 2004. XML-Skeema. Helsinki: Edita Prima Oy.