

KARELIA-AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutusohjelma

Eppu Mutanen

DYNAAMISTEN LOMAKKEIDEN LUOMINEN ANDROID-
ALUSTALLE

Opinnäytetyö
Huhtikuu 2017



OPINNÄYTETYÖ
Huhtikuu 2017
Tietojenkäsittelyn koulutusohjelma

Karjalankatu 3
80220 JOENSUU
+358 13 260 600

Tekijä(t)
Eppu Mutanen

Nimeke
Dynaamisten lomakkeiden luominen Android-alustalle

Toimeksiantaja
Collapick Company Oy

Tiivistelmä

Opinnäytetyön tavoitteena oli toteuttaa Android-mobiilialustalle sovellus, jonka avulla on mahdollista näyttää erilaisia mukautuvia lomakkeita sovelluksen käyttöliittymässä rajapinnasta saadun syötteen avulla. Tarkoituksena oli kehittää toiminnallisuus, jota on mahdollista hyödyntää mahdollisimman moniin eri tarkoituksiin, joissa mobiililomakkeen käytölle ilmenee tarve.

Käytännössä yhdellä toteutuksella haluttiin mahdollistaa erityisesti teollisuuden alan yritysten toiminnassa käytettävien lomakkeiden toteutus mahdollisimman tehokkaasti. Teollisissa yrityksissä lomakkeiden hyödyntäminen on mahdollista tuotannonohjauksen osana ja lomakkeet tarjoavat myös tuotantoa tehostavia ratkaisuja. Opinnäytetyön raportissa on esitelty toteutuksen hyödyntämistä muun muassa teollisuuden alan häiriöiden seurannan, turvallisuushavaintojen kirjauksen sekä tarkastuspöytäkirjan toteutuksen osalta.

Opinnäytetyön tuloksena syntyi toimiva kokonaisuus, joka sisälsi tarvittavat perustoiminnallisuudet ja joka on helposti laajennettavissa. Varsinaisen sovelluksen toiminnallisuudet toteutettiin erilliseen moduuliin, joka on myös helppo ottaa osaksi mitä tahansa muuta Android-projektia. Jatkokehitystä ajatellen toteutuksessa hyödynnetyn Alpaca-ohjelmointikirjaston JSON-syötettä hyödyntämällä voidaan toteuttaa lähes mitä tahansa web-lomakkeiden sisältämiä kenttiä osaksi toteutettua sovellusta. Toteutusta on jatkossa tarkoitus hyödyntää mahdollisimman monipuolisesti erilaisissa toimeksiantajan projekteissa, joissa edellytetään lomakkeiden näyttämistä mobiilisovelluksen käyttöliittymässä.

Kieli
suomi

Sivuja 31

Asiasanat
dynaamiset lomakkeet, mobiililaitteet, Android, ohjelmointirajapinta



THESIS
April 2017
Degree Programme In Business Information Technology

Karjalankatu 3
80220 JOENSUU
FINLAND
Tel. +358 13 260 600

Author(s)
Eppu Mutanen

Title
Creating Dynamic Forms for Android Platform

Commissioned by
Collapick Company Oy

Abstract

The main purpose of this thesis was to develop an application for Android platform that makes it possible to display a variety of forms based on the input received from the application programming interface in application's user interface. The aim was to develop a functionality which can be utilized in maximum variety of purposes when there's need to use mobile forms. In practice, a single implementation was needed to make it possible to create different forms which are used in operation of industrial companies as efficiently as possible. The report of this thesis shows the utilization of the implementation for monitoring of disturbances in industrial sector, logging safety detections and creating inspection reports.

The JSON feed used by the Alpaca programming library was used for processing the structure of the forms. The feed used by Alpaca contains a wide variety of form fields and that feed can be utilized to implement nearly any fields that are used in web forms as part of the implemented application in further development.

The result of this thesis was a well-functioning application which contained the required basic functionalities and is easily extendable for new functionalities. The actual functionalities of the application were implemented in a separate module which is also easy to integrate into any other Android project. In the future the implementation is intended to be used in a wide range of client's different projects which require displaying mobile forms in application's user interface.

Language
finnish

Pages 31

Keywords
dynamic forms, mobile devices, Android, Application Programming Interface

Sisältö

1	Johdanto.....	5
2	Tuotannonohjaus teollisuudessa.....	6
3	Tuotannon tehostaminen.....	7
3.1	Toiminnanohjausjärjestelmät	7
3.2	Lean-ajattelu	9
4	Mobiililaitteiden hyödyntäminen osana tuotantoa.....	10
4.1	Työn tilan ja häiriöiden kirjaus.....	11
4.2	Turvallisuuden seuranta.....	12
5	Dynaamisten lomakkeiden hyödyntäminen mobiililaitteilla.....	13
5.1	Dynaamiset lomakkeet.....	13
5.2	Dynaamisten lomakkeiden toteutus Android-alustalle	14
5.2.1	Sovellusarkkitehtuuri	15
5.2.2	Alpaca	17
5.2.3	Lomakkeen sisällön hakeminen rajapinnasta	19
5.2.4	Näkymien muodostaminen käyttöliittymään.....	20
6	Toteutuksen hyödyntäminen teollisuudessa	23
6.1	Tarkastuspöytäkirja	23
6.2	Turvallisuushavainnot	24
6.3	Häiriöilmoitukset.....	25
7	Tulokset.....	27
8	Jatkokehitysmahdollisuudet	27
9	Pohdinta	29
	Lähteet.....	31

1 Johdanto

Opinnäytetyöni tavoitteena oli toteuttaa toimeksiantajan ja erityisesti teollisuuden alan tarpeisiin vastaava mobiilisovellus, jonka avulla mobiililaitteella pystytään näyttämään käyttäjän tarpeisiin mukautuva lomake tarvittavine sisältöineen. Käyttäjän tarpeisiin mukautuvilla lomakkeilla tässä tapauksessa tarkoitetaan sitä, ettei sovellukseen ole ennalta määritetty käyttöliittymässä näytettävää sisältöä, vaan lomakkeiden sisällön hallinta on toteutettu täysin sovellukseen määritetystä rajapinnasta saadun syötteen avulla.

Mukautuvan sisällön avulla toteutuksen haluttiin toimivan monipuolisesti loppukäyttäjien yritysten tuotantoa tehostavana ratkaisuna, jolla pystytään karsimaan tuottamatonta työtä. Tuotannon tehostamisen lisäksi tavoitteena oli vähentää tai parhaimmillaan jopa poistaa tarve paperisille lomakkeille, joita teollisissa yrityksissä paljon käytetään. Toteutuksesta oli tarkoituksena saada monipuolinen, jota toimeksiantajan olisi mahdollista hyödyntää useissa eri projekteissa.

Opinnäytetyöni toimeksiantajana oli Collapick Company Oy, joka toteuttaa erilaisia mobiili- sekä web-sovelluksia asiakkaiden toiminnan tehostamiseksi ja liiketoiminnallisten haasteiden ratkaisemiseksi. Yritys toteuttaa laajasti myös teollisuuden alan tarpeita vastaavia ratkaisuja, joiden avulla mobiililaitteet voidaan tuoda osaksi teollisuuden prosesseja ja järjestelmiä. (Collapick Company Oy 2017)

Opinnäytetyö oli toiminnallinen ja opinnäytetyön raportin tarkoituksena on kuvata toteutettuja toiminnallisuuksia sekä avata opinnäytetyön aiheen taustoja. Toteutuksessa käytettyjen työkalujen ja menetelmien käytöstä on kerrottu raportissa tietoa hyvin yleisellä tasolla. Toteutuksen toiminnallisuuden kannalta avainasemassa olevien työkalujen ja erityisesti ohjelmointikirjastojen toimintaa on sen sijaan raportoitu tarkemmin.

2 Tuotannonohjaus teollisuudessa

Tuotannonohjauksen kuulee usein olevan tärkeässä roolissa osana teollisuuden alan yritysten toimintaa. Tuotannonohjauksella tarkoitetaan menetelmää, jolla pyritään ohjaamaan tuotantoa siten, että pystytään varmistamaan tilattujen tuotteiden valmistamiseen vaadittava laatu, määrä sekä toimitusaika. Tuotannonohjaukseen kuuluvia asioita ovat yleensä tuotannon ajoitus, varastotilanteen valvonta ja tuotannon kapasiteetin mahdollisimman tehokas hyödyntäminen. (Mantere 2014, 10.)

Tuotannonohjauksessa eri osa-alueiden seuranta voi kuulua yrityksen koosta riippuen yhdelle tai useammalle tuotannonohjaajalle. Tuotannonohjauksesta vastaavien henkilöiden tärkeimpänä tehtävänä on varmistaa tuotannonohjaukseen liittyvien asioiden sujuvuus. Tiedon avulla tuotannonohjaajat pystyvät hallitsemaan tuotannon ajoitusta tarkasti, sillä he tietävät tarkasti valmistusprosessin vaiheet, joissa valmistettava tuote siirtyy tuotannossa eteenpäin. On erittäin tärkeää, että tuotannonohjaajat pystyvät tuotannon tilanteen mukaan reagoimaan nopeasti varastotilanteeseen, jolloin esimerkiksi tuotannon kiihtyessä varastot eivät pääse tyhjenemään missään vaiheessa. Myös päinvastaisessa tilanteessa tuotannonohjaajan on huolehdittava siitä, että varastoihin ei turhaan tilata liian suuria määriä tuotteita, mikäli tuotanto on hiljentynyt. (Teknologiateollisuus ry 2010.)

Tuotannonohjaajien on pystyttävä hallitsemaan myös yllättävät poikkeustilanteet. Henkilöiden on varauduttava siihen, että tuotannossa voi tapahtua normaaliin toimintaan vaikuttavia asioita, jolloin on pystyttävä reagoimaan tilanteeseen ja huolehdittava siitä, että valmistettavat tuotteet saadaan poikkeuksista huolimatta toimitettua asiakkaalle sovitussa aikataulussa. Tällaisia poikkeustapauksia ovat mm. työntekijöiden sairastumiset, tuotantolaitteen rikkoutumiset tai valmistamiseen tarvittavien komponenttien toimitusten viivästykset. Kun poikkeusten aiheuttamat muutokset tuotantoon on saatu järjestettyä, tulee tuotannonohjaajan selvittää poikkeuksen varsinaiset syyt ja varmistaa, ettei vastaavaa tilannetta pääsisi syntymään uudestaan. (Teknologiateollisuus ry 2010.)

