

Nino Silvennoinen

Modernin öljyterminaalin hallintajärjestelmän käyttöä tukeva mobiililaite

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Automaatiotekniikka

Insinööriytyö

3.5.2016

| | |
|--|---|
| Tekijä(t) Otsikko Sivumäärä Aika | Nino Silvennoinen Modernin öljyterminaalien hallintajärjestelmän käyttöä tukeva mobiililaitte 37 sivua + 1 liitettä 3.5.2016 |
| Tutkinto | Insinööri (AMK) |
| Koulutusohjelma | Automaatiotekniikka |
| Suuntautumisvaihtoehto | |
| Ohjaaja(t) | Terminaalipäällikkö Lars Munck Lehtori Timo Tuominen |
| <p>Insinööriyön periaatteena oli koota öljyterminaalien tietojärjestelmä- ja automaatiolaitteiston halutut toiminnot ja ominaisuudet yhteen sovellukseen. Tavoitteena oli luoda esimerkkisovellus etäkäytöstä, sekä pohja etähallinnan kehityssuunnitelmalle. Työssä keskityttiin hallintajärjestelmän ja muiden automaatioon liittymättömien laitteistojen tärkeimpien käyttöaluiden sovittamiseen kootusti mobiililaitteelle. Työn aikana pohdittiin myös mobiililaitteen vaikutusta työskentelyyn ja turvallisuuteen. Työ tehtiin Neste Inkoon terminaalille.</p> <p>Työn alussa määritettiin mallisovellukselta toivotut ominaisuudet. Automaation rajattujen toimintojen etäkäytön lisäksi sovellukseen tahdottiin yhdistää myös huolto- ja kunnossapitojärjestelmän ominaisuuksia.</p> <p>Tuloksena saatiin luotua terminaalille Android-pohjaisen mobiilisovelluksen prototyyppi, jossa yhdistyi terminaalien hallintajärjestelmän tarvittavat ominaisuudet. Sovellukseen saatiin myös kehitettyjä toimintoja, jotka päivittävät vanhoja jokapäiväisen operoinnin menetelmiä nykypäivään.</p> | |
| Avainsanat | Mobiililaitte, Android, etäkäyttö |

| | |
|--|--|
| Author(s) Title | Nino Silvennoinen Supportive Mobile Device to control system of Modern Oil Terminal |
| Number of Pages Date | 37 pages + 1 appendices 3 May 2016 |
| Degree | Bachelor of Engineering |
| Degree Programme | Automation Engineering |
| Specialisation option | |
| Instructor(s) | Lars Munck, Head of Terminal Timo Tuominen, Senior Lecturer |
| <p>The main principle of the Engineering thesis was to assemble features and functions of an oil terminal's information and automation system into one application. The work was commissioned by Neste Inkoo terminal. The aim was to create a sample of a mobile application and a base for further development. Themes like industrial safety and operational functionality were considered throughout the whole development process. The application's impact on those subjects was considered and studied beforehand and also at the final steps of the study.</p> <p>Requirements and the configuration of the work were determined after the initial stages. The aim was to merge certain limited functions of the automation's SCADA-application and sections of the maintenance management systems as well as the access control system.</p> <p>As a result, a prototype of an Android-based mobile application was made, in which the required features are combined. Functions that upgrade the daily operating routines to more safe and modern ways were also included in the application.</p> | |
| Keywords | Mobile device, Android, remote access |

Sisällys

Lyhenteet

| | | |
|-------|--|----|
| 1 | Johdanto | 1 |
| 2 | Inkoon terminaali | 1 |
| 2.1 | Nykytila | 1 |
| 2.2 | Terminaalin automaatiojärjestelmän kokoonpano | 2 |
| 3 | Suunnittelu ja vaatimusmäärittely | 3 |
| 3.1 | Sähköinen päiväkirja | 3 |
| 3.2 | Laitteiden tilan tarkastus ja luku mobiililaitteella | 4 |
| 3.3 | Laitteiden tunnistaminen | 6 |
| 3.4 | Automaation etäkäyttö | 7 |
| 3.4.1 | Etäkäyttö | 7 |
| 3.4.2 | Etäkäyttö teollisuudessa | 8 |
| 3.4.3 | Etäkäytön tietoturva | 8 |
| 3.5 | Yhteys mobiililaitteen ja automaation välillä | 8 |
| 4 | Tekniset vaatimukset ja laitteen valinta | 10 |
| 4.1 | ATEX-soveltuvuus | 10 |
| 4.2 | Ecom Tab-Ex® Series | 12 |
| 4.3 | Tunnisteiden lukeminen | 14 |
| 5 | Ohjelmointi | 14 |
| 5.1 | Android Studio | 14 |
| 5.2 | Viivakoodien luku | 15 |
| 5.3 | Yhteys etätietokoneeseen | 16 |
| 5.3.1 | Yhteyden muodostaminen | 16 |
| 5.3.2 | Levyjaon yhteysongelmat | 17 |
| 5.4 | Käyttäjätietokanta | 18 |
| 5.4.1 | Toiminta | 18 |
| 5.4.2 | Kehitysideat | 19 |
| 5.5 | Päiväkirjasovellus | 20 |
| 5.5.1 | Merkinnän kirjoittaminen | 20 |

| | | |
|-------|--|----|
| 5.5.2 | Päiväkirjan lukeminen | 21 |
| 5.5.3 | Kehitysideat | 22 |
| 5.6 | Automaation toiminnot | 23 |
| 5.7 | Laitekohtaisten tunnisteiden lukeminen | 23 |
| 5.7.1 | Laitteen tiedot | 24 |
| 5.7.2 | Laitteiden tarkastus | 24 |
| 5.7.3 | Laitteiden linjaus | 26 |
| 5.7.4 | Kehitysideat | 29 |
| 6 | Päätelmät | 29 |
| | Lähteet | 31 |
| | Liitteet | |
| | Liite 1. Android -koodiesimerkki | |

Lyhenteet

| | |
|----------|--|
| ATEX | <i>Atmosphères Explosibles</i> . Räjähdysvaaralliset tilat. |
| CIFS | <i>Common Internet File System</i> . Datajako-protokolla. |
| CSV | <i>Comma Separated Value</i> . Tiedostomuoto, jossa tieto on eroteltu pilkuin tai puolipistein. |
| GUI | <i>Graphic User Interface</i> . Graafinen käyttöliittymä. |
| Java | Sun Microsystemsin luoma olio-ohjelmointikieli. |
| NFC | <i>Near Field Communication</i> . RFID-teknoologiaan pohjautuva tunnistus- ja |
| QR-koodi | <i>Quick Response-koodi</i> . 2D- eli kaksiulotteinen viivakoodi. |
| RFID | <i>Radio Frequency Identification</i> . Radiotaajuuksilla toimiva etätunnistus- ja kommunikointimuoto. |
| SCADA | <i>Supervisory Control And Data Acquisition</i> . Valvomosovellus. |
| SMB | <i>Server Message Block</i> . Datajako-protokolla lähiverkoille. |
| XML | <i>Extensible Markup Language</i> . Tiedonsiirtotiedostotyyppi. |
| VPN | <i>Virtual Private Network</i> . Virtuaalinen näennäisverkko. |

1 Johdanto

Neste Inkoon Terminaalin laitteisto on insinööriyön tekohetkellä hiljattain modernisoitu. Terminaalin valvomorakennukseen on keskitetty koko alueen automaation hallinta. Operaattorit työskentelevät paljon kentällä, joten valvomossa ei aina ehdi viettää enempää aikaa kuin operointi vaatii. Automaatiolaitteiston käyttäminen kentällä kuitenkin vaatii operaattorin läsnäoloa valvomossa.

Myös terminaalin tiedonhallintajärjestelmä on uusittu, eli pyritty poistamaan käytöstä kaikki paperinen aineisto ja siirtyä sähköiseen tietokantaan. Työlupien laatiminen, varaston sekä laitetietokannan hallinta suoritetaan nykyään sähköisesti. Kentällä täytettävät operoinnin kannalta oleelliset lomakkeet ja tarkistuslistat olivat kuitenkin vielä paperisessa muodossa, jotka jouduttiin myöhemmin arkistoinnin kannalta syöttämään sähköiseksi aineistoksi.

Insinööriyön tavoitteena oli suunnitella ja kehittää terminaalien käyttöön mobiilikäytön esimerkkisovellus, jolla saataisiin tehostettua operaattorien kentällä työskentelyä mahdollistamalla automaatio- ja tietojenhallintajärjestelmän tarvittavien osa-alueiden etäkäyttö.

2 Inkoon terminaali

Neste Inkoon terminaali on polttoaineen varastointiin ja jakeluun keskittynyt laitos. Alueella suoritetaan laiva- ja autolastauksia sekä -purkuja.

2.1 Nykytila

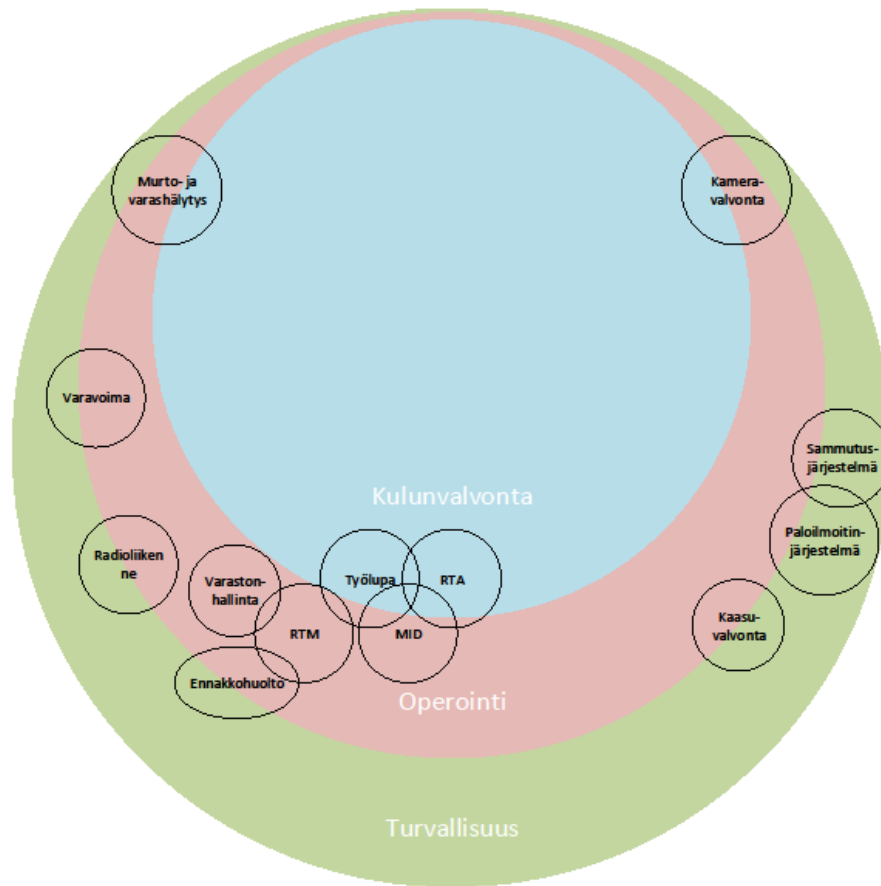
Modernisoinnin tuloksena terminaaliin on syntynyt aikaisempaa enemmän operointikohteita, mikä lisää operaattorien päivittäisiä rutiinitehtäviä huomattavasti. Automaation ohjaus on keskitetty täysin terminaalien keskusvalvomoon, mikä taas yksinkertaistaa operointia ja terminaalien laitteiston hallintaa. Keskusvalvomossa on automaation SCADA-päätteet, joista tapahtuu koko terminaalien automaation ja siihen liitettyjen erillisten järjestelmien ohjaus. Tietoa saadaan kerättyä enemmän eri prosesseista, laitteista ja tapahtumista, jolloin sitä voidaan hyödyntää tehokkaasti toiminnallisuuden ylläpitämiseksi.