Tuotannonohjauksen suurimpia haasteita teollisuudessa on tuotannon rakenteen monimutkaisuus. Tuotetta valmistettaessa valmistettavat osat voivat käydä valmistusprosessin aikana useissa eri valmistuspisteissä ja osalla voi olla useita vaihtoehtoisia valmistustapoja, joiden mukaan valmistusreitit määräytyvät. Tämän kaltainen valmistusprosessi aiheuttaa sen, että tuotannonohjauksesta tulee vaikeaa ja siksi toimintaa on jaettu tuotantotiloissa usein eri linjoihin. Yksittäiset linjat tekevät jonkin yksittäisen tuotteen alusta loppuun, jolloin tuotannon ohjaus on paremmin hallittavissa ja koko tuotantokapasiteetti saadaan hyödynnettyä paremmin. Tällaisessa tuotantolinjassa on usein kuitenkin kohta, joka toimii niin sanotusti pullonkaulana, eli rajoittaa tuotannon toimintaa ja määrää tuotantolinjan kapasiteetin. Tuotannonohjauksessa tulee keskittyä näihin rajoitteisiin ja huolehtia siitä, että tällaista tuotantolinjan osaa kuormitetaan koko ajan mahdollisimman tehokkaasti. (Wikipedia 2017.)

3 Tuotannon tehostaminen

3.1 Toiminnanohjausjärjestelmät

Tuotannonohjausta voidaan hallita pienissä yrityksissä jopa ilman erillisiä apuvälineitä, mutta yrityksen koon kasvaessa tuotannon kapasiteetin, ajoituksen sekä varastotilanteen seuranta on haastavaa, ellei jopa lähes mahdotonta, koska hallittavia asioita on paljon. Suuressa osassa teollisuuden alan yrityksistä, joissa tuotannonohjauksen hoitaminen käsin on mahdotonta, käytetään tuotannonohjauksen apuna erillisiä toiminnanohjausjärjestelmiä, joita kutsutaan ERP (Enterprise Resource Planning) -järjestelmiksi. Toiminnanohjausjärjestelmän tarkoituksena on helpottaa tuotannonohjaukseen liittyvien asioiden seurantaa ja hallintaa ja parantaa yrityksen tehokkuutta. Tuotannonohjauksen lisäksi toiminnanohjausjärjestelmät sisältävät usein jopa tarvittavat työkalut muun muassa kirjanpitoa ja laskutusta varten, jolloin suurin osa yrityksen tarvitsemista sähköisistä palveluista on käytettävissä yhden järjestelmän kautta. (Rouse, Brislen, Krishnakumar, O'Donnell 2016.)

Alun perin toiminnanohjausjärjestelmät kehitettiin avuksi yritysten varaston- ja materiaalinhallintaan. Näitä järjestelmiä kutsuttiin yleisesti MRP (Material Requirements Planning) -järjestelmiksi. MRP-järjestelmien avulla pystyttiin laskemaan tuotantoaikataulua tilausten ja myyntiennusteen pohjalta. Tuotantoaikataulun ja osaluettelon avulla pystyttiin laskemaan materiaalien tarve tuotantoa varten ja vertaamalla tarvetta varastotilanteeseen saatiin tietoon hankittavien materiaalien määrä. 1980-luvulla MRP-järjestelmän päälle kehitettiin lisää ominaisuuksia muun muassa lisäämällä kustannuslaskennan sekä taloushallinnon ominaisuuksia järjestelmään. Ominaisuuksien laajetessa järjestelmiä alettiin kutsua MRP II (Manufacturing Resource Planning) -järjestelmiksi. Pian 1990-luvun alussa MRP II -järjestelmä laajeni ERP-toiminnanohjausjärjestelmäksi, joka kattoi tekniikkaan, talouteen, henkilöstöresursseihin sekä projektijohtamiseen tarvittavat toiminnallisuudet. (Mattsson 2013, 1–2.)

Toiminnanohjausjärjestelmät ovat olleet aina suuria investointeja yrityksille, ja ennen ainoastaan suuremmilla yrityksillä oli varaa toiminnanohjausjärjestelmän käyttöönottoon. Nykyaikaiset toiminnanohjausjärjestelmät ovat kuitenkin mahdollistaneet räätälöitävyyden yrityksen tarpeiden mukaan, sillä ne sisältävät usein toiminnallisuuksia erillisinä moduuleina, joita voidaan lisätä tai poistaa järjestelmästä. Modulaarisuuden avulla yritys saa järjestelmään tarvitsemansa toiminnallisuudet, eikä sen tarvitse maksaa toiminnallisuuksista, joita se ei tule tarvitsemaan. Modulaarisuus on mahdollistanut myös sen, että pienemmillään yrityksillä on ollut vara ottaa toiminnanohjausjärjestelmä osaksi yrityksen jokapäiväistä toimintaa. (Tietoyhteiskunnan kehittämiskeskus ry 2005.)

Yksi haastavimmista ja suurimpia investointeja vaativista vaiheista on toiminnanohjausjärjestelmän käyttöönotto. Käyttöönottovaihe on toimenpide, joka vaatii niin asiakkaalta kuin järjestelmän toimittajaltakin suurta työpanosta. Järjestelmän ottaminen mukaan yrityksen toiminnanohjaukseen vaatii yleensä jonkin verran muutoksia itse tuotantoprosessiin, minkä vuoksi järjestelmän tarjoamia kaikkia ominaisuuksia ei voida ottaa heti käyttöön ilman, että tuotannossa esiintyy häiriöitä. Tärkeää on varmistaa, että tuotanto ei häiriinny järjestelmän käyttöönotosta, vaan se pystytään ottamaan käyttöön vaiheittain tuotannon eri

osa-alueille sitä mukaa, kun tuotantoprosessi sen mahdollistaa. Loppujen lopuksi tavoitteena kuitenkin on, että järjestelmä mahdollistaisi tuotannon tehostamisen sekä tehokkaamman seurannan ja niin sanotusti maksaisi itseään takaisin. (Mattsson 2013, 38–39.)

3.2 Lean-ajattelu

Lean-ajattelu käsitteenä on lähtöisin vuonna 1990 julkaistusta autoteollisuusyritysten kilpailukykyä selvittäneestä tutkimuksesta, jossa havaittiin menestyneiden yritysten toiminnassa useita yhteneviä piirteitä. Tutkimuksessa huomattiin, että keskittymällä vain asiakkaalle lisäarvoa tuottavaan toimintaan voidaan säästää merkittävästi aikaa sekä syntyviä kustannuksia. Yksi keskeinen tausta Lean-ajattelussa on autovalmistaja Toyotan kehittämä tapa, jolla se erottui kilpailijoistaan valmistamalla kilpailijoitaan laadukkaampia tuotteita. Lyhyesti kuvattuna Lean-periaatteiden kehitys perustui alun perin kolmeen alueeseen, jotka olivat lisäarvoa tuottamattoman työn, tuotannon epätasaisuuden sekä ylikuormituksen vähentäminen. (Tyrväinen 2011, 12.)

Skaggsin (2010) mukaan Lean-ajattelu perustuu eri menetelmiin, joilla mahdollisesta tuotannon tehostaminen. Yksi näistä menetelmistä on 5S-järjestelmä, joka keskittyy työmenetelmien standardointiin siten, että se kasvattaa työn tuotavuutta vähentämällä tuottamatonta toimintaa ja parantamalla laatua sekä turvallisuutta. Yrityksen toimitilojen turvallisuus, sekä valmistettujen tuotteiden laatu taas vaikuttavat olennaisesti yrityksen kilpailukykyyn vallitsevilla markkinoilla.

Käytännössä 5S-järjestelmään kuuluu viisi eri osa-aluetta, jotka ovat sorteeraus (sort), systematisointi (set in order), siivous (shine), standardisointi (standardize) sekä seuranta (sustain). Sorteerauksessa työpisteeltä poistetaan tarpeettomat tavarat, jolloin työpisteelle vapautetaan tilaa ja poistetaan mahdollisesti rikki kootuneita tai tarpeettomia työkaluja, jotka ovat työpisteellä vain siltä varalta, että niitä vielä joku joskus tarvitsisi. Systematisoinnissa pyritään löytämään tehokkaita varastointimenetelmiä, joita voivat olla esimerkiksi alueiden selkeät rajaukset, selkeät ja tyhjät käytävät sekä eri asioiden merkitseminen selkein paik-

kamerkinnoin ja värikoodein. Siivoukseen kuuluu työpisteen päivittäinen siivoaminen. Standardisoinnin tarkoituksena on standardoida työpaikan toimivimmat käytännöt yhdessä työntekijöiden kanssa. Näitä standardeja voivat olla muun muassa työpisteeseen kuuluvat työkalut sekä työpisteen siivousaikataulu. Viimeisenä 5S-järjestelmään kuuluu seuranta, jonka avulla pidetään huolta, että sovittuja menetelmiä noudatetaan jatkuvasti. (Skaggs 2010.)

Toinen Lean-ajatteluun perustuva menetelmä on nimeltään Just in Time -ajattelu, joka on erityisesti teollisuudessa käytetty varastonhallinta- ja tuotannonohjausstrategia, jonka tarkoituksena on parantaa tehokkuutta tuotantoprosessin kokonaisuudessa. Just in Time -ajattelun perusideana on toimittaa ainoastaan niitä tuotteita, joita asiakas tarvitsee vasta silloin, kun asiakkaalla on tuotteille tarve. Lisäksi tuotteita toimitetaan ainoastaan sen verran kuin niitä tarvitaan. (Tyrväinen 2011, 16.)

Imuohjaus kuuluu myös Lean-ajattelun toimintatapoihin. Imuohjauksessa hankinta- ja valmistusprosessi aloitetaan vasta siinä vaiheessa, kun asiakas tekee tilauksen. Imuohjauksessa materiaalien ohjaus on suoraan verrannollinen kuluutukseen. Tällainen malli sopii hyvin suurille materiaalivirroille, jolloin tuotannonohjausta ei tarvitse seurata päivittäin, vaan tilannetta voidaan tarkastella harvemmin. Tämä vähentää tuotannonohjauksen kuormitusta. (Tyrväinen 2011, 16.)