Myös turvallisuuteen liittyvät laitteet ovat kehittyneet ja uusia järjestelmiä on tullut käyttöön, kuten erillinen automatisoitu sammutusjärjestelmä. Terminaalilla on käytössä myös sähköinen kulunvalvonta- ja työlupaohjelmisto sekä huolto- ja laiterekisteri, jotka molemmat ovat yhteydessä automaatioon. Tämä mahdollistaa automaation monipuolisen ja tehokkaan käytön operointia tukevana elementtinä.

Uudistetusta automaatiojärjestelmästä huolimatta terminaalilla suoritetaan vielä kenttäoperointia manuaalisesti. Operaattorien päivärutiiniin kuuluu kentällä suoritettavat erilliset alueen ja laitteiston valvonta-, huolto- ja tarkistuskierrokset. Operaattorin suorittaessa kenttäoperointia tulee eteen usein tilanteita, joita ei kentältä käsin voi toteuttaa, esimerkiksi terminaalin päiväkirjamerkinnot, valvomopäätteeltä tapahtuva automaatiolaitteiston operointi sekä kulunvalvontaan liittyvä porttien avaus. Terminaalin toiveena oli kehittää automaation käyttötapoja ja saada käyttöön mobiililaitte, jolla voitaisiin vaivattomasti suorittaa etäältä yleisimpiä operointia helpottavia toimintoja, kuten edellä mainittuja tehtäviä.

2.2 Terminaalin automaatiojärjestelmän kokoonpano

Automaation keskuslogiikkana on Siemens S7-300. Logiikan tulot ja lähdöt on hajautettu alakeskuksiin Simatic ET-200-aseilla. Suurin osa prosessilaitteista, kuten massavirtamittarit, ovat profibus-väylään liitettynä. Käyttöliittymä on rakennettu Windows-ympäristöön Wonderwaren InTouch-ohjelmistolla. Koko automaatio sekä valvomon PC:t on liitetty erilliseen automaation lähiverkkoon, jossa terminaalien eri rajapinnat yhdistyvät. (ks. Kuva 1.)



Kuva 1. Terminaalien järjestelmien jakautuminen eri osa-alueille

3 Suunnittelu ja vaatimusmäärittely

Vaatimusmäärittelyssä paneuduttiin erityisesti terminaalien operointia helpottaviin ratkaisuihin. Terminaalien henkilökunnan toiveita olivat sähköinen päiväkirja, laitteiden linjaus ja tarkistus sähköisesti sekä automaatiovalvomon etäkäyttö. Esille tulleet aiheet päätettiin lopulta yhdistää yhteen työhön, jolloin syntyi idea tablettitietokoneesta, missä olisi kaikki toivotut ominaisuudet yhdessä applikaatiossa.

3.1 Sähköinen päiväkirja

Terminaalilla on käytössä alueella päiväkirjoja, joihin merkitään kaikki kohteessa tapahtuneet poikkeamat ja tehdyt toimenpiteet. Terminaalien eri alueille on omat kirjansa, kun-

kin alueen valvomossa. Päiväkirjat on jaettu terminaalin pääoperointisijaintien sekä merkintöjen selkeyttämiseksi ja erottamiseksi toisistaan. Päiväkirjan täyttö ja tapahtumien merkintä kiireessä on hankalaa, mutta terminaalin tapahtumaseurannan ja turvallisuuden kannalta ehdotonta.

Tablettitietokoneella täytettävä ja luettava lokikirja helpottaisi operaattorien kenttätyökentelyä, sillä he voisivat sijainnista riippumatta kirjata tapahtuman ylös vaivattomasti. Kirjaukset tallentuisivat levyjaolle yhteen päiväkirjatiedostoon, luoden samalla varmuuskopion merkinnästä. Itse kirjaukseen operaattori merkitsisi vain tapahtuman ajankohdan, sijainnin sekä sisällön eli selostuksen, mitä tapahtui. Kirjauksen aikaleima sekä applikaatioon kirjautuneen käyttäjän tiedot lisättäisiin automaattisesti merkinnän tallennusten yhteydessä. Myös kuvien lisäys merkinnän yhteyteen olisi mahdollista. Mobiililaitteelta pystyisi kirjauksien lisäksi lukemaan päiväkirjaa. Merkinnät olisivat helposti saatavilla haku- ja suodatustoimintojen avulla. Kirjauksia pystyisi hakemaan ajan, sijainnin, tapahtuman ja kirjaajan perusteella. Jotta päiväkirjan ominaisuudet tulisi hyödynnettyä mahdollisimman laajasti, olisi laitetunnuksien tai muiden tunnisteiden lisääminen merkintöihin hyödyllistä. Jottei päiväkirjan luku ja kirjoitus rajoittuisi pelkästään yhteen mobiililaitteeseen, tulisi myös alueen valvomopäätteisiin luoda vastaavat sovellukset.

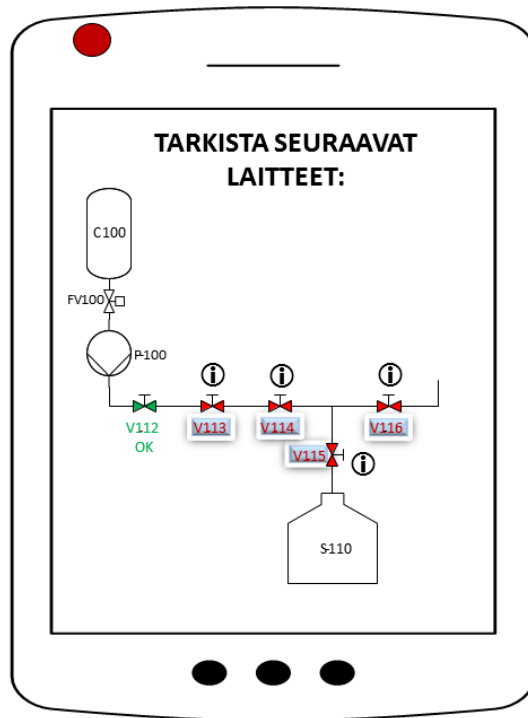
3.2 Laitteiden tilan tarkastus ja luku mobiililaitteella

Laitteiden tarkastus ennen käyttöä on operaattorin rutiinia. Ennen toimenpidettä tarkistetaan laitteiden toimivuus ja vaihdetaan tarvittaessa oikea tila toimenpiteen aloitushetkelle. Tällä varmistetaan, ettei toimenpiteestä aiheudu vaaraa ulkopuolisille sekä ympäröiville laitteille. Tarkistuksessa käytetään dokumenttia, jossa on listattuna kaikki toimenpiteen tarkistuskohteet, niiden tilat sekä muut laitekohtaiset huomiot. Valmis, kuitattu tarkistuslista arkistoidaan tai liitetään liittyvän urakan dokumentteihin. Tarkistuslistojen läpikäyminen ennen laitteiden käynnistämistä on käyttöturvallisuuden kannalta äärimmäisen tärkeää. Listojen huolellisella läpikäymisellä voidaan turvata laitteiden oikea käyttö ja laskea huomattavasti tapaturman riskiä.

Paperinen tarkistuslista on usein epäkäytännöllinen kentällä, varsinkin kosteissa tai likaisissa olosuhteissa. Myös listan arkistointi sen likaannuttua, kastuttua tai vaikka revettyä tuottaa päänvaivaa. Mobiililaitteen avulla suoritettava laitekohtainen tarkistus hel-

pottaisi listojen arkistointia ja säilyvyyttä. Se myös nopeuttaisi operointia ja poistaisi virheen mahdollisuutta. Tarkistuslistat olisi myös mahdollista liittää automaatioon ja sitä kautta lisätä laitoksen turvallisuutta.

Esimerkkinä laitteiden tarkistuksen vaativasta toimenpiteestä voisi olla polttoaineen siirto säiliöstä A, säiliöön B. Putkilinjan A—B varrella on kymmenen venttiiliä, joista viisi on automaatioon liitetty. Automaatioon liitetyt venttiilit pystytään käyttöliittymän kautta avaamaan ja sulkemaan, sekä näkemään kunkin venttiilin tilatiedot. Loput viisi venttiiliä ovat käsiventtiilejä, eli niiden operointi tapahtuu kentällä. Operaattori avaa mobiililaitteelta kyseisen tarkistuslistan, ja lista näyttää ensimmäisen laitteen, tässä tapauksessa venttiilin, joka pitää tarkistaa. Tarkistustoimenpiteessä katsotaan laitteen yleinen kunto, tarkistetaan laitteen tila sekä muutetaan tila toimenpiteeseen kuuluvaksi. Tarkistettuaan venttiilin operaattori kuittaa laitteen hyväksytyksi ja siirtyy seuraavalle laitteelle. Jos operaattorille on epäselvää laitteen sijainiti tai esimerkiksi missä asennossa venttiilin tulee olla, voi hän katsoa laitteen tarkemmat tiedot Info-painikkeesta. Info-painikkeen takana näkyy laitteen toimintakuvaus ja kuva, missä asennossa venttiilin tulee olla. Mikäli laitteen kunnossa havaitaan puutteita, tulee ne korjata ennen toimenpiteen aloittamista. Jos puute on kuitenkin vähäinen ja siitä ei koidu vaaraa tai haittaa, voi listaan merkitä kommentin laitteelle. Kun kaikki laitteet on käyty läpi ja tarkistettu, operaattori merkitsee listan käydyksi, jolloin laite tallentaa listan arkistoon ja merkitsee operaattorin kuittauksen ja aikaleiman automaattisesti. Lisäksi jos mobiililaitte saataisiin keskustelemaan automaation kanssa, olisi listan tallennuksen yhteydessä mahdollista antaa lupa automaatiolle aloittaa kyseinen toimenpide. (ks. Kuva 2.)



Kuva 2. Linjauksen luonnos - esimerkkiruutu linjauksen toiminnasta mobiililaitteella

3.3 Laitteiden tunnistaminen

Laitteiden tarkistamisessa oli alustavien suunnitelmien mukaan ajatuksena hyödyntää viivakoodi- tai RFID-tunnistusta. Jokaisella terminaalilaitteella on yksilöllinen positio-tunnus, jolloin kullekin laitteelle voisi helposti luoda oman viivakoodi- tai RFID-tunnuksen. Tällöin jokaisen laitteen voisi kuitata vaivatta mobiililaitteen viivakoodi- tai RFID-tunnistuksen avulla.

Laittekohtaisia tunnisteita voitaisiin myös käyttää laitetietojen hakemiseen tietokannasta. Kentällä voisi mobiililaitteen kanssa nopeasti tarkistaa laitteen tarkemmat tiedot ja varaosien varastosaldon, kuitata laite huolletuksi tai asettaa laitteelle huoltopyynnön. Myös laitteen liittäminen aikaisemmin mainittuun mobiiliin päiväkirjamerkintään tunnisteen avulla olisi mahdollista.