4 Mobiililaitteiden hyödyntäminen osana tuotantoa

Teollisuuden alan yritykset hyödyntävät nykyisin laajasti erilaisia sähköisiä järjestelmiä osana toimintaansa, mikä mahdollistaa tuotannon tehokkaan seurannan ja ohjauksen. Myös varastojen hallintaa ja taloushallintoa hoidetaan tehokkaasti sähköisten järjestelmien avulla. Lean-ajattelun mukaan työstä tulisi vähentää tai jopa parhaimmassa tapauksessa kokonaan poistaa tuottamaton toiminta ja parantaa työympäristön turvallisuutta, mutta useat teollisuuden alan

yrietykset eivät vielä ole hyödyntäneet mobiilin tuomia mahdollisuuksia näiden asioiden parantamiseksi niin hyvin, kuin se olisi jo teknisesti mahdollista.

4.1 Työn tilan ja häiriöiden kirjaus

Yrityksien toiminnassa ne henkilöt, jotka työskentelevät tuotantotiloissa tuotteen valmistuksen parissa eli niin sanotusti lattiatasolla, hyödyntävät nykyteknologian mahdollistamia ratkaisuja usein päivittäisessä työssään. Käytännössä tämä tarkoittaa sitä, että esimerkiksi tuotannossa hitsauspisteillä työskentelevät henkilöt käyvät tuotantotiloissa sijaitsevilla päätelaitteilla valitsemassa työn, jonka he ottavat seuraavana toteutettavaksi työkseen. Kirjauksen jälkeen järjestelmään tallentuu tieto siitä, että työ on aloitettu ja työntekijä aloittaa työn varsinaisen toteutuksen. Mikäli työntekijä huomaa kuitenkin valmistuksen aikana puutteita tai häiriöitä, jotka estävät työn toteutuksen viemisen eteenpäin, käy hän kirjaamassa päätelaitteelle tiedon työn keskeytyksestä. Keskeytyksen yhteyteen kirjataan kaikki oleellinen tieto tilanteesta ja mahdollisesti liitetään tietojen yhteyteen kuva selventämään ongelmaa. Tässä vaiheessa työntekijän tulee siirtää kuva päätelaitteelle ja tätä kautta siirtää kuva järjestelmään. Tämän jälkeen tieto keskeytyksestä siirtyy eteenpäin ja tilanteen selvittämiseksi tehdään tarvittavia toimenpiteitä mahdollisesti yhdessä esimiehen kanssa. Kun työntekijä lopulta saa oman työvaiheensa valmiiksi, käy hän jälleen päätelaitteen luona kirjaamassa työn valmistuneeksi, jolloin työ siirtyy tuotannossa seuraavaan vaiheeseen.

Järjestelmien tuomat hyödyt mahdollistavat sen, että kaikki yrityksen työntekijät tietävät tarkasti missä vaiheessa työ on kyseisellä hetkellä menossa. Myös tieto häiriöistä siirtyy eteenpäin sähköisesti, eikä työntekijän tarvitse selvittää havaittua ongelmaa kaikille tarvittaville henkilöille erikseen.

Edellä mainitun kaltaisessa tilanteessa on kuitenkin vaihteita, joita pystyttäisiin hyödyntämään paljon tehokkaammin mobiililaitteiden avulla ja samalla vähentämään tuottamattoman työn määrää. Mikäli jokaisella tuotannossa työskentelevällä henkilöillä olisi käytössään henkilökohtainen mobiililaitte, välttyttäisiin yhteisen päätelaitteen tuomilta ongelmilta. Ratkaisussa, jossa tuotantotiloissa on

käytössä yhteiset päätelaitteet, työntekijän tulee siirtyä työpisteeltään päätelaitteen luokse ja tämän jälkeen kirjautua henkilökohtaisilla tunnuksillaan sisään järjestelmään. Mobiililaite taas mahdollistaa sen, ettei työntekijän tarvitse siirtyä työpisteeltään mihinkään, vaan laitteen avulla työn tilan kirjaus onnistuu suoraan työpisteeltä käsin. Myöskään erillistä kirjautumista ei tarvita jokaisella käyttökerralla, sillä parhaimmassa tilanteessa yhdellä mobiililaitteella on pääsääntöisesti vain yksi käyttäjä.

Häiriöiden kirjauksessa mobiililaitteiden tuoma hyöty korostuu entisestään. Henkilökohtaisen mobiililaitteen avulla työntekijällä on mahdollisuus kirjata havaitsemansa häiriöt suoraan häiriön lähteeltä, eli esimerkiksi suoraan viallisen tai puutteellisen tuotteen vierestä. Näin työntekijän on helpompi kirjata asioita ylös järjestelmään, kun hän pystyy kuvailemaan häiriön syyn tarkemmin. Usein kuitenkin kirjausten yhteyteen liitetty kuva havainnollistaa ulkopuoliselle henkilölle tehokkaimmin, mistä häiriössä oikeasti on kyse. Käytettäessä yhteistä päätelaitetta ongelmaksi muodostuu aiemmin mainittujen ongelmien lisäksi se, että työntekijän tulee siirtää ottamansa kuva päätelaitteelle, minkä jälkeen on vasta mahdollista lähettää kuva päätelaitteesta järjestelmään. Kyseinen toimenpide vaatii monta vaihetta ja vie työntekijältä huomattavasti aikaa. Mobiililaite mahdollistaa sen, että työntekijä voi ottaa häiriön aiheuttajasta kuvan ja lähettää sen suoraan järjestelmään ilman erillisiä välivaiheita.

4.2 Turvallisuuden seuranta

Turvallisuuden parantaminen on myös yksi Lean-ajattelun avaintekijöistä ja tärkeässä osassa yrityksen toimintaa. Se on asia, jossa varmasti jokainen yritys pyrkii olemaan mahdollisimman tehokas, jotta turhilta tuotannon keskeytyksiltä pystytään välttymään.

Yleinen tapa turvallisuuden seurantaan on havainnoida tuotantotiloissa näkyviä turvallisuuteen liittyviä asioita järjestämällä turvallisuuskävelyitä. Turvallisuuskävelyihin osallistuvat henkilöt kiertävät tuotantotiloissa ja pyrkivät havaitsemaan tiloissa piilevät riskit sekä vaaraa aiheuttavat tekijät. Turvallisuuskävelyä suorit-

tavalla henkilöllä on mukanaan usein paperinen tarkastuslista, jolle on olemassa valmis lomakepohja. Henkilö kirjaa havaitsemansa asiat lomakkeeseen ja turvallisuuskävelyn päätyttyä hän syöttää lomakkeeseen kirjatut tiedot sähköiseen järjestelmään päätelaitteeltaan. Kyseisessä tilanteessa ongelmalliseksi muodostuu se, että turvallisuuskävelyn suorittanut henkilö joutuu kirjoittamaan jo tekemänsä huomiot uudestaan sähköiseen järjestelmään. Saman asian kirjaus useaan kertaan luo turhaa työtä, joka olisi mahdollista karsia pois mobiililaitteiden avulla.

Mobiililaitte mahdollistaa turvallisuuskävelyn lomakkeen täytön sähköisesti, mikäli turvallisuuskävelyä suorittavalla henkilöllä on laite mukanaan havaintoja tehdessä. Näin havainnoituja asioita ei turvallisuuskävelyn jälkeen tarvitse kirjata enää mihinkään, vaan ne ovat suoraan järjestelmään tallennettuna ja ne löytyvät järjestelmästä helposti myös myöhempiä tarkastelua varten. Lomakkeelle syötettyjen kirjausten lisäksi mobiililaitte luo myös mahdollisuuden tallentaa havaittuja huomioita tarkentavasti kuvien avulla suoraan järjestelmään.

Niin turvallisuuden seurannassa kuin aiemmin mainitussa häiriöiden kirjauksesakin tarvittavat toiminnallisuudet ovat hyvin pitkälle samanlaisia. Tämän ansiosta molempiin tilanteisiin pystyttäisiin hyödyntämään samanlaista ratkaisua, jossa mobiililaitteen lomakkeen sisältöä olisi mahdollista hallita tarpeen mukaan.

5 Dynaamisten lomakkeiden hyödyntäminen mobiililaitteilla

5.1 Dynaamiset lomakkeet

Dynaamisilla lomakkeilla tarkoitetaan opinnäytetyössäni käyttäjän tarpeiden mukaan mukautuvia mobiililaitteilla käytettäviä lomakkeita. Toteuttamani sovelluksen tarkoituksena on palvella laajasti erilaisia käyttötarkoituksia riippumatta siitä, millaista sisältöä käyttäjät tarvitsevat mobiililaitteella käytettävälle lomakkeelle. Toteutus hyödyntää yhteyttä ulkoisiin ohjelmointirajapintoihin, joista lo-

makkeen sisältö haetaan. Rajapinnasta haetusta sisällöstä sovellus lukee lomakkeen rakenteen ja muodostaa käyttöliittymään tarvittavat lomakekentät sisältöineen. Toteutuksen toiminnallisuuksien teknistä toteutusta käsitellään tarkemmin kappaleessa 5.2.

Jo ennen opinnäytetyön toimeksiannon saamista toimeksiantaja oli etsinyt tietoa opinnäytetyön toteutusta vastaavista ratkaisuista, mutta valmiita täysin vastaavia toiminnallisuuksia sisältäviä toteutuksia ei mobiilialustoille ollut löytynyt. Tästä syntyi tarve saada käyttöön toteutus, jonka avulla vaaditut toiminnallisuudet olisi mahdollista toteuttaa. Opinnäytetyön prosessin aikana selvittäessäni aiheen taustoja en myöskään itse löytänyt vastaavia jo valmiita toteutuksia. Tämä vahvisti sen, että opinnäytetyön toteutukselle oli toimeksiantajalla oikea tarve, eikä valmiiden ratkaisujen hyödyntäminen ollut mahdollista tarvittavien toiminnallisuuksien saavuttamiseksi.

5.2 Dynaamisten lomakkeiden toteutus Android-alustalle

Opinnäytetyöni tekninen toteutus on tehty Android-alustalle, joka on ollut viime vuosina hallitsevassa markkina-asemassa oleva mobiililaitteiden käyttöjärjestelmä. International Data Corporationin (2016) laatiman tutkimuksen mukaan Androidin osuus toimitettujen älypuhelimien käyttöjärjestelmistä oli vuoden 2016 kolmannella vuosineljänneksellä yhteensä 86,8 prosenttia. Samassa tutkimuksessa käy ilmi Android-käyttöjärjestelmän markkinaosuuden kasvaneen 7,2 prosenttia pelkästään kolmen vuosineljänneksen aikana vuoden 2015 viimeisestä neljänneksestä, jolloin Androidin osuus oli 79,6 prosenttia kaikista toimitetuista laitteista.

Lukemat vahvistavat myös sen, että Android on sovelluskehittäjille alusta, jonka avulla on mahdollista saavuttaa erittäin suuri käyttäjäkunta. Alusta tarjoaa myös sovelluskehittäjille lukuisia mahdollisuuksia, sillä sen suosion myötä avoimen lähdekoodin Android-kirjastojen tarjonta on kattava. Pelkästään suositusta GitHub-palvelusta löytyy tällä hetkellä 8 634 tulosta haettaessa Java-ohjelmointikielellä toteutettuja Git-versionhallintaa käyttäviä ohjelmointikirjastoja

hakusanalla ”android library”. Git-termillä tarkoitetaan avoimen lähdekoodin versionhallintajärjestelmää, joka mahdollistaa ohjelmistokehityksessä usealle henkilölle yhtäaikaisen projektitiedostojen muokkaamisen järjestelmän ylläpitäessä tietoa tiedostoissa tapahtuvista muutoksista. Opinnäytetyön toteutuksessa hyödynnän useita valmiita ohjelmointikirjastoja, jotka hyödyntävät Git-versionhallintaa. Valmiiden kirjastojen avulla erilaisia toiminnallisuuksia voidaan toteuttaa nopeasti ja helposti.

Androidin vahva markkina-asema tarkoittaa myös sitä, että Android-käyttöjärjestelmän sisältäviä laitteita on lukuisia erilaisia ja erilaisiin käyttötarkoituksiin suunniteltuja malleja on myös alustalle helposti saatavilla. Tämä mahdollistaa sen, että myös teollisuuden alan tarpeisiin on saatavilla erilaisia malleja, jotka soveltuvat hyvin käytettäväksi esimerkiksi varsinaisissa tuotantotiloissa. Tuotantotiloissa ongelmaksi koituu yleensä pöly ja lika, joka vahingoittaa helposti normaaleja mobiililaitteita. Tämän lisäksi laitteisiin voi kohdistua erilaisia iskuja sekä kolhuja riippuen käyttäjän tavasta käsitellä laitetta työpisteellään. Useat älypuhelinvalmistajat tarjoavat nykyisin jo yleiseen kuluttajakäyttöön Android-laitteita, joissa on valmiina pöly- ja roiskesuojaus ja jotka ovat iskunkestäviä. Kyseiset älypuhelimet ovat usein samanhintaisia kuin normaalit älypuhelimet.

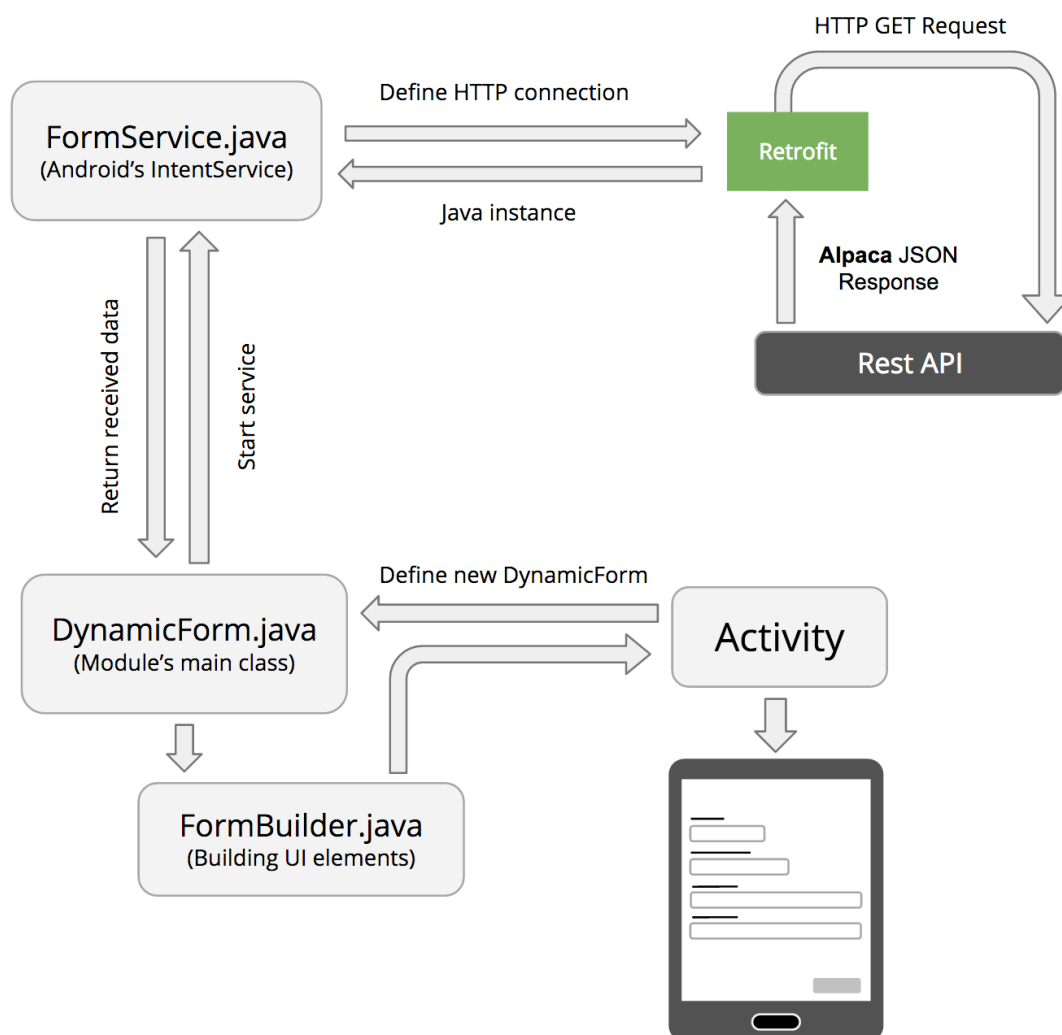
Esimerkkinä erikoiskäyttöön suunnatusta mallista otin Samsungin valmistaman Xcover 3 -älypuhelimien, joka on suunniteltu nimenomaan kestävämpään käyttöä. Samsung (2017) ilmoittaa tuotteen virallisella tuotesivulla laitteen olevan pöly- sekä vesitiivis ja kestävä jopa -20 asteen kylmyyttä. Tämän lisäksi se on suojattu myös iskuilta. Laitteen hinta verkosta tilattaessa on halvimmillaan 163 euroa, joka on jopa yleisesti ottaen varsin edullinen hinta älypuhelimelle. Laitteen hintatietojen etsimiseksi käytin hintavertailu.fi-palvelua, joka etsii tällä hetkellä halutun laitteen hintatietoja yhteensä 175:stä eri verkkokaupasta.

5.2.1 Sovellusarkkitehtuuri

Opinnäytetyön toteutuksessa käytin ohjelmointiympäristönä Googlen kehittämää Android Studiota, joka on Androidin virallinen ohjelmointiympäristö. Ohjelmointiympäristö valikoitui toimeksiantajan käyttämien ohjelmointiympäristöjen perusteella. En myöskään nähnyt mitään perusteltua syytä vaihtaa ohjelmointiympäristöä, sillä olin itse käyttänyt paljon Android Studio –ympäristöä. Android Studion lisäksi Android-projektien toteutuksessa on mahdollista hyödyntää myös muita ohjelmointiympäristöjä, joista esimerkkeinä ovat muun muassa Xamarinin sekä Eclipse. Näistä jälkimmäinen vaatii erillisen Android Development Tools –lisäosan kehitettäessä Android-sovelluksia. On olemassa myös erilaisia työkaluja, joiden avulla voidaan toteuttaa web-sovelluksia ja työkalut kääntävät sovelluksen toimivaksi erilaisilla mobiililaitteilla. Omien havaintojeni pohjalta kuitenkin Android Studio tarjoaa kaikkein parhaimman valikoiman valmiita toimintoja ja työkaluja, joiden avulla Android-projektien toteutus onnistuu tehokkaimmin.

Sovelluksen varsinainen ydintoteutus on toteutettu Androidin kirjastomoduulina, koska ajatuksena oli alusta asti toteuttaa helposti käyttöönotettava ja laajennettava toteutus. Kyseinen moduuli sisältää kaiken toiminnallisuuden liittyen lomakkeen tietojen hakuun rajapinnasta, lomakkeen sisällön käsittelyyn, lomakkeen näyttämiseen käyttöliittymässä sekä käyttäjän syöttämien tietojen käsittelyyn. Kirjastomoduulin tarkoituksena on, että toteuttamani toiminnallisuudet voidaan ottaa mukaan mihin tahansa muuhun Android-projektiin helposti sisällyttämällä moduuli osaksi haluttua projektia. Käyttämällä apuna moduuliin toteuttamani Java-ohjelmointikielen luokkia voidaan moduulille määrittää kaikki tarvittavat tiedot lomakkeen muodostamista varten. Pakollisena tietona moduuli tarvitsee tiedon URL-osoitteesta, josta lomakkeen sisältö haetaan, sekä käyttöliittymäelementin, jonka sisään lomake halutaan esille.

Sovelluksen rakennekaavion avulla (kuvio 1) esitetään sovelluksen ohjelmakoodin rakennetta ydintoiminnallisuuksien osalta. Yksittäiset ohjelmointiluokat, joita käytetään kenttien sisällön tarkastamiseen, on jätetty rakenteesta kuvaamatta kokonaan. Tarkoituksena ei ole tuoda esille kaikkea sovelluksen sisältämää toiminnallisuutta, vaan rakennekaavion avulla visualisoidaan ydintoiminnallisuudet sekä niiden yhteydet toisiinsa.