3.4 Automaation etäkäyttö

Terminaalin automaatiojärjestelmä kattaa koko alueen ohjauksen ja valvonnan. Käyttöliittymä on jaettu kahteen eri valvomoon, joissa molemmissa on identtiset näkymät. Pääosin automaation kaikki operointi tapahtuu valvomoista, mutta joidenkin toimintojen mahdollisuus saada käyttöön myös kentällä helpottaisi operaattorien työskentelyä huomattavasti. Näistä esimerkkeinä voisi olla valaistuksen ohjaus, porttien ohjaus sekä hälytystiedot. Päivystävän operaattorin olisi tällöin helppo suorittaa kulunvalvontaa kentältä käsin, vaikkapa ajoneuvosta. Tällä hetkellä automaatio lähettää vain kriittisimmät hälytystiedot päivystyspuhelimeen, mutta kaikkien hälytysten saaminen mobiilisti kentällä parantaisi reagointi-aikaa virheisiin ja madaltaisi poikkeus- tai vaaratilanteiden riskiä. Kameratevalvonnan yhdistäminen mobiililaitteeseen ja hälytysten yhteyteen laajentaisi entisestään terminaalin turvallisuuden hallintaa.

3.4.1 Etäkäyttö

Etäkäyttö tarkoittaa laitteen tai järjestelmän käyttöä erilliseltä laitteelta tai päätteeltä. Päätteeltä muodostetaan suojattu yhteys verkon yli, yleensä VPN-yhteyden avulla. VPN (Virtual Private Network) on virtuaalinen näennäisverkko. Käytännössä siis etälaitteelle muodostetaan lähiverkko, johon voi liittyä internetin välityksellä mistä tahansa laitteelta, johon on määritelty kyseisen etälaitteen yhteysmäärittelyt.

Etäkäytöstä puhuttaessa yleisimpiä yrityksiä käytössä olevia sovelluksia on etäyhteys päätteeseen, eli ns. etätyöpöytä, jonka ohjelmistoista esimerkkinä ovat VNC ja VMware. Etätyöpöytäsovelluksia käytettäessä muodostetaan yhteys määrättyyn fyysiseen etätietokoneeseen, jolloin molemmissa laitteissa on identtinen näkymä, etätietokoneen näkymä. Tällöin kaikki muutokset kummalta tahansa päätteeltä tehtynä, jopa hiiren osoittimen liikkuminen, voidaan havaita molemmissa päätteissä.

Etäkäyttöä voi myös suorittaa ilman omaa fyysistä etälaitetta, otetaan tarkasteluun esimerkiksi erillisten palveluntarjoajien maksulliset virtuaalipalvelimet. Virtuaalipalvelimet ovat myös verkon yli vapaasti käytettäviä ja konfiguroitavia fyysisiä päätteitä, joiden käyttöominaisuudet määräytyvät asiakkaan toiveiden mukaan. Palveluvaihtoehtoja on lukuisia, kuten yksinkertainen tiedostojen säilytys ns. pilvipalvelimessa tai kokonaisen käyttöjärjestelmän käyttö virtuaalipalvelimella. Kaikissa edellä mainituissa etäkäyttövaihtoehtoissa on siis sama toimintaperiaate: fyysisen keskuslaitteen käyttö toisella laitteella.

3.4.2 Etäkäyttö teollisuudessa

Etäkäyttösovelluksien hyödyllisyys on myös otettu huomioon teollisuudessa. Automaation etäkäyttö tapahtuu samalla periaatteella kuin pc-etäkäyttö, eli logiikkaohjaimen käyttö tapahtuu erillisellä laitteella verkon yli. Valmistajasta ja laitteesta riippuen on etäkäyttömahdollisuuksia olemassa useisiin eri tarpeisiin, näistä esimerkkinä Siemensin TeleControl-ohjelmisto, jolla voidaan käyttää ja valvoa etälaitoksen logiikkaohjainta GPRS-moduulin välityksellä. TeleControlin avulla voidaan luoda valvomon käyttöliittymästä kopio esimerkiksi laitoksen ulkopuoliseen sisarlaitokseen (8, s. 1). Automaation etäkäyttö on tuonut laitoksen valvontaan ja turvallisuuteen huomattavasti helpotusta. Etäkäytöllä voidaan nostaa myös laitoksen valmiutta reagoida poikkeamiin, myös työajan ulkopuolella. Hälytystiedot saadaan välittömästi esimerkiksi mobiililaitteeseen, josta tarvittaessa hälytyksen voi kuitata, poistaa ja mahdollisesti jopa ennaltaehkäistä.

3.4.3 Etäkäytön tietoturva

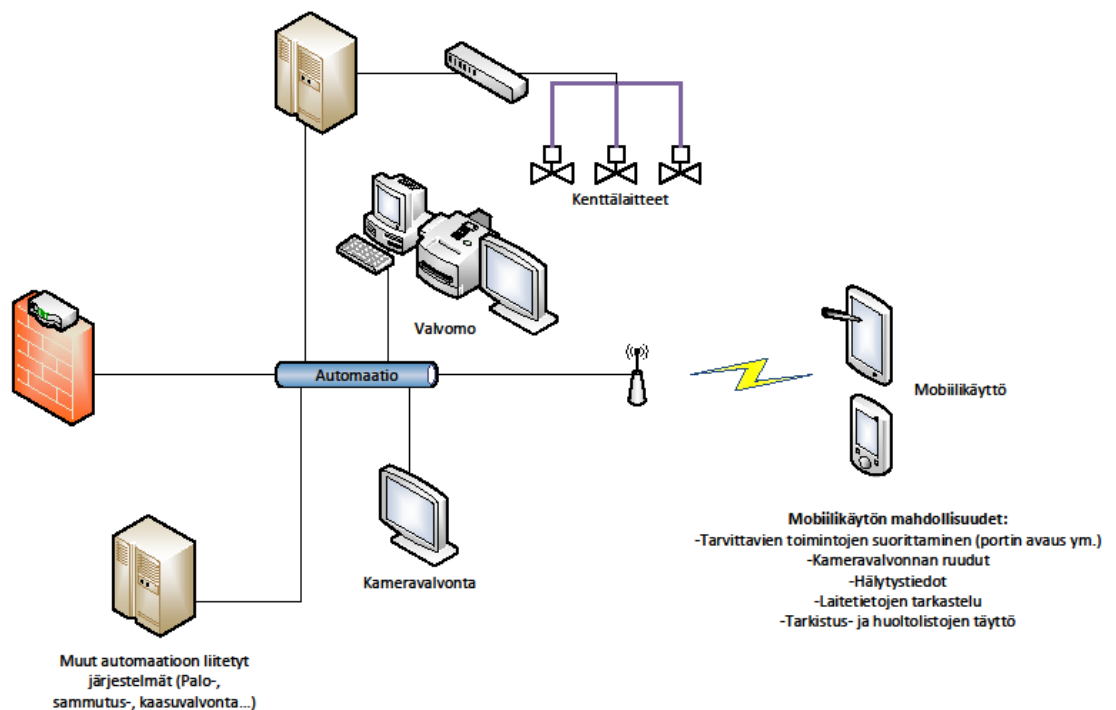
Etäkäytössä on laitoksen turvallisuuden kannalta käänköpuolensa. Aalto-yliopiston vuonna 2013 tekemän tutkimuksen mukaan Suomesta löytyi internetiin liitettyjen järjestelmien hakukoneen (Shodan-hakukone) avulla yhteensä 77 automaatiolaitetta, joista 33 oli nimen perusteella suoraan SCADA-järjestelmiin viittavia. Näistä useimmat olivat linjastojen ja voimalaitosten laitteiston ohjausjärjestelmiä. Tämän tason haavoittuvuus johtui täydellisestä turvallisuus- ja suojausjärjestelmien puutteesta. Joissain tapauksissa jopa käyttäjäsalaus puuttui, mikä mahdollisti laitteistojen suoran käytön. Yksinkertaisilla tuntuvilla ratkaisuilla, kuten palomuurilla, yhteyden salausta- ja suojaustyyppillä tai jopa monimutkaisella salasanalla saadaan tietoturvan tasoa huomattavasti nostettua. (9, s. 4.)

3.5 Yhteys mobiililaitteen ja automaation välillä

Padin kommunikointi keskuslogiikan kanssa piti alun perin hoitaa erillisen Siemens S7-1200 -logiikkaohjaimen avulla, joka olisi liitettyä automaatioverkkoon. S7-1200 olisi toiminut turvallisuussyistä ohjaimena padin ja terminaalien automaation välillä, jottei mobiililaitteella olisi suoraa yhteyttä keskusautomaatioon, niin vikaantumisen kuin väärinkäytön kannalta. Logiikkaohjaimen olisi luotu ohjelma, joka lukisi ja kirjoittaisi, sekä ottaisi vastaan keskuslogiikan sisältöä. Ohjelmointipäätteen ympäristöön luotiin testiksi WLAN-

verkko, jossa S7-1200-logiikkaa onnistuneesti ohjattiin tablettitietokoneella. Logiikkaan luotiin pieni ohjelma, jonka avulla pystyi seuraamaan ja ohjaamaan tulojen ja lähtöjen tiloja. Automaatioverkon langattomaksi saaminen olisi tuottanut huomattavasti lisäkuluja, sillä verkon kantoalueen olisi pitänyt kattaa koko terminaali. Tämä olisi tarkoittanut langattoman verkon tukiasemien asentamista koko alueelle. Terminaalin kattavasta automaation WLAN-verkosta muodostui myös iso tietoturvakysymys. Jos verkon kantoalue olisi yltänyt tietyissä olosuhteissa terminaalin ulkopuolelle, olisi se jo itsessään muodostanut haavoittuvuuden, salauksista huolimatta. Myös alueella toimivien ulkopuolisten urakoitsijoiden ja vierailijoiden pääsy verkkoon olisi täytynyt estää. (ks. Kuva 3.)

Verkko- ja tietoturvakysymyksessä päädyttiin lopulta palomuurin yli muodostettavaan VPN-yhteyteen. VPN-yhteyden tunnelointi- ja salausprotokollat mahdollistavat verkon turvallisen käytön, jolloin mobiililaitetta voidaan käyttää myös laitoksen ulkopuolella.



Kuva 3. Alkuperäinen kommunikaation periaatekaavio WLAN-tukiasemilla

4 Tekniset vaatimukset ja laitteen valinta

Kun käyttökohde on palavien nesteiden varasto, tulee laitteen valinnassa ottaa huomioon sen soveltuvuus räjähdysvaarallisille alueille, yleinen käyttöturvallisuus ja -varmuus, sekä laitteen ulkoisen rasituksen kesto vaihtelevissa olosuhteissa.

4.1 ATEX-soveltuvuus

Tablettia tullaan käyttämään koko terminaalialueen kattavana, jolloin toiminta sijoittuu myös räjähdysvaarallisille alueille, eli Ex-alueille. Ex-alueet jaetaan eri tilaluokkiin räjähdyskelpoisen ilmaseoksen esiintymistodennäköisyyden mukaan (ks. Taulukko 1.). Varastotilat, polttoaineputkistoja ympäröivät alueet, autolastausalue sekä muut pienemmät ja yksittäiset kohteet ovat luokiteltu Ex-alueiksi. Terminaalin Ex-alueet ovat määräysten mukaisesti näkyvästi merkitty keltaisin viivoin ja Ex-varoituskyltein. Ex-alueelle ei saa viedä mitään elektroniikkaa, joka ei ole ATEX-hyväksytty sekä kyseiseen tilaluokkaan sopiva. Laitteiden tilaluokitukset määräytyvät pääosin niiden tiiveyden, suojauksen sekä käyttölämpötilan perusteella.