Kuvio 1. Sovelluksen ydintoiminnallisuuksien rakenne.

Opinnäytetyön sovelluksen ja toiminnallisuudet sisältävän moduulin toiminnallisuudet on toteutettu yhteensopiviksi Android-käyttöjärjestelmän versiosta 5.0 eteenpäin. Tätä aiempia Android-käyttöjärjestelmän versioita sovellus ei tue. Yhteensopivuus versioon 5.0 asti on todettu yhdessä toimeksiantajan kanssa riittäväksi, eikä tukea aiemmille versioille ole nähty tarpeelliseksi toteuttaa.

5.2.2 Alpaca

Alpaca on Gitana Softwaren kehittämä avoimen lähdekoodin JavaScript-kirjasto, jonka avulla on mahdollista luoda web-käyttöön mukautuvia lomakkeita. Alpaca käyttää JSON-sisältöä (JavaScript Object Notation) lomakkeiden sisäl-

lön tietojen hallintaan ja kirjasto tarjoaa valmiit toiminnallisuudet HTML5-lomakkeiden toteuttamiseksi web-käyttöön. JSON-tiedostomuotoa käytetään yleisesti tiedonvälitykseen esimerkiksi siirrettäessä tietoa järjestelmien, palvelimien ja sovelluksien välillä ohjelmointirajapintojen kautta. Alpaca sisältää valmiiksi toteutettuna monipuoliset toiminnallisuudet erilaisten lomakekenttien toteuttamiseksi ja siihen on mahdollista myös kehittää omia toimintoja jo olemassa olevien toiminnallisuuksien lisäksi. (Gitana Software 2017.)

Ennen opinnäytetyön toimeksiantoa toimeksiantaja oli jo osaltaan selvittänyt valmiiksi toteutusratkaisussa hyödynnettäviä vaihtoehtoja. Toimeksiantajan suosituksen kautta Alpaca-kirjasto valikoitui hyödynnettäväksi osana sovelluksen toteutusta. Sen monipuolisuus ja mahdollisuudet osoittautuivat myös sellaisiksi, joita ei ole tällä hetkellä tarjolla muissa avoimen lähdekoodin ohjelmointikirjastoissa. Alpaca käyttää kolmea erilaista JSON-syötettä lomakkeiden sisällön hallintaa varten. Kyseisten syötteiden sisällön kokonaisuuksista ei valmista dokumentaatiota löytynyt, jonka vuoksi sisältöjen rakenteeseen tutustuminen vaati huomattavasti aikaa toteutuksen alkuvaiheessa.

Ensimmäisenä Alpaca lukee Schema JSON-syötteen, joka määrittää lomakkeelle varsinaisen rakenteen eli mallin. Sen avulla pystytään luomaan sisällössä kuvatut kentät käyttöliittymään. Schema voi sisältää myös kullekin kentälle kuuluvia lisätietoja, kuten esimerkiksi otsikon, kuvauksen sekä tiedon siitä, onko kenttä pakollinen. Scheman sisältö on Alpacan tärkein sisältö, sillä pelkästään sen avulla pystytään jo luomaan mukautuvia lomakkeita web-käyttöliittymään.

Toinen Alpacan käyttämistä JSON-syötteistä on Options, joka sisältää tarkentavia lisätietoja yksittäistä lomakkeen kenttää koskien. Sen avulla voidaan ylikirjoittaa Scheman sisältämiä kenttien kuvauksia ja määrittää kentille tarkempia määrittämiä. Tämän syötteen avulla voidaan esimerkiksi määrittää tekstikenttä yksirivisestä kentästä laajemmaksi moniriviseksi kentäksi.

Kolmas JSON-syöte on Data, joka sisältää ainoastaan kenttien sisältämät tiedot. Datan sisältämässä JSON-syötteessä mainitaan ainoastaan kenttä, johon tieto kuuluu sekä arvo, jonka kenttä sisältää. Data JSON-syötteen tehtävänä on

tuoda lomakkeen kenttiin valmiiksi arvot esimerkiksi tilanteessa, jossa tarkastellaan jo aiemmin täytettyä lomaketta.

Toteuttamani sovelluksen moduuli käsittelee Alpacan käyttämää Schema- sekä Options-sisältöä, jonka avulla lomakkeet luodaan. Sisältö noudattaa ennalta määritettyä rakennetta, jonka avulla pystytään selvittämään jokaisen lomakkeen kentän tyyppi, sisältö sekä sijainti lomakkeella. Sisällössä on myös mahdollista määrittää erilaisia sisennettyjä kenttiä, jotka näkyvät käyttöliittymässä muista eroteltuna. Tämän lisäksi on mahdollista määrittää erilaisia dynaamisia kenttiä, joita käyttäjän on mahdollista käyttöliittymästä käsin lisätä tai poistaa.

Opinnäytetyöhön ei tässä vaiheessa sisältynyt ollenkaan Data-syötteen käsittelyä, sillä katsoimme yhdessä toimeksiantajan kanssa sen olevan lisäominaisuus, joka ei ole välttämätön sovelluksen toiminnan kannalta. Data-syötteen sisällön käsittelyn toteuttaminen olisi ollut myös erittäin haastavaa opinnäytetyön aikataulun puitteissa.

5.2.3 Lomakkeen sisällön hakeminen rajapinnasta

Sovellukseen toteuttamani toiminnallisuudet sisältävä moduuli hakee REST-pohjaisesta web-rajapinnasta lomakkeen JSON-sisällön (kuvio 1). REST on lyhenne sanoista Representational State Transfer ja se on eräänlainen sovel-lusarkkitehtuurin tyyli, jota käytetään erilaisissa hajautetuissa järjestelmissä. REST-pohjaiset rajapinnat ovat nykyisin erittäin yleisessä käytössä ja niiden avulla eri alustoilla toimivat sovellukset pystyvät kommunikoimaan keskenään. Pääajatuksena REST-arkkitehtuurissa on jakaa sovelluksen toiminta kahteen osaan, joista toinen on käyttäjän ja toinen palvelimen vastuulla. Yleensä toiminnallisuudet on jaettu siten, että käyttäjän sovellus huolehtii tietojen esittämisestä käyttöliittymässä samalla, kun palvelin huolehtii tietojen tallentamisesta. Palvelimen pääasiallinen tehtävä on yleensä hakea ja muokata tietokantaan tallennettua tietoa. (Pihlajaniemi 2012, 4.)

Opinnäytetyössäni lomakkeiden luontia varten toteutettu moduuli hakee lomakkeen JSON-sisällön olemassa olevasta rajapinnasta hyödyntäen Retrofit-ohjelmointikirjastoa. Retrofit on Squaren kehittämä avoimen lähdekoodin Android- ja Java-kirjasto, jonka avulla Android-projekteissa pystytään luomaan helposti yhteys haluttuun HTTP-rajapintaan. Kyseinen kirjasto on toimeksiantajan yleisesti käyttämä ja toimeksiantajan toiveena oli juuri kyseisen kirjaston hyödyntäminen opinnäytetyötä tehdessä. Tämän vuoksi kirjasto valikoitui käytettäväksi opinnäytetyössäni ja se on osoittanut myös erinomaiseksi työkaluksi rajapintakyselyitä tehtäessä.

Palvelimen palauttaessa moduulille JSON-syötteen, Retrofit-kirjaston laajenuksena oleva kääntäjä muuttaa JSON-sisällön erillisten Java-luokkien instansseiksi. Kääntäjän muodostettua instanssit toteuttamistani Java-luokista sovellus pystyy tulkitsemaan rajapinnasta saatua sisältöä ja suorittamaan haluttuja toimintoja sisällön perusteella. Näitä toteuttamiani luokkia kutsutaan ohjelmoinnissa yleisesti Modelleiksi eli luokiksi, jotka sisältävät sovelluksen tiedon sekä logiikan.

5.2.4 Näkymien muodostaminen käyttöliittymään

Moduulin rajapinnasta saama JSON-sisältö sisältää tiedon jokaisesta lomakkeen kentästä ja kentän tyypistä. Nämä jokaisen kentän sisältämät yksilölliset tiedot välittävät moduulille tiedon suoritettavista toiminnallisuuksista, joiden avulla luodaan haluttu lomake-elementti sovelluksen käyttöliittymään. Käyttöliittymäelementti, johon lomake-elementti luodaan, on määritetty sovelluksessa ja tieto halutusta käyttöliittymäelementistä on tuotu moduuliin sovelluksesta käsin. Opinnäytetyössäni olen toteuttanut valmiiksi tekstikenttien, numeraalisia arvoja sisältävien kenttien, puhelinnumerokenttien, sähköpostikenttien, salasana kenttien, pudotusvalikoiden sekä valintaruutujen toteutuksen JSON-sisällön perusteella. Tekstikentät, numeraalisia arvoja sisältävät kentät, puhelinnumerokentät, sähköpostikentät sekä salasana kentät ovat ulkoasultaan keskenään hyvinkin identtisiä, mutta ne eroavat osittain sisällöltään.

Yleinen tekstikenttä on edellä mainituista kentistä kaikkein vapaamuotoisin ja siihen käyttäjän on mahdollista syöttää mitä tahansa merkkejä. Yleisessä tekstikentässä ei ole rajoitettu tekstin sisällön muotoa lainkaan, joten kenttä soveltuu käytettäväksi tilanteissa, joissa lomakekenttään tulee syöttää esimerkiksi vapaamuotoista tekstiä. Kenttä on myös mahdollista määrittää syötteessä moniriviseksi, jolloin tekstikenttä näkyy normaalin yhden rivin sijaan monirivisenä tekstikenttänä. Tätä toiminnallisuutta hyödynnetään erityisesti silloin, kun lomakkeella on tarve kerätä useamman virkkeen pituista käyttäjän syöttämää tekstiä.

Numeraalisia arvoja sisältävään kenttään käyttäjä voi syöttää ainoastaan numeroita ja käyttöliittymässä käyttäjälle ei anneta edes mahdollisuutta syöttää kenttään muita arvoja. Kentän ollessa lomakkeella aktiivisena, avautuu mobiililaitteen näkymään Android-käyttöjärjestelmän käyttämä kosketusnäppäimistö, jonka sisältö on rajoitettu näyttämään ainoastaan numeroita kyseisen kentön osalta. Numeraalisia arvoja sisältäviä kenttiä voidaan hyödyntää silloin, kun on tarve kerätä haluttuja arvoja. Pelkkien numeroiden syöttäminen helpottaa myös syötettyjen lukujen ohjelmallista tarkastelua, mikäli lukujen tarkastelulle on tarvetta.

Puhelinnumerokenttä on numeraalisia arvoja sisältävän kentän kanssa lähes identtinen, mutta numeraalisten arvojen lisäksi kenttään voi syöttää puhelinnumeroiden sisältämiä erikoismerkkejä sekä kirjaimia. Erikoismerkeille voi olla tarve erityisesti syötettäessä ulkomaisia puhelinnumeroita, jotka saattavat sisältää numeroiden lisäksi mahdollisesti myös muita merkkejä.

Sähköpostikentän sisällöksi käyttäjä voi syöttää mitä tahansa merkkejä, kuten yleiseen tekstikenttään. Sähköpostikentän erona yleiseen tekstikenttään verrattuna on kuitenkin kentän erityinen tarkastus lomakkeen tietoja lähetettäessä. Toteuttamani ohjelmalliset tarkistukset varmistavat, että kentän sisältö vastaa sähköpostiosoitteiden vaatimaa muotoa. Sisällön tarkastamisessa hyödynnän valmista Java-ohjelmointikielelle toteutettua EmailValidator-luokkaa. Mikäli käyttäjän syöttämä sisältö ei vastaa sähköpostiosoitteen yleistä muotoa, näytetään käyttäjälle virheilmoitus käyttöliittymässä.

Salasanakentät ovat myös tekstikenttiä, joiden sisältöä ei ole käyttäjältä rajoitettu lainkaan. Kenttään syötetyt merkit ainoastaan piilotetaan käyttäjältä sisältöä syötettäessä, jotta ulkopuoliset eivät voi nähdä kenttään syötettyä sisältöä. Salasanakenttää voidaan hyödyntää lomakkeissa, joissa käyttäjän tarvitsee syöttää käyttäjätietoja esimerkiksi kirjaututtaessa tai rekisteröidytessä mobiilisovellukseen.

Pudotusvalikot sisältävät usein useita eri vaihtoehtoja, jotka käyttäjä saa sovelluksessa esille valitsemalla kyseisen kentän aktiiviseksi. Mikäli kenttää ei ole valittu aktiiviseksi, näytetään käyttöliittymässä vain pudotusvalikosta vaihtoehto, joka on valittu aktiiviseksi. Oletuksena pudotusvalikossa näytetään valinnaksi vaihtoehto, joka on ollut rajapinnasta tullessa syötteessä ensimmäisenä listattuna. Pudotusvalikoita voidaan hyödyntää silloin, kun lomakkeella halutaan käyttäjän valitsevan vaihtoehdon ennalta määritettyjen vaihtoehtojen joukosta.

Valintaruudut ovat nimensä mukaisesti kenttiä, joihin käyttäjän on mahdollista valita valinta aktiiviseksi. Valintaruutu voi myös oletuksena olla jo aktiivinen, mikäli valinta on määritetty saadussa syötteessä. Valintaruutuja hyödynnetään lomakkeissa erityisesti silloin, kun käyttäjältä on tarve saada hyväksyminen johonkin lomakkeella esitettyyn asiaan.

Lomakkeen käyttöliittymäelementteihin on toteutettu erilaisia sisällön tarkastuksia, joiden avulla rajapinnasta saadun syötteen avulla voidaan määrittää yksittäisiä kenttien rajoituksia. Yksittäinen kenttä voidaan esimerkiksi määrittää pakolliseksi, jolloin kentän sisällön puuttuessa ei lomakkeen tietoja tallenneta käyttäjän painaessa tallennuspainiketta, vaan käyttöliittymässä näytetään kenttään liittyvä virheilmoitus. Lomakkeessa oleville kentille voidaan määrittää myös pituusrajoituksia, jolloin on mahdollista hallita sitä, kuinka monta merkkiä käyttäjän tulee vähintään syöttää tai kuinka monta merkkiä kenttään voi enintään syöttää.

Opinnäytetyön toteutuksesta puuttuu toiminnallisuuksia, joita alkuperäinen Alpacan web-pohjainen toteutus tarjoaa, mutta kaikkien toiminnallisuuksien toteutus ei opinnäytetyössä ollut mahdollista rajallisen aikataulun takia. Alkuperäinen

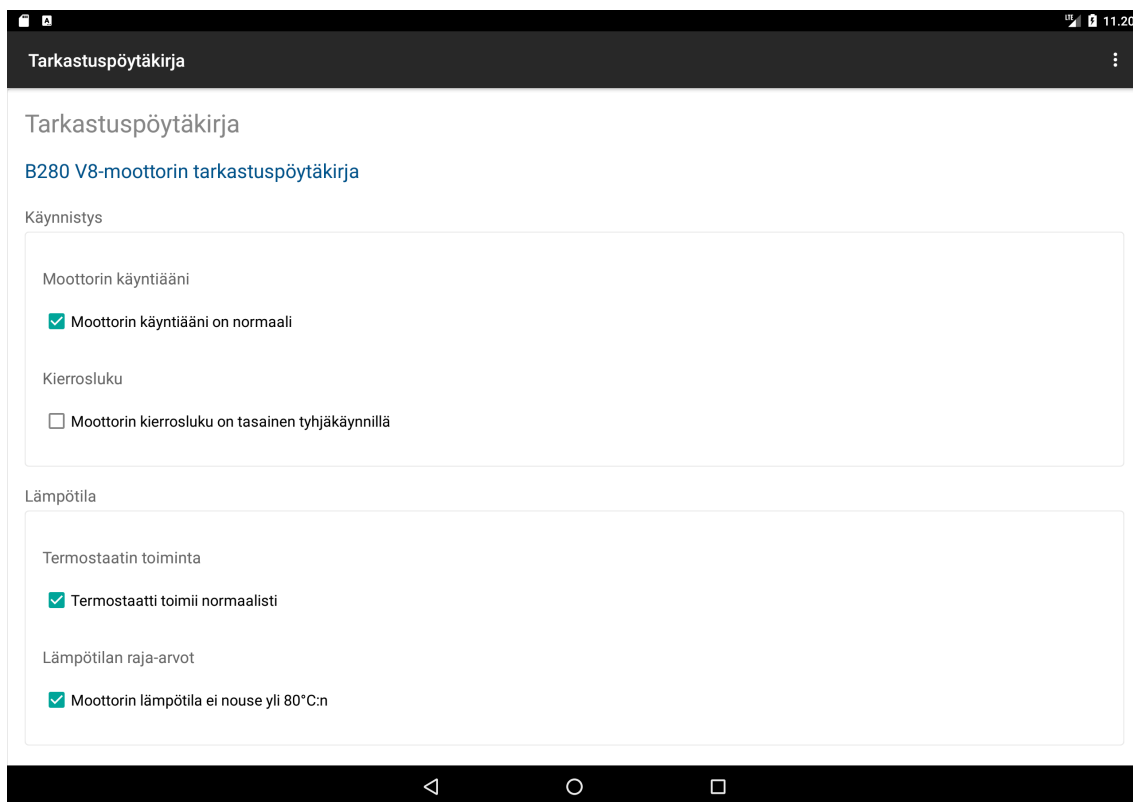
web-pohjainen toteutus sisältää lukuisia eri käyttöliittymäelementtejä ja määrittäjiä, joita yksittäiselle lomakkeen kentälle voidaan määrittää. Näistä toiminnallisuuksista on yhdessä toimeksiantajan kanssa valittu toteutettaviksi ne toiminnallisuudet, jotka mahdollistavat tässä vaiheessa toteutuksen hyödyntämisen mahdollisimman monissa eri käyttötarpeissa.

6 Toteutuksen hyödyntäminen teollisuudessa

Opinnäytetyöni toteutuksen mukautuvuus mahdollistaa toteutuksen hyödyntämisen teollisuudessa useilla eri osa-alueilla. Sen avulla toteutettuja lomakkeita voivat hyödyntää teollisuuden alan yrityksissä tuotannon parissa työskentelevät sekä esimiestehtävissä toimivat henkilöt. Esimerkkejä erilaisista käyttötapauksista, joissa toteutusta voidaan hyödyntää, kuvataan tarkemmin tämän luvun alaluvuissa. Esille tuodut esimerkit ovat opinnäytetyötä varten toteutettuja, eivätkä ne ole samassa muodossa käytössä toimeksiantajan toteutuksissa. Sisällöiltään esimerkit kuitenkin vastaavat yleisesti käytössä olevia lomakkeita, joita teollisuuden alan yritykset käyttävät toiminnassaan.

6.1 Tarkastuspöytäkirja

Tarkastuspöytäkirjan avulla valmistunut tuote käydään yksityiskohtaisesti läpi ja varmistetaan tuotteen täyttävän vaaditut laatuksiteerit. Esimerkkitapaukseksi kuvassa 1 on kuvattu yksittäisen moottorin tarkastuspöytäkirja, jonka valintaruutuihin merkitään toteutuneet kohdat. Mikäli tarkasteltavassa tuotteessa havaittaisiin puutteita, ei valintaruutuun laitettaisi merkintää ollenkaan. Tarkastuspöytäkirjassa voidaan hyödyntää myös tekstikenttiä, joihin voidaan tarvittaessa kirjata muita huomioita. Tämän lisäksi tarkastuspöytäkirjassa voi olla tarve syöttää erilaisia numeerisia arvoja esimerkiksi tarkastettaessa tuotteesta tarvittavia arvoja ja lukemia.



Kuva 1. Moottorin tarkastuspöytäkirjan mobiilinäkymä.