Taulukko 1. ATEX-tilaluokat ja niiden määritteet (4, s. 5)

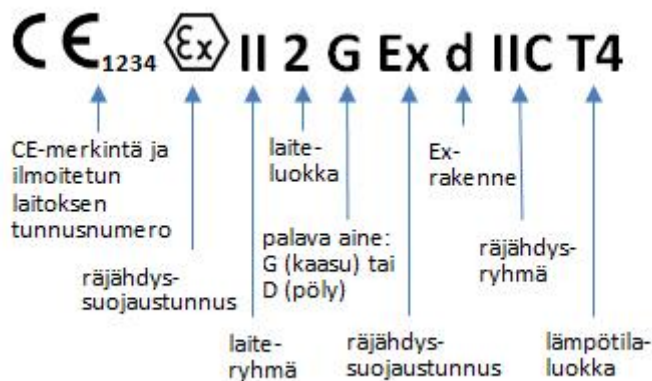
| Syttyvät materiaalit | Tilaluokka | Räjähdyksvaarallisen ilmanalan hetkelliskäyttö |
|----------------------|------------|--|
| Kaasut ja höyryt | 0 | Tila, jossa ilman ja kaasun, höyryn tai sumun muodossa olevan palavan aineen muodostama räjähdyskelpoinen ilmaseos esiintyy jatkuvasti, pitkä- aikaisesti tai usein. |
| | 1 | Tila, jossa ilman ja kaasun, höyryn tai sumun muodossa oleva palavan aineen muodostama räjähdyskelpoinen ilmaseos esiintyy normaalitoiminnassa satunnaisesti. |
| | 2 | Tila, jossa ilman ja kaasun, höyryn tai sumun muodossa olevan palavan aineen muodostaman räjähdyskelpoisen ilmaseoksen esiintyminen normaalitoiminnassa on epätodennäköistä ja se kestää esiintyessään vain lyhyen ajan. |
| Pölyt | 20 | Tila, jossa ilman ja palavan pölyn muodostama räjähdyskelpoinen ilmaseos esiintyy jatkuvasti, pitkä- aikaisesti tai usein. |
| | 21 | Tila, jossa ilman ja palavan pölyn muodostama räjähdyskelpoinen ilmaseos esiintyy normaalitoiminnassa satunnaisesti. |
| | 22 | Tila, jossa ilman ja palavan pölyn muodostaman räjähdyskelpoisen ilmaseoksen esiintyminen normaalitoiminnassa on epätodennäköistä ja se kestää esiintyessään vain lyhyen ajan. |

Otetaan tilaluokituksista esimerkkinä bensiini- ja dieselvarasto. Varastosäiliöt sijaitsevat ilmastoidussa, suljetussa tilassa. Säiliöissä on bensiiniä sekä dieseliä. Ilmanvaihdoista huolimatta varastossa voi aika ajoin esiintyä hiilivetyjä, jotka muodostavat ilman kanssa räjähdyskelpoisen ilmaseoksen. Kaasuja voi ilmaantua varaston operointialueille esimerkiksi polttoaineiden siirron aikana, jolloin täytettävän säiliön pinnan noustessa sisään virtaava polttoaine syrjäyttää säiliöön syntyneet kaasut. Normaalisti nämä kaasut johdetaan syrjäytymiskaasujen käsittelylaitoksiin, mutta kaasunpaineen kasvaessa korkeaksi säiliöiden päällä olevat varoventtiilit voivat satunnaisesti päästää pieniä määriä höyryjä ulos. Ex-alue on tällöin tilaluokitukseltaan 1, sillä kaasut esiintyvät normaalitoiminnassa satunnaisesti. Säiliöiden sisäosat ovat lueteltu vastaavasti 0-tilaluokkaan.

Laiteluokituksien merkinnät

ATEX-hyväksytyissä laitteissa tulee olla sertifikaatin mukaiset merkinnät (ks. Kuva 4.). Merkinnöistä käy ilmi laitteen soveltuvuus kyseiseen ATEX-luokiteltuun tilaan. Tilaluokassa 0 tai 20 käytetään laiteluokan 1 laitteita. Tilaluokassa 1 tai 21 käytetään laiteluokan

1 tai 2 laitteita, sekä tilaluokassa 2 tai 22 käytetään vastaavasti laiteluokan 1, 2 tai 3 laitteita. Laitteen soveltuvuus räjähdysvaaran aiheuttavalle palavalle aineelle tulee olla oikea, tässä tapauksessa G = kaasu (gas). Räjähdysuojatunnus Ex sekä laitteen Ex-rakenne kertovat laitteen rakenteellisesta räjähdysuojasta (d = räjähdyspaineen kestävä). Räjähdysryhmä on jaettu kahteen ryhmään: Ryhmä I on kaivoksissa käytettävät laitteet ja ryhmä II on kaikissa muissa räjähdysvaarallisissa tiloissa kuin kaivoksissa käytettävät laitteet. C-kirjan räjähdysryhmän perässä on täsmennys laitteen Ex-rakenteeseen: C = MESG < 0,5 mm, tarkoittaa räjähdyspaineen kestävä kotelon testauksessa määritettyä maksimiturvarakoa, *Maximum Experimental Safe Gaps*. T4 on laitteen lämpötilaluokka. Lämpötilaluokat määräytyvät laitteen suurimman sallitun pintalämpötilan sekä kaasun itsesyttymislämpötilan mukaan. (3, s. 19–29.)



Kuva 4. ATEX-laitteen merkinnät ja niiden selitteet (6.)

4.2 Ecom Tab-Ex® Series

ATEX-luokitellut tablettitietokoneet eivät vielä insinööriyön aloitushetkellä tarjonnallaan loistaneet. Monia 2-tilaluokkaan, sekä rankempiin olosuhteisiin (ns. rugged) sopivia mobiililaitteita oli saatavilla, kuten Panasonicin Toughbook. Vasta työn aloittamisen jälkeen Ecom toi yhteistyössä Samsungin kanssa markkinoille Android 5.1.1 (Lollipop) -käyttöjärjestelmällä ja 8" näytöllä varustettu 1-tilaluokan Tab-Ex Zone 1/21 -tablettitietokoneen. Tabletti on suunniteltu petrokemian ja lääketeollisuuden laitoksiin, jossa operointia suoritetaan erilaisten kemikaalien ja polttoaineiden vuoksi räjähdysvaarallisilla, sekä olosuhteiltaan haastavilla alueilla. Laitteen tekniset ominaisuudet:

- Android 5.1.1 Lollipop-käyttöjärjestelmä
- 1.2GHz neliydinprosessori, 1.4GB RAM ja 16GB tallennustila
- 8” näyttö, jota voi käyttää hansikkailla
- 3 megapikselin kamera sekä 1.3 megapikselin etukamera
- GPS, jota myöhemmin voi hyödyntää kartta- ja paikannussovelluksissa
- NFC.

Laitteeseen oli saatavilla todella paljon terminaalin olosuhteiseen sopivia ja käyttöä helpottavia lisätarvikkeita, kuten ajoneuvoon kiinnitettävä telakka, olkahihna ja stylus-kynä.

Laitteen ATEX-sertifikaatin mukaiset merkinnät:

- II 2 GD
- Ex db ia op is IIC T5 Gb
- II 2D Ex tb IIIC T100°C Db
- Ta = -20°C to +50°C



Kuva 5. Ecom Tab-Ex Zone 1/21 -tablettitietokone

4.3 Tunnisteiden lukeminen

Tunnisteiden lukeminen tultaisiin luomaan aluksi viivakoodipohjaisena, eli tablettitietokoneella tulisi olla valmius lukea viivakoodeja. Alkuperäisten suunnitelmien mukaan haussa oli NFC/RFID-lukuvalmiudella varusteltu tablettitietokone, mutta Ex-tiloihin soveltuvien laitteiden tarjonnan ja niiden rajoittuneiden ominaisuuksien vuoksi päädyttiin viivakoodilukuun. Viivakoodien, QR-koodien ja muiden graafisten tunnisteiden lukemiseen riittää pelkkä kamera, jossa kuitenkin tulisi olla nopea tarkennus ja riittävä tarkkuus pienempienkin viivakooditarrojen lukemiseen. Myös kameran valoisuus hämärissä olosuhteissa tulisi olla riittävällä tasoilla, sillä operoitavia laitteita on myös heikossa valaistuksessa. Viivakooditarrojen hankkiminen, tai tässä tapauksessa tekeminen, muodostuu myös huomattavasti muovisia RFID-tageja halvemmaksi, sillä niiden tekeminen ja tulostaminen tietokoneella on todella vaivatonta.

Vaikka käyttöön tulevassa laitteessa onkin NFC-ominaisuus, mallisovelluksessa hyödynnetään siitä huolimatta viivakoodilukua. Ecomin tablettitietokoneen NFC-toiminto tuli ilmi vasta laitteen tilauksen yhteydessä ja sovellukseen oli jo ohjelmoitu viivakoodilukua hyödyntävä ohjelma, joka todettiin toimivaksi ja tarpeiden mukaiseksi.

5 Ohjelmointi

5.1 Android Studio

Laitteen ohjelmointi suoritettiin Android Studio -ohjelmiston avulla. Android-ohjelmoinnissa käytetään Java-kieltä sekä ulkoasun ohjelmointiin XML-kieltä. Android Studio on täysin ilmainen ohjelma ja vapaasti ladattavista Android-kehittäjien nettisivuilta (1). Ohjelma pohjautuu JetBrainsin IntelliJ IDEA -koodausohjelmaan, joka on suunniteltu Javan koodauskäyttöön. Ohjelman laaja ja toisaalta kompakti käyttöliittymä mahdollistavat tehokkaan koodauksen kaikkine lisäominaisuuksineen, kuten älykkään virheenetsinnän ja koodinkääntäjän. Android-ohjelma koostuu kahdesta pääosasta: graafisesta käyttöliittymästä (GUI) ja itse koodista. GUI:n kokoonpanossa käytetään XML-formaattia ja ohjelman lähdekoodin luomisessa Javaa.

5.2 Viivakoodien luku

Viivakoodien lukemisessa mallisovelluksessa käytetään ZXing-kirjastoa. ZXing on ilmainen, avoimen lähdekoodin kirjasto, jossa on valmius lukea 1D/2D-viivakoodeja. Kirjastosta karsittiin kaikki online-sisältö pois, jolloin sitä voidaan käyttää ilman internet-yhteyttä. Tätä menetelmää kutsutaan Scanning via Intent -tavaksi. Tällöin kirjastosta käytetään vain IntentIntegrator.java- ja IntentResult.java-ohjelmatiedostoja, eli kutsuvaan ohjelmaan integroiva ohjelmakoodi sekä lukutuloksen käsittelykoodi. Kevennetty ZXing riittää ominaisuuksiltaan demonstroimaan mallisovelluksessa käytettävien tunnisteiden lukemisen. (ks. Liite 1.)

Kirjaston toiminta on yksinkertainen. Kirjaston luku-ominaisuutta kutsutaan aktiviteetissä, jolloin viivakoodiluku avautuu laitteen kameran ruutuun (ks. Kuva 6.). Ensimmäistä kertaa lukuominaisuutta käytettäessä laite pyytää lupaa ladata tarvittava lukuapplikaatio internetin välityksellä, ellei tätä ole jo tehty. Latauksen valmistuttua laite on valmis jatkossa käytettäväksi ilman internet-yhteyttä. Kamera kohdistetaan luettavaan tunnisteeseen, jolloin laite tarkentaa koodiin ja automaattisesti lukee viivakoodin, mikäli riittävä kuvantarkkuus on saavutettu. Luettuaan koodin sovellus palautuu kutsuvaan aktiviteettiin lukutulos string-muodossa mukanaan. Saatua tulosta voi käyttää haluamallaan tavalla.



Kuva 6. Viivakoodin lukunäkymä

5.3 Yhteys etätietokoneeseen

Etätietokoneen levyjakoon päästiin käsiksi SMB/CIFS-protokollan avulla. SMB/CIFS on Windowsin ja kolmansien osapuolien väliseen levyjakoon vaadittava protokolla. Protokolla mahdollistaa kahden eri tietokoneen keskinäisen yhteyden samassa verkossa. Androidia ei suoraan voida yhdistää Windowsin levyjakoon, mutta tätä tarkoitusta varten on olemassa kirjasto, JCIFS. JCIFS on kirjasto SMB/CIFS-protokollan käyttämiseen Javalla. (6, s. 1.)

5.3.1 Yhteyden muodostaminen

JCIFS-kirjasto sisältää valmiit luokat ja komennot toimivan levyjaon luomiseen Windowsin ja Androidin välille. Levyjaon muodostamiseen vaaditaan etäkoneen IP-osoite, käyttäjätunnus ja salasana. Näillä tiedoilla voidaan suorittaa NTLM-todennuksen, jolla varmistetaan kohdetietokoneesta ja käyttäjästä (ks. Kuva 7). NTLM on Windowsin oma autentikointiprotokolla, jonka avulla varmistetaan paikallisverkon välillä toiseen koneeseen asiaankuuluva pääsy.

```

public void haetiedosto(final String etaTiedostonNimi, final String etaTiedostonPolku, final String
    new Thread(new Runnable() {

        @Override
        public void run() {

            String USER_NAME = "xxxxxxx";
            String PASSWORD = "xxxxxxx";

            String NETWORK_FOLDER = "smb://192.168.0.1/Kansio/";

            String FINAL_PATH = NETWORK_FOLDER + etaTiedostonPolku + "/";

            boolean successful = false;
            try {
                String user = USER_NAME + ":" + PASSWORD;
                NtlmPasswordAuthentication auth = new NtlmPasswordAuthentication(user);
                String path = FINAL_PATH + etaTiedostonNimi;

```

Kuva 7. Esimerkkiote NTLM-todentamisesta JCIFS-kirjastolla

NTLM-todennuksen onnistuttua, JCIFS-kirjaston komennoilla suoritetaan haluttu toimennepide, esimerkiksi kopioidaan levyjaosta tiedosto mobiililaitteelle (ks. Kuva 8.).

```

SmbFile sFile = new SmbFile(path, auth);
File lFile = new File(sdcard + "/" + tiedosto);
lFile.createNewFile();

SmbFileInputStream sfis = new SmbFileInputStream(sFile);
FileOutputStream sfos = new FileOutputStream(lFile);

byte[] b = new byte[8192];
int n, tot = 0;
while ((n = sfis.read(b)) > 0) {
    sfos.write(b, 0, n);
    tot += n;
}

sfis.close();
sfos.close();

```

Kuva 8. Tiedoston kopiointi levyjaolta mobiililaitteelle

5.3.2 Levyjaon yhteysongelmat

Androidilla ei suoraan ole valmiutta levyjakoon Windows-koneen kanssa. Windowsista täytyy määrittää yhteysasetuksia ja rekisteritietoja Android-laitteelle sopivaksi, sekä Androidilla käyttää sovellusta, jossa SMB/CIFS-protokollan käyttö on sovitettu Androidille sopivaksi, esimerkiksi mallisovelluksessa käytössä olevan JCIFS-kirjaston avulla. Levyjaon testaamiseksi ja tiedostonhallintaan käytettiin AndSMB-sovellusta (2, s. 1.).

Levyjaon käytössä ilmeni lopulta ensikäytön jälkeen ongelma. Määrätyn ajan jälkeen levyjako lakkasi toimimasta. Mobiililaitteen VPN-yhteys toimi moitteetta, mutta nimenomaan levyjako Androidin ja Windowsin välillä ei suostunut toimimaan. Levyjako saatiin palautettua takaisin toimintaan käynnistämällä Windowsin päästä levyjakoa ylläpitävä palvelu. Yhteys toimi taas viikon verran kunnes lakkasi toimimasta. Tämän vuoksi suuri osa mobiililaitteen toiminnoista jäi käytännössä testaamatta. Levyjakoa ja tietohallintajärjestelmiä hallinnoiva osapuoli ilmoitti selvittävänsä asian ja etsivänsä asiaan ratkaisun.

Levyjaon ylläpitopalvelun jatkuva uudelleenkäynnistäminen, esimerkiksi määrättyyn kelloonaikaan päivittäin, ei ole vaihtoehto, sillä sama palvelu ylläpitää myös muita yhteyksiä. Tämä saattaisi luoda ongelmia, mikäli kesken toisen sanoman lähetyksen tai vastaanoton palvelu käynnistettäisiin uudelleen. Suunnitteilla on myös vaihtoehtoisia yhteystapoja, kuten tietokantapohjainen tiedonsiirto, mikäli ongelmaan ei löydetä ratkaisua.

5.4 Käyttäjätietokanta

Yksinkertaisin tapa nostaa tietoturvan tasoa on luoda sovellukselle käyttäjätietokanta. Jokaisella käyttäjällä on omat henkilökohtaiset tunnukset, joiden avulla heillä on käyttöoikeus ohjelmaan. Tarvittaessa ohjelmaan voidaan luoda eri käyttäjätasoa, jolloin alemman tason käyttäjillä ei ole pääsyä tai oikeuksia kaikkiin ohjelman ominaisuuksiin. Käyttäjähierarkiasta yksinkertainen esimerkki voisi olla kolmeen käyttäjätasoon jaettu hierarkia. Alin taso olisi ns. peruskäyttäjä, jonka oikeudet kattaisi ohjelman tarvittavat perustoiminnot. Seuraava taso olisi valvojatasa, jolla olisi pääsy osaan ohjelman tarkemmista asetuksista ja säädöistä. Ylin taso, eli huolto tai ohjelmointitasa, kattaisi esteettömän pääsyn koko ohjelmaan. Mikäli ohjelmaan sisällytetään haavoittavia ominaisuuksia tai asetuksia, on käyttäjähierarkian luominen olennainen osa käyttöturvan takaamiseksi. Tällöin voidaan ennaltaehkäistä esimerkiksi ohjelman käyttökoulutuksen puutteesta johtuvat vahinkotapaukset tai jopa tahallinen väärinkäyttö. Tapahtumaloki sidottuna kirjautuneeseen käyttäjään auttaa myös ongelmien jälkiselvityksessä.

5.4.1 Toiminta

Androidissa itsessään on valmius SQLite-tietokantaan. SQLite on kevyt, paikallinen levyille tallentava tietokanta. Tietokantaan luotiin taulukko, johon sisällytettiin käyttäjänimi,

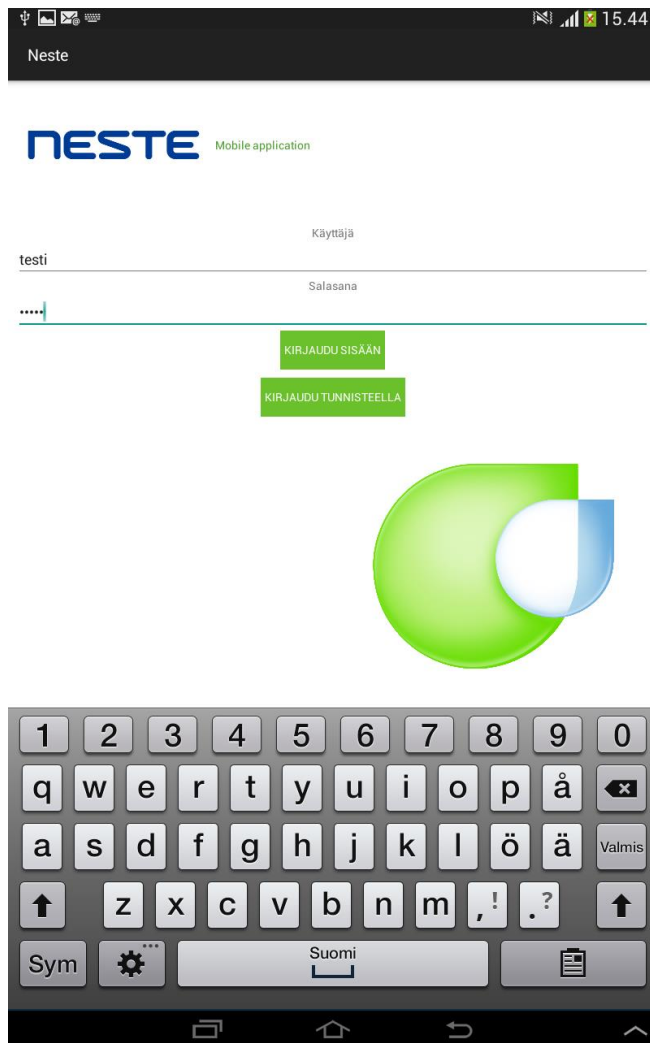
salasana, etunimi, sukunimi, henkilökohtainen viivakooditunniste. Kirjautumistapoja on kaksi, perinteinen käyttäjänimen ja salasanan avulla, sekä henkilökohtaisen tunnisteeseen avulla. Henkilökohtainen tunniste voi olla RFID- tai viivakooditunniste, ja nämä molemmat ovat sisällytettyinä laitoksen työntekijöiden henkilökorttiin. Mallisovelluksessa käytetään viivakooditunnisteeseen lukua.

Käyttäjän kirjautuessa käyttäjänimellä ja salasanalla sisään kirjautumissivun tekstikenttään syötettyjä tietoja verrataan tietokannassa oleviin (ks. Kuva 9.). Jos käyttäjänimi löytyy tietokannasta, verrataan sen yhteyteen liitettyä salasanaa kirjoitettuun salasanaan. Vastaavasti tunnisteella kirjautuessa verrataan sitä tietokantaan. Mikäli nämä tiedot täsmäävät, käyttäjä saa luvan kirjautua sisään ja ohjelma siirtyy päävalikkoon. Kirjautunut käyttäjä näkyy päävalikon yläkulmassa, muodossa etunimi sukunimi. Jos tiedot eivät täsmää, ohjelma ilmoittaa ”Käyttäjää ei tunnistettu” ja kirjautuminen hylätään.

Käyttäjätietoja voidaan hyödyntää käyttökokemuksen tehostamiseksi. Käyttäjäleiman automaattinen kirjautuminen varmistaa käyttäjän vastuun tekemistään muutoksista ja toiminnoista ja samalla nopeuttaa operointia. Esimerkiksi päiväkirjamerkintöjä kirjoittaessa merkinnän kuittaus tulee automaattisesti kirjautuneen käyttäjän nimellä.

5.4.2 Kehitysideat

Myöhemmin sovellukseen on tarkoituksena lisätä RFID-kirjautumistoiminto sekä tietoturvallisempi tietokanta, joka ei olisi paikallisesti laitteessa käytössä, vaan esimerkiksi etätietokoneella ja saatavissa vain oikeamuotoisen kutsun avulla, jolloin tiedot lähetettäisiin salattuina. Kehitysmielessä helpoin ja käyttäjäystävällisin ratkaisu oli saada niin automaatiolle kuin mobiililaitteelle yhdistetty käyttäjätietokanta, jolloin muutokset käyttäjärekisteriin hoituisi yhtenäisesti ja hallitusti. Suunnitteilla on myös käyttäjätietojen tarkka tapahtumaloki, josta kaikki käyttäjän tekemät toiminnot ohjelmassa kirjattaisiin lokitiedostoon.



Kuva 9. Kirjautumissivu

5.5 Päiväkirjasovellus

Päiväkirjasovellus toimii yksinkertaisella periaatteella. Merkintöjä voidaan kirjoittaa ja lukea mobiililaitteella vapaasti ja tarvittaessa hakea eri suodattimien avulla.