Sähköisen tarkastuspöytäkirjan avulla kirjattuja tuloksia on helppo hakea ja arvioida mikäli tarve myöhemmälle tarkastelulle ilmenee. Sähköisesti kerätyillä tuloksilla mahdollistetaan myös erilaisten tilastojen muodostaminen tuloksista, ja usein havaittujen puutteiden pohjalta on mahdollista toteuttaa esimerkiksi automatisoituja ilmoituksia, joihin tuotannossa pystytään reagoimaan.

6.2 Turvallisuushavainnot

Turvallisuushavaintolomakkeen avulla kerätään tietoa tuotantotiloissa havaituista puutteista turvallisuuteen liittyen. Turvallisuushavainnot voidaan tehdä tuotannon lattiatasolla työskentelevät henkilöt. Tämän lisäksi turvallisuushavaintojen tekoa varten voidaan järjestää erillisiä turvallisuuskävelyjä, joissa valitut henkilöt kiertävät tuotantotiloja ja merkitsevät lomakkeelle havaintojaan.

Opinnäytetyössä toteutetun sovelluksen avulla on mahdollista tuoda helpotusta tekstin luvussa 4.2 mainitsemaani ongelmaan, jossa havainnot kirjataan ensin paperille ja lopuksi kaikki kirjatut huomiot syötetään uudelleen järjestelmään

työasemalta käsin. Kuvassa 2 kuvatun lomakkeen avulla turvallisuushavaintojen kirjaaminen onnistuu suoraan turvallisuuskävelyn aikana mobiililaitteen avulla. Kävelyä suorittavan henkilön ei tarvitse kuljettaa mukanaan paperisia lomakkeita, joihin huomiot kirjataan, vaan mobiililomakkeeseen on mahdollista syöttää suoraan havaitut huomiot.

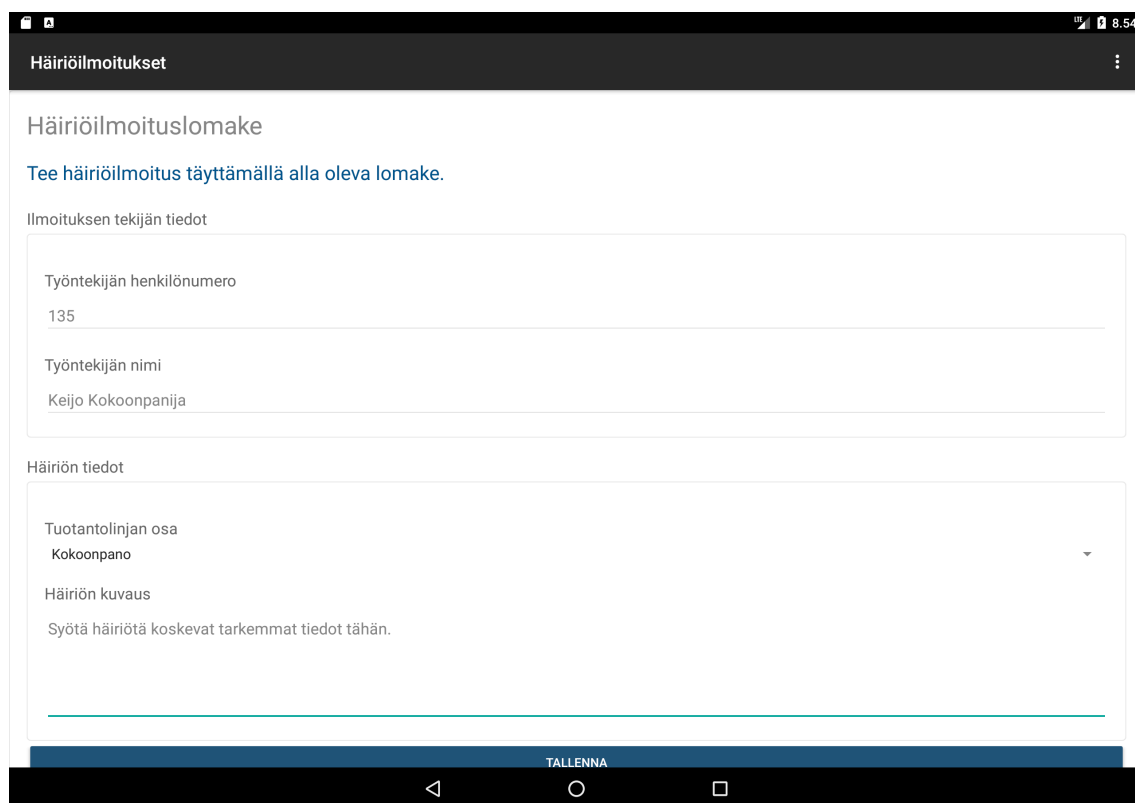
Kuva 2. Turvallisuushavaintolomakkeen mobiilinäkymä.

6.3 Häiriöilmoitukset

Häiriöilmoituslomake on tuotannossa käytettävä tapa ilmoittaa työpisteellä sattuneesta häiriötilanteesta, kuten esimerkiksi laiterikosta tai muusta työntekoon vaikuttavasta asiasta. Usein toiminnanohjausjärjestelmää käytettäessä häiriöilmoitukset kirjataan toiminnanohjausjärjestelmään erillisen päätelaitteen avulla, joka sisältää sähköisen lomakkeen häiriöilmoituksen tekoa varten. Erillisen järjestelmän puuttuessa häiriöilmoitukset kirjataan ainoastaan paperille, joka toimitetaan eteenpäin tarvittaville henkilöille. Ilmoitusten tekeminen ei edellä mainituissa tilanteissa ole välttämättä työntekijälle kovin vaivatonta, minkä vuoksi ilmoitusten tekoa saatetaan pahimmassa tapauksessa jopa laiminlyödä. Toteut-

tamani sovelluksen avulla työntekijä voi tehdä häiriöilmoituksen suoraan työpisteeltä käsin ilman, että hänen tarvitsee poistua työpisteeltään. Tämä mahdollistaa sen, että työntekijä pystyy kuvaamaan tarkemmin tapahtuneen häiriötilanteen suoraan työpisteeltä käsin nähdessään häiriön lähteen kuvausta kirjoittaessaan.

Kuvassa 3 on esimerkki häiriöilmoituslomakkeesta, jollainen työntekijän mobiililaitteessa voidaan esittää. Kyseisessä esimerkissä työntekijän henkilönnumero sekä työntekijän nimi on lomakkeelle haettu suoraan kirjautuneen käyttäjän tietojen perusteella. Ilmoituksen tekijän tarvitsee esimerkkitapauksessa ainoastaan valita tuotantolinjan osa, jossa häiriö on tapahtunut sekä kirjoittaa kuvaus tapahtuneesta. Mahdollistamalla ilmoitusten tekemisen suoraan mobiililaitteen avulla voidaan pienentää työntekijän kynnystä tehdä havaittu häiriöilmoitus ja näin tuotannossa pystytään tehokkaammin reagoimaan ilmenneisiin häiriötilanteisiin.



Häiriöilmoitukset

Häiriöilmoituslomake

[Tee häiriöilmoitus täyttämällä alla oleva lomake.](#)

Ilmoituksen tekijän tiedot

Työntekijän henkilönnumero
135

Työntekijän nimi
Keijo Kokoonpanija

Häiriön tiedot

Tuotantolinjan osa
Kokoonpano

Häiriön kuvaus
Syötä häiriötä koskevat tarkemmat tiedot tähän.

TALLENNNA

Kuva 3. Häiriöilmoituslomakkeen näkymä mobiililaitteella.

7 Tulokset

Opinnäytetyön tavoitteena oli saada toimeksiantajalle toimiva toteutus, jota olisi mahdollista hyödyntää monipuolisesti teollisuuden alalla ja samalla tehostaa teollista tuotantoa poistamalla tuottamatonta työtä. Opinnäytetyön alkuvaiheessa oma tietämykseni teollisuuden alan tuotannon toiminnasta oli varsin vähäistä, mikä asetti haasteita opinnäytetyön toteutukseen. Tarkoituksena oli kuitenkin huomioida toteutuksessa pääsääntöisesti teollisuuden alan tarpeita, vaikka toteutusta onkin mahdollista hyödyntää vastaamaan myös muiden toimialojen alan tarpeita. Toteutuksen edetessä ja asiaan tutustuesssa teollisuuden alan toiminta hahmottui itselleni paremmin ja toimeksiantajan kautta sai kattavasti tietoa teollisuuden alan tämän hetkisistä tarpeista, joihin ratkaisuja pyritään löytämään. Tämä mahdollisti opinnäytetyössä toteutettavien toiminnallisuuksien valitsemisen sen pohjalta, mistä toimeksiantaja hyötyisi eniten teollisuuteen liittyvissä projekteissa.

Toteutuksen alkuvaiheessa tekemäni ratkaisu toteuttaa kaikki toiminnallisuudet omana moduulinaan osoittautui onnistuneeksi ratkaisuksi, sillä sen avulla toteutuksen hyödyntäminen on osoittautunut helpoksi. Moduuliin toteutetut toiminnallisuudet mahdollistavat myös sen, että lomakkeiden käyttöliittymä pystytään räätälöimään täysin asiakkaan mukaan siitä sovelluksesta käsin, johon moduuli on liitetty. Ulkoasun räätälöinti mahdollistaa toteutuksen mukautuvuuden esimerkiksi eri asiakkaiden brändeihin, eikä se ole sidottu opinnäytetyön toteutuksessa toteuttamaani ulkoasupohjaan.

8 Jatkokehitysmahdollisuudet

Opinnäytetyö on alusta asti toteutettu mahdollisia jatkokehitystarpeita ajatellen. Toteutetun sovelluksen rakenteessa on pyritty ottamaan huomioon mahdollisimman helppo toiminnallisuuksien laajennettavuus sekä jo olemassa olevien toiminnallisuuksien muokkaus. Toteutus ei ole myöskään sidottu toteuttamaani

sovellukseen, sillä kaikki toiminnallisuudet sisältävä moduuli on mahdollista ottaa helposti osaksi mitä tahansa muuta projektia, jolloin toteuttamani toiminnallisuudet voidaan yhdistää olemassa oleviin tai täysin uusiin Android-sovelluksiin.

Ohjelmakoodin rakenne on jatkokehitystä ajatellen suunniteltu siten, että Alpacan web-käytössä käyttämän JSON-sisällön lukeminen uusien toiminnallisuuksien osalta onnistuu melko pienellä vaivalla. Uusien toiminnallisuuksien käyttöönotto onnistuu päivittämällä tekstin luvussa 5.2.3 kuvattuja ohjelmointiluokkia, jotka sisältävät sovelluksen logiikan. Luokkien rakenteen päivittämisen jälkeen uusille ominaisuuksille voidaan toteuttaa haluttuja toimintoja. Uudet toiminnot ohjelmoidaan ohjelmointiluokkaan, joka suorittaa ainoastaan käyttöliittymäelementtien toteutukseen liittyviä toimintoja. Toteutuksessa olen pyrkinyt erottamaan kaikki sovelluksen eri toiminnallisuudet toisista erilleen, jolloin toteutus on helpommin hallittavissa ja laajennettavissa. Rakenne mahdollistaa myös sen, että muiden ohjelmoijien on helpompi tutustua ohjelmakoodiin ja muokata toteutettuja toiminnallisuuksia ilman, että toteutuksen rakennetta ja ohjelmakoodia tarvitsee dokumentoida tarkasti.

Sovellus hyödyntää vahvasti toteutuksessa rajapintayhteyttä, josta lomakkeen sisältö saadaan haettua. Sovelluksen monikäyttöisyyttä ajatellen olisi tärkeää toteuttaa tarvittavat toiminnallisuudet käyttäjän syöttämän sisällön lähettämiseksi takaisin rajapintaan. Rajapintaan tallennettaessa syötetyt tiedot saataisiin kerättyä talteen palvelimen tietokantaan ja välitettyä edelleen muille käyttäjille sekä olemassa oleviin järjestelmiin. Rajapintayhteyden avulla lomakkeilta kerättyjen tietojen tallennus olemassa oleviin järjestelmiin olisi mahdollista mikäli rajapinnasta olisi yhteys järjestelmän käyttämään tietokantaan.

Integraatio esimerkiksi toiminnanohjausjärjestelmän tietokannan kanssa mahdollistaisi lomakkeiden tehokkaan hyödyntämisen, sillä lomakkeille olisi mahdollista tuoda myös tietokannasta valmiiksi tarvittavia esitäytettyjä tietoja. Esimerkiksi kirjautuneen käyttäjän tietojen perusteella lomakkeelle ei tarvitsisi erikseen syöttää käyttäjän henkilötietoja, vaan tiedot haettaisiin suoraan toiminnanohjausjärjestelmän tietokannasta.

Jatkokehitystä ajatellen tärkeässä asemassa on myös erilaisten tiedostolataukseen tarkoitettujen kenttien toteutus, joiden avulla käyttäjän on mahdollista ladata lomakkeelta suoraan palvelimelle kuvia sekä erilaisia tiedostoja tarpeen mukaan. Erityisesti kuvien lataus mahdollistaa toteutuksen tehokkaamman hyödyntämisen esimerkiksi teollisuuden alan häiriöiden ja poikkeusten seurannassa. Kuvien avulla voidaan vähentää tai poistaa kokonaan käyttäjältä tarve kirjoittaa erillistä sanallista kuvausta tapahtuneesta häiriöstä. Kuvien avulla myös tapahtuneesta häiriöstä tiedon vastaanottaneet henkilöt pystyvät saamaan tarkemman kuvauksen tilanteesta.

Alpaca käyttämä JSON-sisältö mahdollistaa toiminnallisuuksien toteutuksen yhteensä lähes viidellekymmenelle erilaiselle ja eri tyyppiselle lomakekentälle, joista tämän hetkiseen toteutukseen on kokonaisuutta ajatellen toteutettu vain hyvin pieni osa. Useat kentistä ovat keskenään hyvin pitkälle samanlaisia, mutta ne voivat sisältää erilaisia rajoituksia esimerkiksi kenttään syötettävän sisällön suhteen. Tarvittaessa opinnäytetyön toteutusta on mahdollista laajentaa hyödyntämään vaikka kaikkia Alpaca tarjoamia toiminnallisuuksia, mutta opinnäytetyön aikatauluun laajempien toiminnallisuuksien toteutus ei mahtunut. Toteutettaviksi ominaisuuksiksi valittiin yhdessä toimeksiantajan kanssa ne, joiden katsottiin olevan tällä hetkellä kaikkein oleellisimpia. Laajentamalla toteutusta myöhemmässä vaiheessa on sovellusta mahdollista hyödyntää yhä laajemmin vastaamaan erilaisten loppukäyttäjien tarpeita.

9 Pohdinta

Opinnäytetyön aihe osoittautui erittäin mielenkiintoiseksi ja sisällöltään hyvinkin ajankohtaiseksi. Mielenkiintoa opinnäytetyön toteutusta kohtaan lisäsi tieto siitä, että hyvin onnistuessaan toteuttamani ratkaisua hyvin todennäköisesti hyödynnettäisiin tulevaisuudessa useissa eri asiakasprojekteissa. Tämä tieto osaltaan nosti tavoitteeni opinnäytetyön onnistumisen suhteen korkealle, sillä halusin toteutuksestani olevan hyötyä myös jatkossa.

Kuten aiemmin raportissa mainittiin, opinnäytetyön prosessin aikana ei täysin vastaavia toteutuksia löytynyt. Valmiiden vastaavien toteutusten puuttuminen toi osaltaan haasteita toteutukseen ja toteutusratkaisussa täytyi monilta osin luottaa vahvasti omaan osaamiseen. Tekninen toteutus kuitenkin sujui oman arvioni mukaan hyvin ja mitään toiminnallisuutta ei sovelluksesta jäänyt puuttumaan sen takia, että toteutuksen aikana olisi ilmennyt liian suuria teknisiä haasteita.

Aiheena dynaamisten lomakkeiden käyttö on myös tällä hetkellä ajankohtainen ja se liittyy isompaan kokonaisuuteen, jossa useat teollisuuden alan yritykset ovat vähitellen vahvasti siirtymässä hyödyntämään mobiililaitteita osana joka päiväistä toimintaansa. Yrityksille on erittäin tärkeää tehostaa jatkuvasti tuotantoon ja pientenkin tuotantoa tehostavien toimenpiteiden avulla voidaan saavuttaa pitkällä aikavälillä huomattavia taloudellisia hyötyjä. Tuotannossa mobiilin hyödyntäminen mahdollistaa monien vaiheiden toteuttamisen vaivattomammin ja nopeammin. Yhtenä mobiilin mahdollistamista tuotantoa tehostavista toimenpiteistä ovat myös dynaamiset lomakkeet, jotka toimivat yhtenä pienenä osana mobiilin tarjoamien mahdollisuuksien kokonaisuudessa.

Tässä vaiheessa on vielä vaikeaa todeta opinnäytetyön mahdollistamia lopullisia hyötyjä, sillä toimeksiantajan asiakkaiden, eli toteutuksen loppukäyttäjien käyttökokemuksia ei vielä tässä vaiheessa ole mahdollista selvittää. Toteutuksen hyötynä näen kuitenkin sen, että sitä on helppo mukauttaa nopeasti vastaamaan asiakkaiden muuttuvia tarpeita. Oman kokemukseni mukaan ohjelmistoprojekteissa asiakkaiden tarpeet toteutettujen ominaisuuksien suhteen muuttuvat usein varsinkin siinä vaiheessa, kun asiakkaat saavat toteutuksesta ensimmäisen version käyttöönsä. Tämä johtuu usein siitä, etteivät asiakkaat itsenkään hahmota tarvitsemiaan toiminnallisuuksia ennen kuin he pääsevät ensimmäisiä kertoja kokeilemaan toteutettuja toiminnallisuuksia. Helpolla mukautettavuudella, joita dynaamiset lomakkeet tarjoavat, mahdollistetaan reagoiminen asiakkaan muutostarpeisiin pienellä työmäärällä ja nopealla aikataululla.

Lähteet

- Collapick Company Oy. 2017. Palvelut
<http://www.collapick.com/fi/palvelut>. 26.3.2017.
- Gitana Software. 2017. Alpaca – Easy Forms for jQuery.
<http://alpaca.js.org/index.html>. 17.3.2017.
- International Data Corporation. 2016. Smartphone OS Market Share, 2016 Q3.
<http://www.idc.com/promo/smartphone-market-share/os>. 18.3.2017.
- Mantere, T. 2014. Organisaation tietojärjestelmät -wikikirja.
http://lipas.uwasa.fi/~timan/TITE2060/OT2014_wiki.pdf. 11.2.2017.
- Mattsson, T. 2013. Toiminnanohjausjärjestelmät. Metropolia-ammattikorkeakoulu. Auto- ja kuljetustekniikan koulutusohjelma. Insinööri-työ.
- Pihlajaniemi, J. 2012. REST-pohjaisen web-rajapinnan kehittäminen. Metropolia-ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Insinööri-työ.
- Rouse, M., Brislen P., Krishnakumar K., O'Donnell, J. 2016. ERP (enterprise resource planning). <http://searchsap.techtarget.com/definition/ERP>. 9.4.2017.
- Samsung. 2017. Galaxy Xcover 3.
<http://www.samsung.com/fi/smartphones/galaxy-xcover-3-g389f/>. 18.3.2017.
- Skaggs, T. 2010. Essential in Lean Manufacturing is the 5-S Philosophy.
http://www.leanexpertise.com/TPMONLINE//articles_on_total_productive_maintenance/leanmfg/5sphilosophy.htm. 11.2.2017.
- Teknologiateollisuus ry. 2010. Tuotannonohjaus.
http://www.edu.fi/download/120998_6187_Tuotannonohjaus.pdf. 11.2.2017.
- Tietoyhteiskunnan kehittämiskeskus ry. 2005. Yrityksen tietojärjestelmät.
<http://www.tieke.fi/pages/viewpage.action?pageId=3441230>. 9.4.2017.
- Tyrväinen, T. 2011. Lean-ajattelu ja sen hyödyntäminen logistiikkakeskuksen toiminnan virtaviivaistamisessa. Metropolia-ammattikorkeakoulu. Tuotantotalouden koulutusohjelma. Insinööri-työ.
- Wikipedia. 2017. Tuotannonohjaus.
<https://fi.wikipedia.org/wiki/Tuotannonohjaus>. 12.2.2017.