5.5.1 Merkinän kirjoittaminen

Mobiililaitteella luotu merkintä tallentuu päiväkirjatiedostoon, etätietokoneen levyjaolle sekä mobiililaitteelle. Mikäli merkintä on tehty ilman toimivaa yhteyttä etätietokoneeseen, tallentuu merkintä pelkästään paikallisesti laitteeseen. Seuraavalla kerralla yhteyden muodostuessa ja päiväkirjaa käytettäessä uusimmat merkinnät kopioituvat etätietokoneelle.

Päiväkirjamerkintä koostuu kuudesta eri osasta: milloin merkintä on merkitty (Merkitty), tapahtuman päivämäärä, tapahtuman kellonaika, tapahtuman sijainti (Kohde), tapahtuman kuvaus (Merkintä) sekä merkinnän kuittaus. Lisäksi halutessaan käyttäjä voi liittää kuvan merkinnän yhteyteen. Merkinnän kellonaika ja kuittaus kirjautuvat automaattisesti merkinnän yhteyteen (ks. Kuva 10.).

Merkinnän yhteyteen voidaan liittää kuva. Liittäminen tapahtuu merkintäsivun ”Liitä kuva”-painikkeesta, jolloin kamerasovellus avautuu ja otettu kuva tallentuu liitteeksi. Halutessaan otettu kuva voidaan hylätä ja ottaa uusi kuva tilalle.

Uusi merkintä

Päiväkirjamerkintä

23 Mar 2014

Tapahtuman päivämäärä: 24 Jou 2015

25 Tam 2016

12 52

Tapahtuman kellonaika: 13 : 53

14 54

Kohde: Kaikki

Merkintä:

Kuittaus: Nino Silvennoinen

LIITÄ KUVA TALLENNNA MERKINTÄ

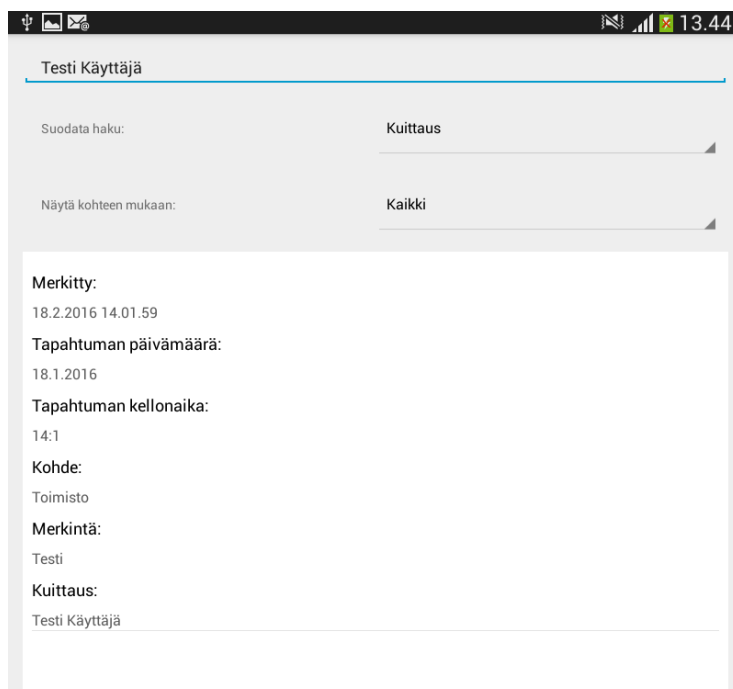
Kuva 10. Päiväkirjamerkintä

5.5.2 Päiväkirjan lukeminen

Päiväkirjan lukeminen muodostettiin ListView-toiminnolla. Lista hakee päiväkirjatiedosta kaikki merkinnät ja jakaa ne omiin osioihinsa. Listalta voidaan hakea merkintöjä suodattimien avulla, jolloin valittujen kriteerien täytyessä listalle jää kyseiset merkinnät. Päiväkirjan hakukenttä on sidottu haun suodattimiin, eli kaikki mitä hakukenttään syötetään, haetaan valitun suodattimen takaa. Jos haun suodattimeksi on valittu esimerkiksi

Kuittaus, kohdistuu haku tällöin vain merkinnän kuittauksiin. Eli jos haetaan kuittausta nimellä Testi Käyttäjä, listalle jää ainoastaan kaikki Testi Käyttäjän kirjoittamat merkinnät (ks. Kuva 11.).

Päiväkirjamerkinnot voidaan näyttää myös suoraan sijainnin perusteella. Valikosta Näytä kohteen mukaan valitaan haluttu sijainti ja lista päivittyy näyttäen kaikki valitun sijainnin merkinnät. Sijainnit ovat tällä hetkellä vanhojen päiväkirjasijaintien mukaiset, mutta tarvittaessa niitä voidaan lisätä, mikäli tarve vaatii. Päiväkirjamerkinnot yhteyteen liitetyt kuvat saadaan auki klikkaamalla merkintää. Ruutuun avautuu pienempi popup-ikkuna, jossa liitekuva näkyy.



Kuva 11. Päiväkirjan hakutoiminnon suodatus

5.5.3 Kehitysideat

Päiväkirjamerkinnot voidaan hyödyntää automaation ja tietojenhallintajärjestelmän käytössä. Tarvittaessa voidaan luoda uusia kenttiä merkinnän yhteyteen, kuten onko kyseessä huoltotoimenpide, jolloin merkinnän tiedot voidaan liittää suoraan laitetietokannan huoltorekisteriin. Myös merkinnän liitettävyyden suoraan laitteeseen toisi hyötyjä, jolloin esimerkiksi automaation valvomonäytöltä näkisi suoraan laitetta tarkastellessa kaikki

päiväkirjamerkinnot. Myös muiden liitetiedostojen kuin pelkästään kuvien liittäminen tehostaisi päiväkirjan käytettävyyttä. Päiväkirjaominaisuus tullaan myös luomaan valvomopäätteille, joista pystytään myös lukemaan ja kirjoittamaan samaa päiväkirjaa.

5.6 Automaation toiminnot

Automaation etäkäyttöön otettiin testikohteeksi portin avaus ja sen tilatietojen lukeminen. Toimintoa ei päästy testaamaan käytännössä automaation kanssa, johtuen etätietokoneen ja Android-laitteen välisistä yhteysongelmista (ks. Luku 5.3.2). Avaustoimintoa kuitenkin simuloitiin onnistuneesti testikäyttöön luodun WLAN-verkon sisällä.

Portin avaustoiminto luotiin yksinkertaisella luku- ja kirjoitusmenetelmällä. Suoraa yhteyttä ei logiikkaohjaimeen luotu, vaan kommunikointi hoidetaan csv-tiedostojen avulla. Tiedostoja on kaksi, luku- ja kirjoitustiedostot. Mobiililaitte kirjoittaa etätietokoneen levyjaolla sijaitsevaan kirjoitustiedostoon rivin, joka sisältää toiminnon ja tilan. Automaatio tunnistaa muutoksen tiedostossa, lukee rivin ja suoritettuaan toiminnon poistaa rivin. Samalla automaatio kirjoittaa lukutiedostoon rivin, josta käy ilmi, että portti on auki. Ohjelma tunnistaa muutoksen tiedostossa, päivittää kohteen tilatiedon ja lopuksi poistaa rivin lukutiedostosta.

Vaikka portin avauspyyntö toimi testiympäristössä moitteetta, automaation tilatietojen tarkkailu on vielä kehitysasteella. Levyjaon tiedostojen muutostenhallinta Androidin päässä osoittautui haastavaksi. Androidille on olemassa valmis luokka (FileObserver), jolla laitteen sisäisiä tiedostoja ja kansioita ja niihin tehtyjä muutoksia pystytään tarkkailemaan. Luokan tueksi voidaan luoda taustapalvelu, joka ilmoittaa tai tarvittaessa suorittaa halutun toiminnon muutoksen perusteella. Tämän luokan ja taustapalvelun soveltaminen SMB-protokollaan ei onnistunut suoraan. Toiminnon kehitys jatkuu ja vaihtoehtoisia menetelmiä on harkinnan alla, kuten tietokantapohjainen kommunikointi.

5.7 Laittekohtaisten tunnisteiden lukeminen

Laitteiden luku -toiminnon alla on seuraavat vaihtoehdot: Laitteen tiedot, Laitteiden tarkastus ja Laitteiden linjaus. Tunnisteiden lukemisessa käytetään viivakoodilukua.

5.7.1 Laitteen tiedot

Laitteen tietojen hakemisessa käytetään laitekohtaista tunnistetta, mallisovelluksessa viivakoodia. Ohjelmalla luetaan laitteen viivakoodi ja tämän jälkeen saadaan näytölle laitteen tarkemmat tiedot. Viivakoodin avulla saadaan jokainen laite yksilöityä ja kutsusta haettua kunnossapitojärjestelmästä tarvittavat tiedot. Tämä ominaisuus ei ole vielä otettu käyttöön ja on vielä kehityksen alla.

5.7.2 Laitteiden tarkastus

Laitteiden tarkistuksessa voidaan hoitaa yksittäisiä laitteiden huoltotarkistuksia tai isompia tarkistuskokonaisuuksia, kuten laitteiston viikkotarkastus, jossa tarkastuskohteita voi olla monia. Listan kohteet ovat tallennettu etätietokoneen levyjakoon. Jokainen luettava tarkastuslista on csv-tiedostoformaattissa. Csv-tiedoston jokainen rivi on yksi tarkastusrivi. Yksi rivi on jaettu eri kolumneihin, jotka tarkastuksen tyypistä riippuen kertovat yhden tarkastuskohteen tiedot.

Otetaan esimerkiksi huoltotarkastus (Kuva 12.).

```
100;Testitarkastus;
g;10;RYHMÄ 1
t;1;KOHDE 1;0
t;2;KOHDE 2;1
t;3;KOHDE 3;2
T;4;KOHDE 4;0
t;5;KOHDE 5;2
t;6;KOHDE 6;1
G;20;RYHMÄ 2
t;7;KOHDE 1;0
t;8;KOHDE 2;1
t;9;KOHDE 3;0
t;11;;0
t;;;0
t;13;;0
```

Kuva 12. Huoltotarkastuksen csv-tiedoston muoto

Ohjelma lukee ensimmäisen rivin ja saa sen ensimmäisestä kolumnista tarkastusnumeron (100). Tarkastusnumero on yksilöllinen joka tarkastukselle, ja jo pelkästään tällä voidaan tunnistaa ja nimetä tarkistus. Seuraavassa kolumnissa on kuitenkin tarkastuksen nimi (Testitarkastus), jotta nimi olisi helppo tulostaa tarkastuksen otsikoksi.

Seuraavan rivin ensimmäisessä kolumnissa on g-kirjain (group), joka määrittää seuraavat rivit yhdeksi tarkastusryhmäksi. Ryhmän numero on 10 ja nimi Ryhmä 1. Seuraava rivi on ensimmäinen tarkastuskohde (t), jonka numero on 1 ja nimi Kohde 1. Tarkastusrivin viimeisen kolumnin numero määrittää tarkastustyyppin.

- 0 = Valintapainikkeet (OK, Korjattu, Vikaa)
- 1 = Lukuarvokenttä
- 2 = Tekstikenttä

Isot ja pienet kirjaimet rivien alussa, sekä tyhjät kolumnit ovat ohjelman toimivuuden testauksen kannalta tehtyjä. Näin saatiin testivaiheessa varmistettua, että ohjelma lukee pieniä ja isoja kirjaimia sekä tunnistaa tyhjät kolumnit. Näillä tiedoilla saadaan luotua tarkastuslista, joka on muodoltaan sama kuin huoltojärjestelmän tarkastukset (ks. Kuva 13.)

The screenshot shows a mobile application interface for device testing. The title bar at the top reads "Laitteiden tarkastus". Below it, the main heading is "Testitarkastus". The interface is organized into two groups, "RYHMÄ 1" and "RYHMÄ 2".

Under "RYHMÄ 1", there are six test items (KOHDE 1 to KOHDE 6). Each item has a status indicator consisting of three radio buttons: "OK", "Korjattu", and "Vikaa".

- KOHDE 1: Status indicators are present.
- KOHDE 2: Input field with the text "Syötä lukuarvo".
- KOHDE 3: Input field with the text "Syötä teksti".
- KOHDE 4: Status indicators are present.
- KOHDE 5: Input field with the text "Syötä teksti".
- KOHDE 6: Input field with the text "Syötä lukuarvo".

Under "RYHMÄ 2", there is one test item (KOHDE 1) with status indicators.

At the bottom of the screen, there is a "TALLENNA" button and a "TULOS" label. The Android navigation bar is visible at the very bottom.

Kuva 13. Laitteiden tarkastus – testilistanäkymä

Lista käydään tarkastuskohde kerrallaan läpi. Mikäli monivalintapainikkeista valitaan Korjattu tai Vikaa, ilmestyy ruutuun pakollinen kommenttikenttä, johon käyttäjän täytyy kertoa, mitä vikaa tai mitä laitteesta korjattiin ja milloin.

Kun lista on käyty läpi, voidaan tarkastus tallentaa. Jos jokin kohde jäi tarkastamatta, lista ilmoittaa pienellä tekstillä ruudun alalaidassa (Toast), mikä tai mitkä kohteista on kyseessä ja maalaa syöttökentän tai painikkeet punaisiksi. Valmiista tarkastuksesta saadaan raportti, joka on myös csv-tiedosto. Tiedosto tallennetaan levyjaolle kansioon, johon huoltojärjestelmä on yhteydessä. Tällöin huoltojärjestelmään saadaan kirjattua tehdyt tarkastukset.

5.7.3 Laitteiden linjaus

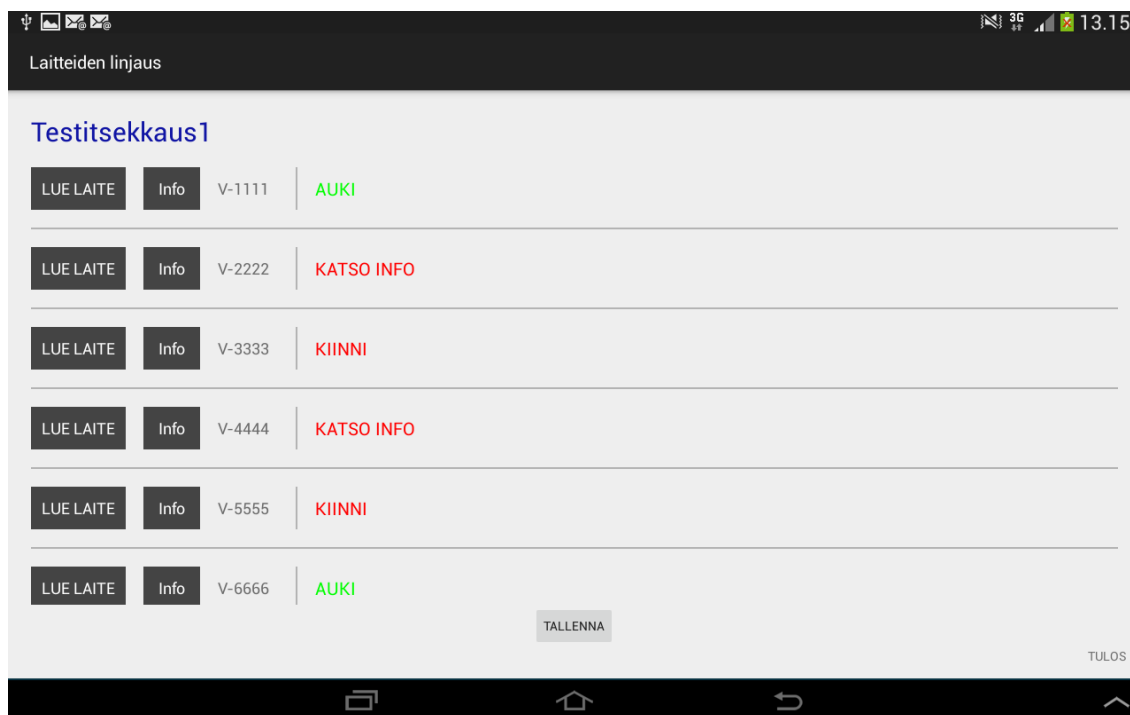
Linjaustoiminto luotiin tilanteisiin, jossa esimerkiksi putkilinjan varrella olevien venttiilien ja laitteiden tilat tulee tarkastaa ennen niiden käyttöä. Esimerkkiohjelmassa aukeaa lista eri tarkastuskohteista, joista valitaan haluttu toimenpide, esimerkiksi Siirto S-15 – S-16. Listat ovat myös csv-muodossa, jolloin pystytään uudelleenhyödyntämään ohjelmaan aiemmin luotuja csv-lukusovelluksia (ks. Luku 5.7).

Otetaan tarkasteluun linjauksen esimerkkitiedosto (ks. Kuva 14.). Ensimmäiseltä riviltä poimitaan tarkastuksen nimi ja tiedostonimi (Testaus & Testitsekkaus). Seuraava rivi on ensimmäinen tarkastuskohde. Ensimmäinen kolumni sisältää laitteen yksilöllisen tunnisteen. Tämä tullaan lukemaan viivakoodina tunnistamista varten. Toinen kolumni on tarkastuskohteen tunnus. Kolmas on tarkastuksen tyyppi, mikä tässä kohtaa ei vaikuta mihinkään, sillä jokainen tarkastustyyppi on oletuksena viivakoodiluku. Arvo "0" merkitään ohjelman toimivuuden kannalta, jotta kolumnissa olisi jotain tietoa, tosin tyhjää sellaista. Tarkastuksen tyyppi -kolumni on jätetty olemaan mahdollista tulevaa tietoa tai toimintoa varten. Neljäs kolumni pitää sisällään tarkastuskohteen kuvauksen. Viides kolumni kertoo tarkastuksessa vaadittavan laitteen tilan.

```
Testaus;Testitsekkaus;;;
1111;V-1111;0;Tuotelinjan sulkuventtiili;AUKI
2222;V-2222;0;Tuotelinjan ilmausventtiili - Normaaliikäytössä AINA kiinni;KATSO INFO
3333;V-3333;0;Tuotelinjan tyhjennysventtiili;KIINNI
4444;V-4444;0;Tuotelinjan säätöventtiili - Normaaliikäytössä asento 50%;KATSO INFO
5555;V-5555;0;;;KIINNI
6666;V-6666;0;Tuotelinjan sulkuventtiili;AUKI
;FV-7777;0;Laitteen tarkemmat tiedot;AUKI
8888;P-8888;0;;;KIINNI
9999;MV-9999;0;Laivauslinjan loppupään sulku;AUKI
1112;HV-1112;0;Pumppu 2. paineventtiili;KIINNI
1113;FFV-1113;0;Jep;KATSO INFO
1114;P-1114;0;Juu;KIINNI
1115;GA-1115;0;AUKI
```

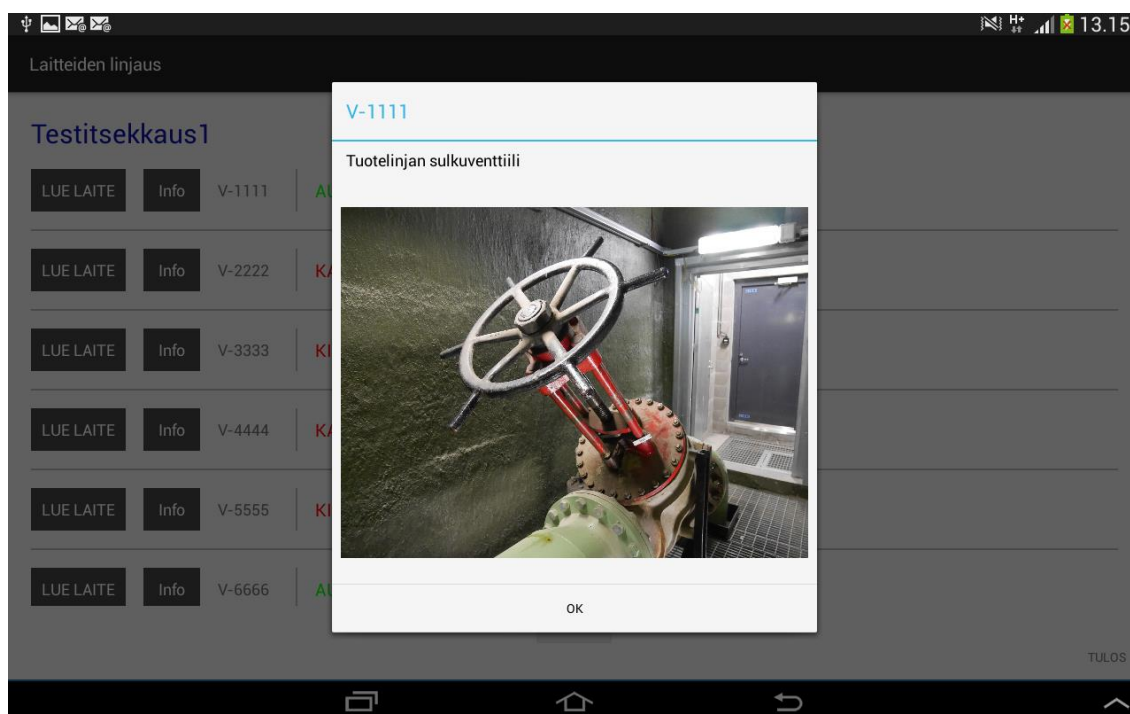
Kuva 14. Laitteiden linjauksen csv-tiedoston muoto

Ohjelma avaa luettavan listan, laskee csv-tiedoston rivi- ja rivikohtaisen kolumnimäärän ja tämän perusteella luo tarkastuslistan käytettävän näkymän (ks. Kuva 15.).



Kuva 15. Laitteiden linjaus – laitelista

Lista luetaan laite kerrallaan läpi. Mikäli laitteen sijainnissa tai tilassa on epäselvyyksiä, voi käyttäjä tarkistaa laitteen tiedot INFO-painiketta painamalla (ks. Kuva 16.).



Kuva 16. Laitteiden linjaus – Info-painikkeen laitenäkö

Kun laite on luettu, ohjelma pyytää seuraavaksi kommenttia. Kommentti ei ole pakollinen, joten sen voi jättää halutessaan tyhjäksi, mikäli kommentoitavaa laitteen tilasta, toimivuudesta tai muusta oleellisesta ei ole. Kaikki ulkoiset havainnot on kuitenkin hyvä kirjata listaan ja tarvittaessa ilmoittaa eteenpäin. Kun kommentti on kirjattu, se näkyy luetun laitteen perässä. Myös LUE LAITE -painike muuttuu OK:ksi ja kyseistä laitetta ei voi enää lukea. Jos vahingossa luetaan väärä laite, lista ilmoittaa kyseisen laitteen kohdalla selkeästi, että kyseessä on väärä laite. Laitteen voi lukea uudestaan, ja ohjelma hyväksyy tämän vasta, kun kyseessä on oikea laite.

Kun listan jokainen laite on käyty läpi ja luettu, voidaan tarkastus tallentaa. Tallennuksen yhteydessä raportista luodaan csv-tiedosto, joka tallentuu automaattisesti arkistokansioon. Raportin voi myös halutessaan tallentaa pdf-tiedostona tai tulostaa paperisena. Mikäli kesken tarkastuksen yritetään poistua tai palata edelliselle sivulle, ohjelma varmistaa käyttäjältä kysymällä, ettei poistuminen ole vahinko.

5.7.4 Kehitysideat

Laitteiden linjauksessa olisi tärkeää saada ohjelma toimimaan yhteistyössä automaation kanssa. Kun on tulossa toimenpide, johon vaaditaan laitteiden tarkastusta, voisi valvomonäytöltä valita käytettävän putkilinjan tai alueen. Automaatio lähettäisi suoraan oikean listan mobiililaitteelle, mitkä laitteet tulisi tarkistaa. Laitteiden operointia ei voisi suorittaa automaation valvomonäytöltä, mikäli tarvittavaa tarkastusta ei ole suoritettu. Kun tarkastus on tehty, lähettää mobiililaitte automaatiolle tiedon ja operointi voidaan aloittaa.

Listan ulkomuotoa voisi kehittää parempaan suuntaan, esimerkiksi vaihtoehtoinen näkymä, jossa lista näyttäisi aina yhden laitteen kerrallaan ruudussa ja sen luettua siirtyisi seuraavaan laitteeseen. PI-kaaviotyypinen näkymä voisi myös helpottaa havainnollistamaan laitteet paremmin kentällä.

Myös tiedonsiirrossa käytettävä csv-muoto voitaisiin päivittää monimuotoisempaan, esimerkiksi XML-muotoon, jolloin käytettäviä tietoja voitaisiin hyödyntää laajemmin.

6 Päätelmät

Työn tavoitteena oli suunnitella ja kehittää mallisovellus öljyterminaalien hallintajärjestelmän haluttujen osa-alueiden etäkäyttöön. Työssä paneuduttiin myös terminaalien työkentelytapojen nykytilanteeseen ja niiden kehittämiseen tehokkaammaksi ja turvallisemmaksi mobiililaitteen avulla. Valitettavasti mobiililaitteen käyttöä kentällä ei päästy työn aikana testaamaan, johtuen Androidin ja Windowsin välisen levyjaon yhteysongelmista (ks. Luku 5.3.2), sekä Ecomin tablettitietokoneen toimitusajoista. Kaikki työssä tehdyt sovellukset todettiin toimiviksi testiolosuhteissa. Kun ohjelman toimivuus on taattu kentällä, voidaan toimintoja ja sisältöä lähteä kehittämään laajemmaksi ja tehokkaammaksi. Kehitysideoita on työn aikana tullut paljon, ja suurin osa näistä tullaan ottamaan käyttöön (ks. Luvut 5.4.2; 5.5.3; 5.7.4.).

Työn aikaiset havainnot käytännön operoinnista kentällä auttoi paljon ideoimaan eri ominaisuuksia ja kehityskohteita mobiililaitteelle. Myös operaattoreilta kysytyt kehitysideat ja mielipiteet ohjasivat eteenpäin työssä. Uusia ideoita syntyi myös mobiililaitteelle, kuten polttoaineiden tiheys- ja määrälaskuri, sähköisten työluopakaavakkeiden täyttämisen ja laitteiston varastosaldon tarkistaminen.

Mobiililaitteen ja ohjelman saaminen testikäyttöön operaattorien kesken on seuraava askel insinööriyön jälkeen. Laitteen jalkauttaminen ja ottaminen osaksi päivittäisiä työruutiineja tulee tehdä hallitusti, sekä seuraten laitteen ja operaattorin välistä käyttötehokkuutta. Laitteelle järjestetään käyttökoulutus, jotta sen turvallinen ja oikeaoppinen käyttö olisi mahdollista. Mikäli huomataan, että mobiililaitteen käyttö heikentää työturvallisuutta tai -tehokkuutta, tulee sen käyttöominaisuuksia ja -tarkoituksia muokata uudelleen. Käyttökokemukset, kehitysideat, vikailmoitukset ja muu palaute tulee kerätä ja tämän perusteella kehittää laitteen käyttöä. Näin on mahdollista taata jatkuva kehitys työnteon tehostamisen ja turvallisuuden välillä, sillä kumpaakaan näistä ei tule korostaa toisen kustannuksella.

Lähteet

- 1 Android Studio Overview. Verkkodokumentti. Android. <<http://developer.android.com/tools/studio/index.html>> Luettu 25.8.2015.
- 2 AndSMB Features. 2014. Verkkodokumentti. Lysesoft. <<http://www.lysesoft.com/products/andsmb/>> Luettu 20.4.2016.
- 3 ATEX-laitedirektiivi. 2014. Direktiivi. EU-Komissio. <http://eur-lex.europa.eu/legal-content/fi/TXT/PDF/?uri=OJ:JOL_2014_096_R_0309_01&from=EN> 29.3.2014. Luettu 25.4.2016.
- 4 ATEX-olosuhdedirektiivi. 2000. Direktiivi. EU-Komissio. <<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2000:023:0057:0064:fi:PDF>>28.1.2000. Luettu 25.4.2016.
- 5 ATEX-opas – Räjähdyksvaarallisten tilojen turvallisuus. 2015. Verkkodokumentti. Tukes. <http://www.tukes.fi/Tiedostot/vaaralliset_aineet/esitteet_ja_opaat/ATEX_opas.pdf> Luettu 3.1.2016.
- 6 JCIFS-documentation. 2014 Verkkodokumentti. <<https://jcifs.samba.org/src/docs/api/>>Last Update 29.10.2014. Luettu 20.9.2015.
- 7 Microsoft NTLM. 2016. Verkkodokumentti. Microsoft. <[https://msdn.microsoft.com/en-us/library/windows/desktop/aa378749\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa378749(v=vs.85).aspx)> Luettu 29.4.2016.
- 8 Siemens TeleControl. 2015. Verkkodokumentti. Siemens. <<http://w3app.siemens.com/mcms/infocenter/dokumentcenter/sc/ic/Documentsu20Brochures/Brochure-Telecontrol-EN.pdf>> Luettu 28.4.2015.
- 9 Tiilikainen, Seppo. Manner, Jukka. 2013. Suomen automaatioverkkojen haavoittuvuus. Tutkimus. <<https://research.comnet.aalto.fi/public/Aalto-Shodan-Raportti-julkinen.pdf>> 21.3.2013. Luettu 13.4.2016.

Android -koodiesimerkki

Yksinkertainen viivakoodiluku-ohjelma luotuna Android Studiolla.

MalliLuokka.java

```
package com.example.admin.testaus;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

/**
 * Luodaan Java-luokka ja määritetään se aktiviteetin pohjaksi. Lisäksi
 * luokassa on painike, jonka painamisesta tapahtuu haluttu toiminto:
 */

public class MalliLuokka extends Activity implements View.OnClickListener {

    /**
     * Esitellään luokassa tarvittavat muuttujat:
     */

    Button btn;
    TextView ScanResults;
    int id;
    IntentIntegrator viivakoodiLuku;

    /**
     * Luodaan luokan ensimmäinen metodi, joka suorittaa koodin aktivi-
     * teetin käynnistyessä:
     */

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_malliluokka);

        /**
         * Asetetaan luokka aktiviteetin yhteyteen.
         * Seuraavaksi määritetään muuttujille toiminnot:
         */

        viivakoodiLuku = new IntentIntegrator(this);

        /**
         * Määritetään viivakoodiLuku-muuttujalle luokka
         */

        ScanResults = (TextView) findViewById(R.id.viivakoodiTulos);

        /**
         * Etsitään tekstinäkymälle id-tunnuksen perusteella vastaava
         * näkymä XML-aktiviteetista
         */
    }
}
```

```

        /**
        * Seuraavaksi etsitään painikkeelle vastine XML-aktiviteetista
        ja määritetään sille toiminto
        */
        btn = (Button) findViewById(R.id.viivakoodiLukubtn);
        btn.setOnClickListener(this);
    }

    @Override
    public void onClick (View v){
        id = v.getId();
        switch (v.getId()) {
            case R.id.viivakoodiLukubtn:
                viivakoodiLuku.initiateScan();
                /**
                * Toiminto on siis aktivoida viivakoodilukunäkymä
                */
                break;
        }
    }
}

/**
* Määritetään metodi, jossa käsitellään viivakoodista saatu luku-
tulos
*/
public void onActivityResult(int requestCode, int resultCode, Intent
intent) {
    IntentResult result = IntentIntegrator.parseActivityResult(re-
questCode, resultCode, intent);
    if (result != null) {
        final String contents = result.getContents();
        if (contents != null) {
            this.showDialog(R.string.result_succeeded);
            /**
            * Asetetaan saatu tulos aikaisemmin haettuun teks-
            tinäkymään
            */
            ScanResults.setText(contents);
        } else {
            this.showDialog(R.string.result_failed);
        }
    }
}
}

```

MalliLuokka.xml

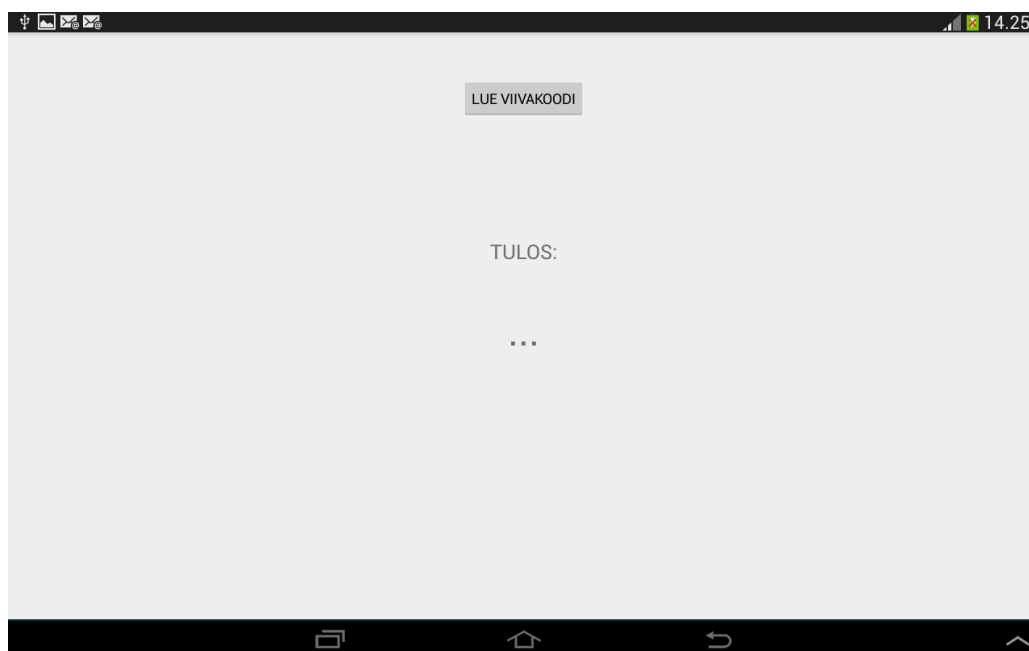
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"        android:layout_width="match_parent"
    android:layout_height="match_parent"        android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MalliLuokka">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="LUE VIIVAKOODI"
        android:id="@+id/viivakoodiLukubtn"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="42dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TULOS:"
        android:id="@+id/textView3"
        android:layout_marginTop="238dp"
        android:layout_centerHorizontal="true"
        android:textSize="25dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="..."
        android:id="@+id/viivakoodiTulos"
        android:layout_below="@+id/textView3"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="47dp"
        android:textSize="50dp" />

</RelativeLayout>
```

Sovellus QR-koodin lukemisen jälkeen:

