

TALLINNA POLÜTEHNIKUM

IT ja telekommunikatsiooni erialaosakond

Andreas Kuuskaru

**LAOLIIKUMISTE JA TOOTMISE
RAPORTEERIMISE RAKENDUS ANDROID
OPERATSIOONISÜSTEEMIDEGA SEADMETELE**

Lõputöö

NOOREM TARKVARAARENDAJA

TA-15

Juhendaja: Urmas Siitan

Tallinn 2018

AUTORIDEKLARATSIOON

Deklareerin, et käesolev lõputöö, mis on minu iseseisva töö tulemus, on esitatud Tallinna Polütehnikumi lõputunnistuse taotlemiseks noorem tarkvaraarendaja erialal. Lõputöö alusel ei ole varem eriala lõputunnistust taotletud.

Autor A. Kuuskaru

(allkiri ja kuupäev)

Töö vastab kehtivatele nõuetele

Juhendaja U.Siitan

(allkiri ja kuupäev)

SISUKORD

LÜHENDITE JA TÄHISTE LOETELU	5
SISSEJUHATUS	6
1 TEOREETILINE OSA	8
1.1 Ettevõtte tutvustus ja probleemi püstitus.....	8
1.1.1 Usesoft AS	8
1.1.2 Majandustarkvara ProdMaster	8
1.1.3 Lisaarendust vajavad moodulid	8
1.2 Arendamisel kasutatud tehnoloogiad	9
1.2.1 Delphi.....	9
1.2.2 Apache	9
1.2.3 Android SDK	9
1.2.4 Microsoft SQL Server.....	10
1.2.5 Java SDK.....	10
1.2.6 ZXing ("zebra crossing")	10
1.2.7 Visual Studio Code	10
1.2.8 Microsoft SQL Server Management Studio.....	10
2 RAKENDUSLIK OSA	11
2.1 Rakenduste arendus.....	11
2.1.1 Eeltöö ja projekti loomine.....	11
2.1.2 Arenduskeskkonna Androidi arenduseks ettevalmistamine	11
2.1.3 Rakenduste disainimine	11
2.2 Koodi kirjutamine, programmeerimine	15
2.2.1 Object Pascal.....	15
2.2.2 PHP	15
2.3 Tootmistellimuste raporteerimise rakenduse funktsionaalsus.....	15
2.3.1 Sisselogimine	15
2.3.2 Kasutaja oleku kontroll	16
2.3.3 Uue töö alustamine.....	16
2.3.4 Töö raporteerimine.....	17
2.3.5 Paus ja jätkamine	17
2.4 Laoliikumiste rakenduse funktsionaalsus.....	17
2.4.1 Sisselogimine	17
2.4.2 Andmete sisestus.....	18
2.4.3 Andmete skaneerimine.....	18

2.4.4	Valimine ja otsimine	19
2.4.5	Töö raporteerimine.....	20
2.4.6	Laoseisu vaatamine	20
2.5	Andmete vahetus rakenduse ja andmebaasi vahel	21
2.6	Rakenduste testimine ja silumine	21
2.7	Paigaldus ja uuendamine	22
2.7.1	Rakenduse paigaldus.....	22
2.7.2	Rakenduse uuendamine.....	23
3	Tulemused, järeldused ja ettepanekud	25
KOKKUVÕTE.....		26
KASUTATUD KIRJANDUS		27
LISAD		28

LÜHENDITE JA TÄHISTE LOETELU

GB	GigaBait (ingl <i>GigaByte</i>)
SQL	Struktureeritud päringute keel (ingl <i>Structured Query Language</i>)
PHP	Hüperteksti eelprotssessor (ingl <i>Hypertext Preprocessor</i>)
USB	Universaalne jadasiin (ingl <i>Universal Serial Bus</i>)
SDK	Tarkvaraarenduse komplekt (ingl <i>Software Development Kit</i>)
Andmebaasidraiver	PHP laiend, mis võimaldab suhelda andmebaasiga (ingl <i>Database driver</i>)
Microsoft SQL Server Management Studio	Integreeritud keskkond mistahes SQL infrastruktuuri haldamiseks SQL Serverist SQL andmebaasini.

SISSEJUHATUS

Tänapäevases infoühiskonnas on väga palju erinevaid tarkvarasid, mille põhieesmärk on muuta inimeste elu lihtsamaks. On erinevaid lahendusi nii eraisikutele kui ettevõtetele, kuid iga lahenduse edu sõltub suuresti selle kasutusmugavusest. Kuigi ka kasutusmugavus võib iga ettevõtte ja inimese jaoks olla erineva tähendusega, on selle juures siiski põhilisteks märksõnadeks kiirus ning efektiivsus. Suur osa pakutavatest tarkvaraprogrammidest, mis turul on saadaval, on siiski standardlahendused ning ei vasta täpselt kõikide ettevõtete vajadustele. Selleks, et kliendi soove täita ja nende kasutusmugavust parandada on tihti vajalik teha juurdearendusi tarkvara täiustamiseks, mis sünnivad kliendi ning arendaja tihedas koostöös ning nõuavad tihti palju ressursse.

Käesoleva lõputöö põhieesmärgiks ongi leida lahendus kliendi soovile oma majandustarkvara ProdMaster täiustamiseks, mille tulemusena paraneb tootmisettevõtetele pakutava lahenduse efektiivsus. Selleks arendati Androidi platvormile kaks mobiilirakendust, mis täiustasid ProdMaster majandustarkvara lao- ning tootmismooduleid.

Laoliikumiste rakenduse eesmärgiks on võimalikult kiirelt ja lihtsalt laos esemete liigutamise raporteerimine ning tootmistellimuste raporteerimise rakenduse eesmärgiks on võimalikult kiirelt ja lihtsalt aidata kasutajal raporteerida tehtud tööd ning selle mahtu.

Rakenduste arendamiseks kasutati Windows 10 operatsioonisüsteemiga arvutit, millel on *Intel® Core™2 Duo E7500* protsessor ning 4GB operatiivmälu. Rakenduste testimiseks ja silumiseks kasutati Samsung Galaxy Tab A 7.0 (2016) tahvelarvutit, kus operatsioonisüsteemiks on Android 7.0. Mõlemad rakendused on programmeeritud kasutades *Delphi 10.2* arenduskeskkonda, mis võimaldab *Object Pascal*-it kasutades luua multi platvorm rakendusi (ingl *multi platform application*) ühe lähtekoodiga, operatsioonisüsteemides Windows, MacOS, iOS, Android ja Linux. Rakenduse Androidi operatsioonisüsteemis jooksutamiseks lisati arenduskeskkonda Android SDK. Rakenduste *Microsoft SQL* andmebaasiga ühendamiseks on kasutatud *Apache* veebiserverit, milles programmeerimiskeelena kasutati PHP-d. Laoliikumiste rakenduse kaamera pildilt andmete tuvastamiseks kasutati avatud lähtekoodiga ZXing ("*zebra crossing*") teeki (ingl *library*). Rakendused on algselt mõeldud Androidi operatsioonisüsteemidega

seadmetele, mis tähendab omakorda seda, et kasutajakogemus ja kasutajaliides teistel platvormidel võivad erineda Androidil rakenduste kasutamisest.

Lõputöö esimene peatükk sisaldab rakendustega seotud asutuste tutvustust ning probleeme. Teises peatükis räägitakse rakenduste arendusest ning funktsionaalsusest. Kolmandas peatükis tuuakse välja tulemused, järeldused ning ettepanekud.

1 TEOREETILINE OSA

1.1 Ettevõtte tutvustus ja probleemi püstitus

1.1.1 Usesoft AS

Usesoft AS on äritarkvaralahendustele keskendunud Eesti IT firma, kelle peamisteks klientideks on Eestis tegutsevad ehitus- ja tootmisvaldkonna ettevõtted. Usesoft AS on asutatud 1991. aastal ja firmas töötab hetkel enam kui 20 spetsialisti ning konsultanti. Ettevõtte 2018. aasta prognooskäive on ~ 1,65 miljonit eurot.

Firma pakub majandustarkvara lahendusi tootmisvaldkonnale ning nende tootevalik sisaldab:

- *IFS Applications (Industrial and Financial Systems)* – majandustarkvara suurte ja väga suurte ettevõtete ressursside planeerimiseks ja haldamiseks.
- *ProdMaster* – majandustarkvara tootmisressursside haldmiseks ja planeerimiseks ettevõtetele kes on keskendunud eelkõige tootmisele. [1]

1.1.2 Majandustarkvara ProdMaster

Usesoft AS-i poolt väljatöötatud majandustarkvara ProdMaster on MRPII (*Manufacturing Resource Planning*) majandustarkvara, mille ülesandeks on tootmisettevõtete äriprotsesside haldamine. ProdMaster aitab juhtida efektiivset tootmist - planeerida nii materjali kui ka tootmisressursse ning hallata müüki, ostmist ja laovarvestust. Selleks on tarkvaral olemas nii tootmis-, ostu-, müügi- kui laomoodul ning võimalik on osta ka erinevaid liideseid. Seejuures tarkvara hinnad algavad 299 eurost kuus. [2]

1.1.3 Lisaarendust vajavad moodulid

Ettevõtte ühel kliendil oli senises tarkvara kasutamises üks probleem, mis oli seotud tootmismooduliga, kus probleemiks oli antud tarkvara kasutamisel andmete ajakohasus ja raporteerimise mugavus. Ettevõtte tootmises on tootmistellimused, kus on kirjas, mis materjali läheb vaja ja milliseid operatsioone on vaja teha. Senini käis töö paberi peal. Meister andis hommikul paberi töötajale ning alles õhtul sai töötaja tehtud töö raporteerida ehk lõpetada, kasutades selleks arvutis asuvat programmi. Nüüd sooviti uueks lahenduseks võimalust otse tootmises tellimusi raporteerida. Selleks peab tootmises olema vaid Android seade, kuhu kasutaja sisestab tootmistellimuse numbri ning valib operatsiooni, mida tahab konkreetse tellimuse juures teha. Pärast seda sisestab koguse ja vajadusel ka praagi koos

põhjusega ning kommentaari. Selline lahendus võimaldab operatiivsemalt tootmist jälgida. Kuna siiani polnud sobivat lahendust välja töötatud, mis sellele ettevõttele oleks sobinud, siis tuli kogu protsess ideest kuni teostuseni arendada.

Teine probleem polnud seotud otseselt kliendi tellimusega, vaid majandustarkvara omaniku sooviga testida lisaarendust laomoodulile. Ladudes toimub pidev kaupade ümberpaigutamine. Igal liigutatud laoühikul tuleb kindlaks teha kust ning kuhu midagi liigutati. Seda tehakse tihti nii, et loetakse silmadega infot triipkoodilt ning sisestatakse siis programmi, mis on tülikas ja aeganõudev. Lisaks sellele võib käsitsi info sisestamine tekitada vigu sisestatud informatsioonis. Probleemi lahendamiseks tekkis idee luua rakendus, mis aitab lihtsustada ning kiirendada info sisestamist rakendusse kasutades valikuvõimalusi ning triipkoodi skaneerimise võimalust. Selleks, et raporteerida tehtud tööd tuleb sisestada triipkoodilt: tootekood, lähteaadress ning sihtaadress. See kiirendaks info sisestamise protsessi ning aitaks vältida olukordi, kus puutetundlikul ekraanil informatsiooni sisestades midagi valesti kirjutatakse.

1.2 Arendamisel kasutatud tehnoloogiad

1.2.1 Delphi

Delphi on arenduskeskkond, mis võimaldab *Object Pascal*-it kasutades luua multi platvorm rakendusi (ingl *multi platform application*) ühe lähtekoodiga, operatsioonisüsteemides Windows, MacOS, iOS, Android ja Linux. *Delphi*'t kasutati, sest see oli ettevõtte põhiline tarkvaraarendusprogramm. [3]

1.2.2 Apache

Apache on veebiserver, mis võimaldab *PHP*'d kasutades luua dünaamilisi veebikohti ning ühendada erinevate andmebaasidega. *Apache*'t kasutati, et ühendada Android andmebaasiga. [4]

1.2.3 Android SDK

Android Software Development Kit on Androidi tarkvaraarenduspakett, mis sisaldab rakendusliidese teeke (ingl *API libraries*) ning vajalikke tööriistu rakenduste kompileerimiseks, testimiseks ning silumiseks. Seda kasutati rakenduse kompileerimiseks ning testimiseks Androidi platvormil [5]

1.2.4 Microsoft SQL Server

Microsoft SQL Server on andmebaaside haldur, kus hoitakse erinevaid andmebaase ning andmeid. Ettevõtte kasutas *Microsoft SQL Server*'it andmete hoidmiseks ning haldamiseks. [6]

1.2.5 Java SDK

Java Software Development Kit on Java tarkvaraarenduspakett, mis sisaldab rakendusliidese teeke (ingl *API libraries*) ning vajalikke tööriistu rakenduste kompileerimiseks, testimiseks ning silumiseks. [5]

1.2.6 ZXing ("zebra crossing")

ZXing on avatud lähtekoodiga teek, mis aitab tuvastada pildilt erinevaid 1D ja 2D koode ning neid tekstiks muuta. Seda teeki kasutati, sest see on avatud lähtekoodiga ning ühildub *Delphi*'ga. [7]

1.2.7 Visual Studio Code

Visual Studio Code on lähtekoodi redaktor (ingl *Source code editor*). [8]

1.2.8 Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio on integreeritud keskkond *SQL* infrastruktuuri haldamiseks. Seda keskkonda kasutati *SQL* päringute ning protseduride testimiseks. [9]

2 RAKENDUSLIK OSA

2.1 Rakenduste arendus

2.1.1 Eeltöö ja projekti loomine

Enne rakenduste arendamise alustamist uuriti, kuidas töötavad olemasolevad *Microsoft SQL* protseduurid, mida on hiljem rakenduses vaja kasutada. (Lisa 1) Selleks kasutati *Microsoft SQL Server Management Studio*'t. Pärast seda paigaldati *Apache* veebiserver ning lisati andmebaasidraiver, et suhelda *Microsoft SQL* andmebaasiga.

Et testida *driver*-i toimimist, loodi *PHP* skript, kuhu sisestati info andmebaasi kohta. Pärast edukat ühendamist üritati jooksutada samu protseduure, mis eelnevalt testiti *Microsoft SQL Server Management Studio*'s ning võrreldi väljundandmeid.

Selleks, et alustada rakenduse arendamist loodi kõigepealt *Delphi* arenduskeskkonnas uus projekt. Projekti loomiseks tuli valida **File > New > Multi-Device Application > Blank Application**. Seejuures tuleb tähele panna, et valitud oleks *Delphi* rakendus, mitte C++ rakendus. Pärast seda luuakse uus *Delphi* projekt ning tühi vorm (ingl *Blank Form*).

2.1.2 Arenduskeskkonna Androidi arenduseks ettevalmistamine

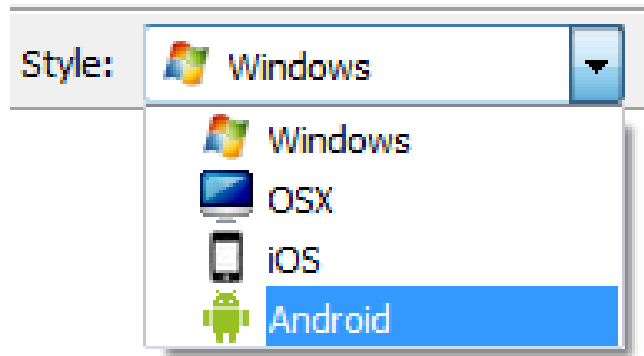
Androidi arenduse alustamiseks on vaja enne lisada *Delphi* arenduskeskkonda järgmised tarkvaraarenduse komplektid (ingl *software development kit*):

- *Android SDK*
- *Java SDK*

Delphi paigaldamisel arvutisse on tavaliselt vaikimisi kõik tarkvaraarenduse komplektid kaasas, juhul kui need puuduvad tuleb need alla laadida ja paigaldada ning lisada *Delphi* seadetes nende paigaldamise asukoht.

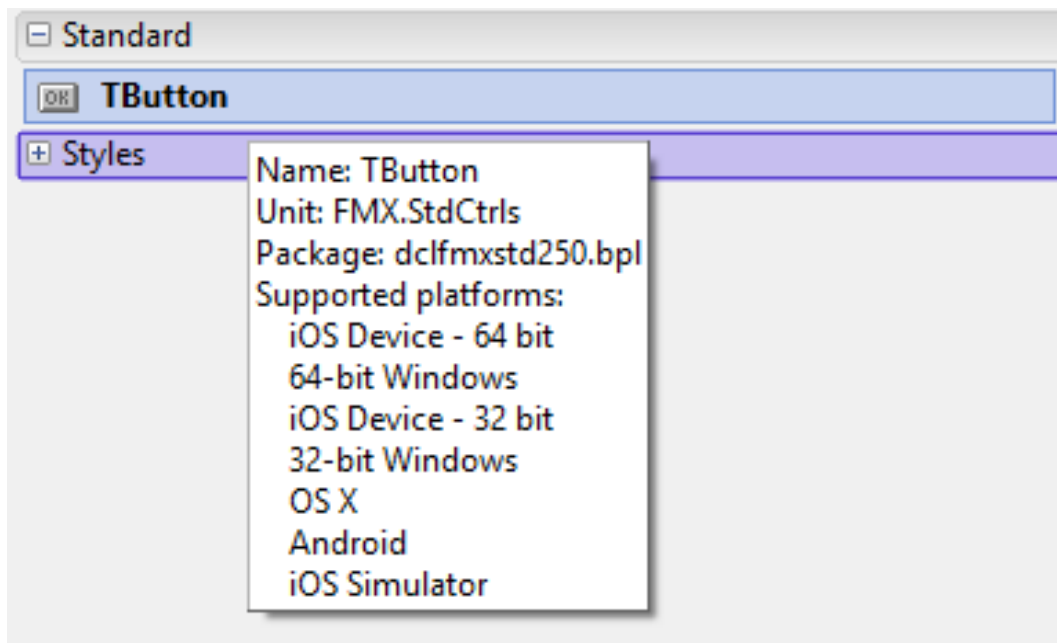
2.1.3 Rakenduste disainimine

Rakendusi saab disainima hakata peale uue projekti ning vormi loomist. Kuna *Delphi* võimaldab valmistada rakendusi erinevatele platvormidele samade komponentide ning koodiga, saab valikust valida, millise platvormi disaini tahetakse vaadata, ning milline hakkab rakendus välja nägema erinevatel platvormidel (vt. joon 1).



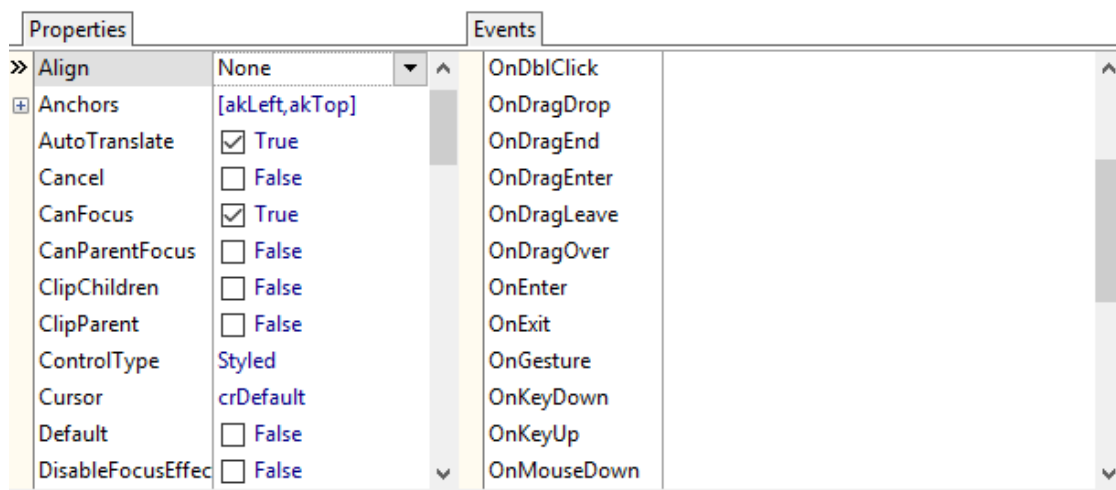
Joonis 1. Erinevate platvormide stiili valik

Erinevaid komponente saab vormile valida tööriistalaualt (ingl *tool palette*). Kui hõljuda hiirega tööriistalaua komponendi peal kuvatakse menüü, kus on kirjas, millistel platvormidel antud komponent töötab (ingl *supported platforms*) (vt. joon 2).



Joonis 2. Toetatud platvormide infomenüü





Pärast komponendi hiirega vormile lohistamist saab objekti omaduste (ingl *object properties*) menüüst muuta konkreetse valitud komponendi omadusi või objekti sündmuste (ingl *object events*) menüüst luua sündmusi (vt. joon 3).



Joonis 3. Omadused ja sündmused

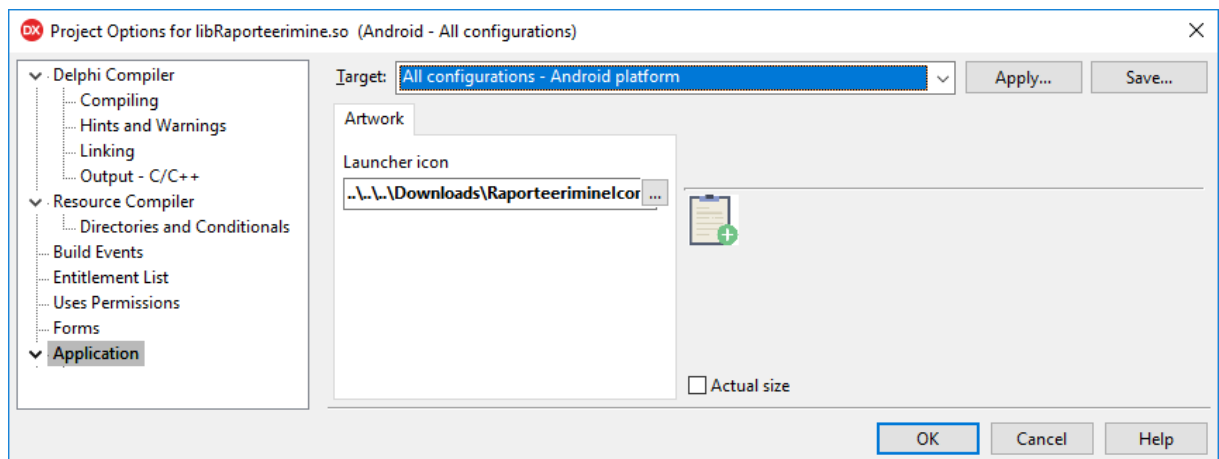
2.1.3.1 Ikooni muutmine

Iga rakenduse muudab omapäraseks ja äratuntavaks ikoon. Ikooni puudumisel mõned platvormid ei luba rakendusi käivitada ning sellepärast paneb *Delphi* rakendusele vaikimisi ikooni, mis tagab selle, et rakendus käivituks (vt. joon 4).

Platform	Image	Path
Windows		bin\delphi_PROJECTICON.ico
OS X		bin\delphi_PROJECTICON.icns
iOS		bin\Artwork\iOS\
Android		bin\Artwork\Android\

Joonis 4. Erinevate platvormide vaikimisi ikoonid

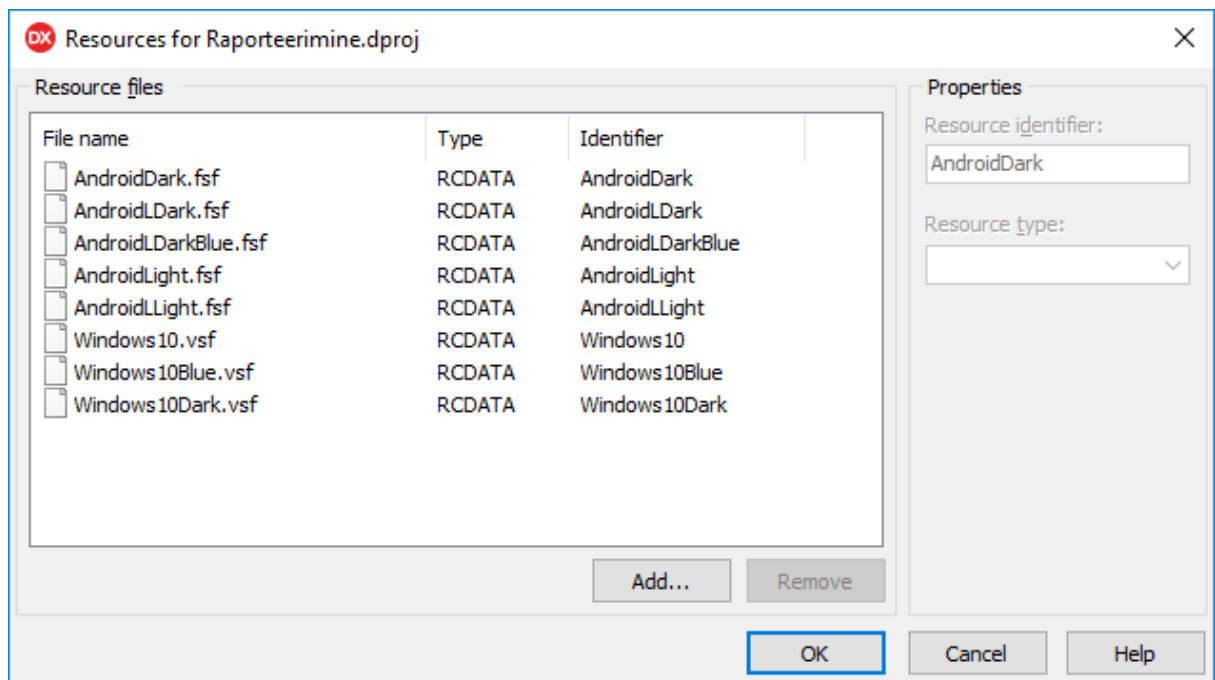
Ikooni muutmiseks tuleb suunduda **Project > Options > Application** ning valida, millise platvormi ikooni muuta tahetakse, antud juhul on see Android. Seejärel otsida arvutist välja valitud pilt ning see lisada (vt. joon 5).



Joonis 5. Rakenduse ikooni muutmine

2.1.3.2 Stiilide lisamine

Stiilide lisamine rakendusse on valikuline, kuid see on vajalik selleks, et rakendus näeks kõikidel platvormidel välja nii nagu konkreetsele operatsioonisüsteemile omapärane või juhul, kui tahetakse rakendusse lisada heleda ja tumeda stiili valikut. Nende lisamiseks või eemaldamiseks tuleb minna **Project > Resources and Images**. Vajutades nupule „Add“ saab lisada uusi stiili faile (vt. joon 6).



Joonis 6. Stiilide lisamine või eemaldamine

2.2 Koodi kirjutamine, programmeerimine

Koodi kirjutamine jaguneb kaheks osaks:

- 1) *Delphi*'s rakenduse valmistamine *Object Pascal*'it kasutades
- 2) *Apache* veebiserveris informatsiooni vahendamine andmebaasi ja rakenduse vahel kasutades *PHP*'d

2.2.1 Object Pascal

Delphi Object Pascal'i kood vastutab kõige selle eest, mis toimub rakenduse siseselt. Pärast disaini (ingl *front end*) valmimist saab hakata vormile sobitatud komponentidele lisama töötlevat tagasisüsteemi (ingl *back end*) eelnevalt loodud komponendi omaduste ja sündmuste abiga.

2.2.2 PHP

PHP kood täidab põhiliselt kolme ülesannet:

- Andmete vastuvõtmine rakendusest
- Andmebaasi protseduuride jooksumine
- Info tagastamine rakendusele

Kood asub *Apache* veebiserveris ning selle redigeerimiseks kasutati *Visual Studio Code*'i.

2.3 Tootmistellimuste raporteerimise rakenduse funktsionaalsus

Tootmistellimuste raporteerimise rakenduse funktsionaalsus toodud Lisas 2.

2.3.1 Sisselogimine

Rakenduse käimapanemisel avaneb vaikimisi sisselogimise aken, kuhu kasutaja saab sisestada oma kasutajanime ja sisse logida. Kui kasutajanimi on olemas, kuvatakse rakenduse sisu ning kontrollitakse, kas kasutajal on pooleliolevat tööd (vt. joon 7).

Joonis 7. Raporteerimise vaade

2.3.2 Kasutaja oleku kontroll

Kasutaja olekut kontrollitakse pärast igat andmebaasi poole pöördumist. Kontroll näitab, kas kasutajal on pooleliolevat tööd. Kui on, siis tagastatakse rakendusele, mis tüüpi töö on pooleli, kas abitöö või päris tellimus. Pärast seda täidetakse rakenduses väljad poolelioleva töö kohta ja neid muuta ei lubata. Poolelioleva töö info hulka kuuluvad: tootmistellimuse number, tootmistellimuse osa, operatsioon, kas kasutaja on pausil ja vajadusel ka abitegevus.

Kui aga kasutajal ei ole pooleliolevat tööd, siis lubab rakendus alustada uut tööd.

2.3.3 Uue töö alustamine

Uue töö alustamiseks on 2 viisi:

- 1) Kui tegemist on abitööga, näiteks koristamine, seadistus vms, siis kasutaja saab valida abitegevuste valikust sobiva töö ja seda koheselt alustada.
- 2) Kui aga töö, mida tahetakse teha, on tootmistellimus, tuleb kõigepealt sisestada tootmistellimuse number ning vajadusel ka tootmistellimuse osa ja seejärel saab otsida selle tellimusega seonduvaid operatsioone, mis kuvatakse operatsioonide

Liitkasti (ingl *Combobox*). Pärast operatsiooni valikut saab alustada tööd. Peale töö alustamist pöörduetakse andmebaasi poole ja jookсутatakse protseduuri sisestatud andmetega. Kui töö on alustatud teavitatakse kasutajale, et töö on edukalt alustatud. Pärast seda saab hakata alustatud tööd raporteerima.

2.3.4 Töö raporteerimine

Alustatud töö raporteerimiseks sisestab kasutaja tehtud töö koguse ja vajadusel ka praagi koguse, praagi põhjuse ning kommentaari. Kui on sisestatud praagi kogus, peab olema ka valitud põhjus ning vastupidi.

Kui aga tahetakse raporteerida abitööd, siis saab sisestada ainult kommentaari, sest abitööl ei saa olla kogust ega praaki.

Kui raporteeritud kogus ületab kogust, mis on vaja teha kuni 30%, näidatakse kasutajale hoiatust, kus on valik, kas lubada suuremad kogused või mitte. Suurema hulga puhul kui 30% hoiatust ei kuvata, vaid tuleb veateade, et raporteeritud kogus on liiga suur.

Juhul, kui töö sai edukalt raporteeritud, suundub rakendus tagasi algseisundisse, kus kasutaja saab alustada uut tööd.

2.3.5 Paus ja jätkamine

Töö ajal saab ka kasutaja teha pausi. Selleks on raporteerimise nupu kõrval pausi nupp. Pausi alustades jäetakse andmebaasi meelde, mis andmed olid sisestatud kasutaja poolt ning „Pausi“ nupp muutub „Jätka“ nupuks. Tööd jätkates võetakse need andmed uuesti andmebaasist ja lisatakse rakendusse. Juhul, kui pausil olles suletakse rakendus, täidetakse rakendus nende samade andmetega järgmisel sisselogimisel.

2.4 Laoliikumiste rakenduse funktsionaalsus

2.4.1 Sisselogimine

Rakenduse käimapanemisel kuvatakse vaikimisi sisselogimise vaade, kuhu kasutaja saab sisestada kasutajanime ning parooli ja sisse logida. Edukal sisselogimisel kuvatakse rakenduse sisu.

2.4.2 Andmete sisestus

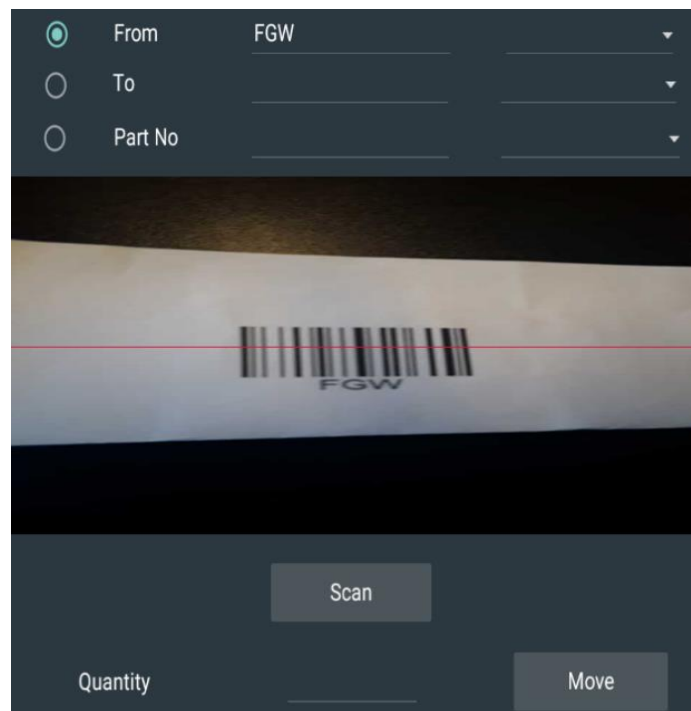
Liigutatud eseme raporteerimiseks tuleb sisestada järgmised andmed:

- 1) Aadress, kust liigutati (*From*)
- 2) Aadress, kuhu liigutati (*To*)
- 3) Toote number (*Part No*)
- 4) Kogus (*Quantity*)

Aadresside ja toote numbri sisestamiseks on 3 võimalust:

- 1) Klaviatuurilt kirjutades
- 2) Andmete skaneerimine
- 3) Valimine/Otsimine nime järgi

Pärast aadresside ja toote numbri sisestust tuleb sisestada ka kogus (vt. joon 8).



Joonis 8. Andmete sisestamise võimalused

2.4.3 Andmete skaneerimine

Andmete skaneerimiseks kasutab rakendus eelnevalt seadetest valitud meetodit.

Juhul, kui valitud on skaneerimine kaameraga, on vaikimisi pilt kaamerast kuvatud ning võimalik on andmete skaneerimine valides vasakult, missugust koodi parasjagu skaneerida

tahetakse. Pärast seda tuleb asetada triipkood kaamera keskele punase joone alla ning vajutada nupule „Scan“. Saadud tulemus kuvatakse vastavasse kasti ning skaneerimise kursorit liigutatakse automaatselt edasi. Nii on võimalik kõik kolm koodi järjest skaneerida, ilma, et peaks manuaalselt valima, mida skaneeritakse.

Kui aga on valitud skaneerimine välise *Bluetooth* skaneerijaga, siis kaamerast pilti ei näidata, aga automaatne edasiliikumine toimib endiselt pärast triipkoodi tuvastamist.

2.4.4 Valimine ja otsimine

Vajadusel saab aadresse valida ka valikust, vajutades kõrvalasuva liitkasti peale (vt. joon 9).

CARLOS carlos	<input type="radio"/>
CARLOS2 carlos	<input type="radio"/>
CINCIN CincinattiCell	<input type="radio"/>
CONSUMABLES Allusedconsumables(offices+factory)	<input type="radio"/>
DRILL Drill	<input type="radio"/>
FGW FinishedGoodsWarehouse	<input type="radio"/>

Joonis 9. Aadresside valik

Toote numbri valimisel tuleb vajutada samuti liitkasti peale ning seejärel nime (*Name_1*) järgi otsida toote numbrit (*Part_No*). Seejärel kuvatakse tulemused allolevas tabelis. Juhul, kui kasutuses on kaameraga skaneerimine, siis pilt kaamerast asendub tabeliga. Iga uue sisestatud tähe peale uuendatakse tabeli sisu (vt. joon 10).

Part No		001-0002_CN	EM0
Name_1	Part_No		
EM023332_1_2PCBStopper	001-0002		
EM023332_1_2PCBStopper	001-0002_CN		

Joonis 10. Toote numbri otsing

2.4.5 Töö raporteerimine

Sisestatud andmete raporteerimiseks tuleb vajutada nuppu „Move“. Juhul kui raporteerimine oli edukas, näidatakse kasutajale hetke laoseisu kõikidel aadressidel tabelis. Vastupidisel juhul, kui raporteerimine ebaõnnestus, näidatakse, kus esines viga.

2.4.6 Laoseisu vaatamine

Kasutajale kuvatakse laoseis automaatselt pärast raporteerimist või minnes manuaalselt andmebaasi vaatesse. Manuaalselt vaatamiseks tuleb vajutada üleval ribal asetsevat „Database“ nupule. Peale vajutamist küsitakse kasutajalt toote numbrit, mille põhjal kuvatakse laoseis tabeli kujul (vt. joon 11).

Tabelis kuvatakse:

- Aadress (*Addr*)
- Kogus (*Qty_OnHand*)
- Toote number (*Part_No*)
- Partii number (*Batch_No*)

DATABASE		INPUT	
Addr	Qty_OnHand	Part_No	Batch_No
CARLOS	2	001-0002	120202-1
DRILL	2	001-0002	120202-1
FGW	107	001-0002	0987654
KRISTI	38	001-0002	120202-1

Joonis 11. Laoseisu vaatamine

2.5 Andmete vahetus rakenduse ja andmebaasi vahel

Andmed liiguvad rakenduse ja andmebaasi vahel *Apache* veebiserveri abiga. Rakendus ütleb veebiserverile, millist funktsiooni on vaja teostada ja vajadusel annab kaasa ka kasutaja sisestatud andmed. Enne andmete saatmist krüpteeritakse andmed võtmega. Kui veebiserver on rakenduse poolt saadetud andmed kätte saanud, dekrüpteeritakse need võtmega. Seejärel loeb veebiserver, millist funktsiooni on vaja teostada. See on vajalik selleks, et server teaks, milliseid parameetreid ehk andmeid on vaja küsida. Pärast seda luuakse ühendus andmebaasiga ja kasutaja andmed lisatakse *SQL* protseduuri. Protseduurist saadud vastus saadetakse tagasi rakendusele.

2.6 Rakenduste testimine ja silumine

Rakenduse testimiseks Androidi seadmes lisati arenduskeskkonda *Delphi Androidi SDK* ning ühendati arvutiga USB kaabli abiga Samsung Galaxy Tab A 7.0 (2016) tahvelarvuti. Et androidi seadet arenduseks kasutada, on vaja enne suunduda **Seaded -> Arendaja valikud -> USB silumine** ning see sisse lülitada. Kui USB silumine on sisse lülitatud, küsib androidi seade USB kaabli arvutiga ühendamisel silumise lubamist. Pärast silumise lubamist ilmub antud seade arenduskeskkonnas ning on valmis arenduseks.

2.7 Paigaldus ja uuendamine

2.7.1 Rakenduse paigaldus

Selleks, et rakendust kasutada, tuleb enne kogu süsteem ülesse seada. Ülesse seadmine kujutab endast kliendi poolele *Apache* veebiserveri paigaldamist ning rakenduse installimist ja seadistamist seadmetesse.

Apache veebiserveri paigaldamisel tuleb tähele panna seda, et arvuti, kuhu paigaldama hakatakse, oleks kindlasti samas võrgus Androidi seadmetega. Peale seda tuleb seadmetesse rakendus paigalda ning teha esimene seadistus. Esimene seadistus kujutab endast rakenduse seadetesse sisse logimist kokkulepitud kasutajanime ja parooliga. Pärast edukat sisselogimist saab rakenduse seadistaja sisestada järgmised parameetrid:

- 1) Veebiserveri aadress
- 2) *Microsoft SQL Server*'i nimi
- 3) Andmebaas
- 4) Vajadusel saab valida ka rakenduse stiili.

Kui kõik andmed on väljadesse sisestatud saab andmed salvestada ning testida ühendust serveriga vajutades nupule „*Test*“. Andmed salvestatakse seadmesse ainult juhul, kui serveriga ühendus oli edukas ning pärast seda avaneb sisse logimise võimalus. Kui seadistades tekib probleem, siis antakse seadistajale teada probleemi põhjus. Salvestamisel võetakse kõik sisestatud info ja salvestatakse seadmesse krüpteeritult (vt. joon 12).

SISSE LOGIMINE SEADED

Address Test

Andmebaas

AndroidLight

Serveri nimi

Salvesta seaded

Salvestamata!

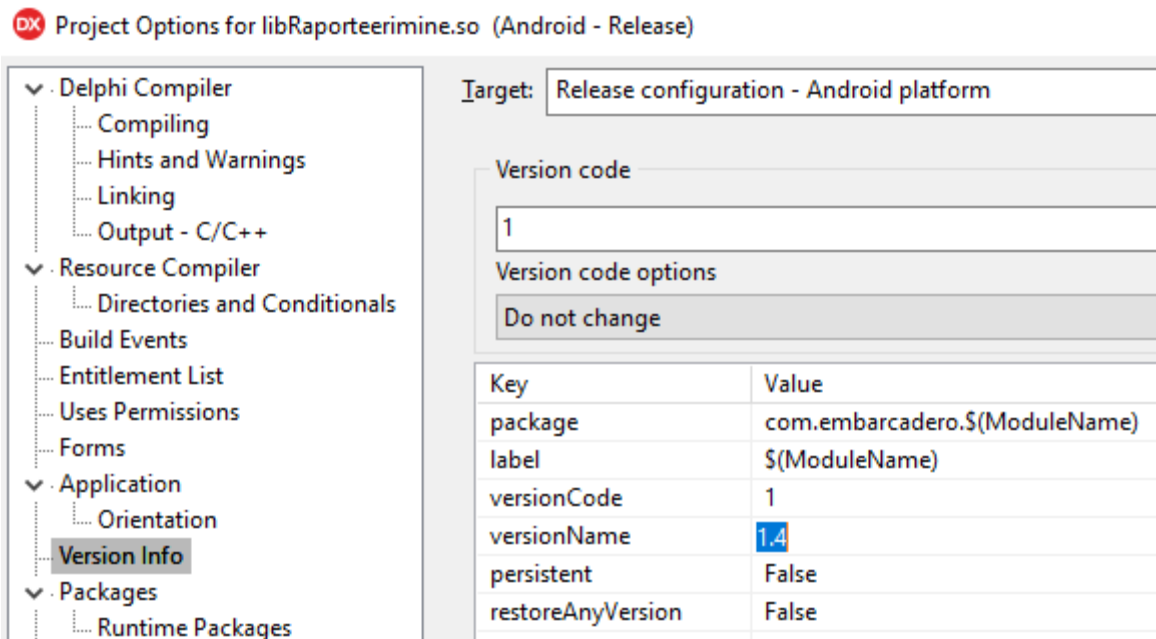
Joonis 12. Rakenduse seadistamine

2.7.2 Rakenduse uuendamine

Kui rakendus on seadistatud, siis pärast igat rakenduse käimapanekut kontrollitakse, kas see versioon, mis on seadmes, on sama, mis veebiserveris olev versioon. Juhul, kui rakendusel on saadaval uuendus, siis kuvatakse kasutajale sõnum, et saadaval on uuendus ja pärast seda antakse võimalus, kas uuendada või sulgeda rakendus.

Kui kasutaja tahab rakendust uuendada, siis laetakse veebiserverist seadmesse uusim rakenduse *APK* (ingl *android application package*) versioon ning käivitatakse see automaatselt. Pärast installimist saab kasutaja uuesti rakendust kasutama hakata. Rakenduse uuendamisel ei kustutata eelnevaid andmeid, mis olid ülespanekul seatud.

Versiooni uuendamiseks tuleb kõigepealt *Delphis* minna **Project > Options > Version Info** ning siis paremalasuvas tabelis ära muuta parameeter **versionName** (vt. joon 13).



Joonis 13. Versiooni muutmine

Pärast versiooninumbri muutmist tuleb luua rakendusele uus *APK* ning asetada see veebiserveri „*Updates*“ kausta.

3 Tulemused, järeldused ja ettepanekud

Kahest arendatud rakendusest on raporteerimise rakendus igapäevaselt ettevõttes kasutuses ning uuenenud pärast algset paigaldust. Kuna tegemist on multi platvorm rakendustega, siis ettepanekuks oleks vajadusel samade rakenduste tööle panemine ka teiste operatsioonisüsteemidega seadmetel.

KOKKUVÕTE

Käesolevas lõputöös käsitleti Androidile kahe rakenduse arendamist, kasutatud metoodikat ning autori poolseid tulemusi ja järeldusi.

Peamised põhjused rakenduste loomiseks olid info sisestamise protsessi kiirendamine, inimliku vea tekkimise võimaluse vähendamine ning raporteerimise protsessi moderniseerimine.

Lõputöö esimeses peatükis toodi välja rakendustega seotud asutused ning probleemid. Teises peatükis toodi välja arenduses kasutatud vahendid ning kolmandas peatükis räägitakse rakenduse arendusest.

Tänu antud lõputööle valmis asutuses Usesoft AS esimene lahendus, mis on mõeldud kasutamiseks Androidi platvormil.

KASUTATUD KIRJANDUS

- [1] U. AS. [Online]. Available: <http://usesoft.ee/ettevottest/>. [Accessed 4 mai 2018].
- [2] ProdMaster. [Online]. Available: <http://prodmaster.ee/>. [Accessed 4 mai 2018].
- [3] Delphi. [Online]. Available: <https://www.embarcadero.com/products/delphi>. [Accessed 16 märts 2018].
- [4] Apache. [Online]. Available: <https://httpd.apache.org/>. [Accessed 16 märts 2018].
- [5] SDK. [Online]. Available: <https://akit.cyber.ee/term/8856>. [Accessed 16 märts 2018].
- [6] S. Server. [Online]. Available: <http://www.vallaste.ee/>. [Accessed 16 märts 2018].
- [7] ZXing. [Online]. Available: <https://github.com/zxing/zxing>. [Accessed 16 märts 2018].
- [8] V. S. Code. [Online]. Available: <https://code.visualstudio.com/>. [Accessed 16 märts 2018].
- [9] M. S. S. M. Studio. [Online]. Available: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>. [Accessed 8 mai 2018].

LISAD

Lisa 1 Kasutatud protsedurid

Kasutaja tuvastamine

Kasutaja kontrolliks käivitatakse protseduur:

ProdMaster.PM_ExtProcedures_Scanfurn_CheckUser @ResourceID

Parameetrid:

ResourceID	Kasutaja tunnus
------------	--------------------

Protseduuri väljundiks on n-realine tabel:

Name	Kasutaja täisnimi
Found	Kasutaja tuvastatud/Sellist kasutajat pole

Abitegevuste nimekiri

Abitegevuste nimekirja lugemiseks käivitatakse protseduur:

ProdMaster.PM_ExtProcedures_Scanfurn_GetTimeTypes

Protseduuri väljundiks on n-realine tabel:

Time_Type	Abitegevus
-----------	------------

Praagi põhjuste nimekiri

Praagi põhjuste nimekirja lugemiseks käivitatakse protseduur:

ProdMaster.PM_ExtProcedures_Scanfurn_GetScrapReas

Protseduuri väljundiks on n-realine tabel:

Code	Praagi kood
Reason	Praagi põhjus

Kontrolli töölise olekut

Protseduur kontrollib, kas töölisel on pooleli operatsiooni ja kui on, väljastab selle operatsiooni andmed.

ProdMaster.PM_ExtProcedures_Scanfurn_CheckWorkerState @ResourceID

Parameetrid:

@ResourceID varchar(12) Kasutaja tunnus

Protseduuri väljundiks on üherealine tabel:

Result	Pooleli töö/Pole tööd pooleli
Is_Additional	Pooleli päris tellimus/Pooleli abitöö
Is_Pause	Pausil/Ei ole pausil
Order_No	Pooleli tellimuse nr
Release	Pooleli tellimuse osa
Operation	Operatsioon
Qty_Ordered	Kogus tellimusel
Qty_Ready	Juba valmis kogus
Add_Work	Pooleli abitöö liik

Kontrolli tootmistellimust

Valitud tootmistellimuse olemasolu ja oleku kontrollimiseks käivitatakse protseduur:

ProdMaster.PM_ExtProcedures_Scanfurn_CheckProdOrder @OrderNo, @Release

Parameetrid:

OrderNo	Tootmistellimuse või koondtellimuse nr.
Release	Tootmistellimuse osa või NULL, kui koondtellimus

Protseduuri väljundiks on 1-realine tabel:

Result	Tulemus
--------	---------

Võimalikud tulemused koos tähendusega:

0 – kõik korras

1 – sellist tootmistellimust pole

2 – tootmistellimus on vales olekus

3 – tootmistellimus on mõne koondtellimuse koosseisus

4 – tootmistellimusel on erinevad tehnoloogiad

Leia võimalikud operatsioonid

Töölise jaoks antud tellimusel võimalike operatsioonide leidmiseks käivitatakse

protseduur: ProdMaster.PM_ExtProcedures_Scanfurn_GetAvailOper @ResourceID,
@OrderNo, @Release

Parameetrid:

ResourceID	Kasutaja tunnus
OrderNo	Tootmistellimuse või koondtellimuse nr.
Release	Tootmistellimuse osa või NULL, kui koondtellimus

Protseduuri väljundiks on n-realine tabel:

Operation	Võimalik operatsioon
-----------	----------------------

Leia operatsiooni kogused

Valitud operatsiooni koguste leidmiseks käivitatakse protseduur:

ProdMaster.PM_ExtProcedures_Scanfurn_GetAvailQuantity @OrderNo, @Release,
@Operation,

Parameetrid:

OrderNo	Tootmistellimuse või koondtellimuse nr
Release	Tootmistellimuse osa või NULL, kui koondtellimus
Operation	Valitud operatsioon

Protseduuri väljundiks on üherealine tabel:

Qty_Ordered	Kogus tellimusel
Qty_Ready	Valmis kogus

Alusta operatsiooni

Valitud operatsiooni või abitegevuse alustamiseks käivitatakse protseduur:

ProdMaster.PM_ExtProcedures_Scanfurn_StartOper @OrderNo, @Release,
@Operation, @ResourceID

Parameetrid:

OrderNo	Tootmistellimuse või koondtellimuse nr või NULL kui alustatakse üldist abitegevust
Release	Tootmistellimuse osa või NULL, kui koondtellimus
Operation	Valitud operatsioon
AddWork	Abitegevuse tunnus
ResourceID	Kasutaja tunnus

Protseduuri väljundiks on üherealine tabel:

Result	Tulemus
--------	---------

Võimalikud tulemused koos tähendusega:

- 0 – Kõik korras
- 1 – Sellist abitööd pole
- 2 – Abitööde tellimust ei leitud
- 3 – Abitööde tellimusel pole operatsiooni
- 4 – Tootmistellimust ei leitud
- 5 – Operatsiooni ei leitud
- 6 – Antud ressurss ei saa seda operatsiooni teostada
- 7 – Tellimus on juba lõpetatud

Raporteerimise operatsiooni kogus või peatamine

Validud operatsiooni või abitegevuse lõpetamiseks või peatamiseks käivitatakse protseduur: ProdMaster.PM_ExtProcedures_Scanfurn_ReportOper @OrderNo, @Release, @Operation,

@ResourceID, @Quantity, @Scrap

Parameetrid:

OrderNo	Tootmistellimuse või koondtellimuse nr või NULL kui alustatakse üldist abitegevust
Release	Tootmistellimuse osa või NULL, kui koondtellimus
Operation	Validud operatsioon
ResourceID	Kasutaja tunnus
Quantity	Valmis saanud kogus(võib olla 0 töö peatamiseks)
Scrap	Praagi kogus
ScrapReas	Praagi põhjus
Comment	Kommentaari
AllowOver	Ei lubata suurem kogus/lubatakse suurem kogus

Protseduuri väljundiks on üherealine tabel:

Result	Tulemus
--------	---------

Võimalikud tulemused koos tähendusega:

0 – Kõik korras

-1 – Kogus on liiga suur. Kas lubame?

1 – Abitööde tellimust ei leitud

2 – Abitööde tellimusel pole operatsiooni

3 – Antud ressurss pole abitööd alustanud

4 – Antud ressurss pole seda operatsiooni alustanud

5 – Tellimus on juba lõpetatud

6 – Operatsioon on juba lõpetatud

7 – Raporteeritav kogus on liiga suur

8 – Tellimusel puudub partii nr.

9 – Tellimusel puudub lõpptoote aadress

10 – Vigane praagi põhjus

- 11 – Abitööle ei tohi olla kogust ega praaki
- 12 – Viga tellimuse raporteerimisel

Alusta pausi

Valitud operatsiooni või abitegevuse pausi alustamiseks käivitatakse protseduur:

ProdMaster.PM_ExtProcedures_Scanfurn_StartPause @OrderNo, @Release,
@Operation, @ResourceID, @Comment

Parameetrid:

OrderNo	Tootmistellimuse või koondtellimuse nr. või NULL, kui alustatakse üldise abitegevuse pausi
Release	Tootmistellimuse osa või NULL, kui koondtellimus või alustatakse üldise abitegevuse pausi
Operation	Valitud operatsioon
ResourceID	Kasutaja tunnus
Comment	Kommentaar

Protseduuri väljundiks on üherealine tabel:

Result	Tulemus
--------	---------

Võimalikud tulemused koos tähendusega:

- 0 – Kõik korras
- 1 – Abitööde tellimust ei leitud
- 2 – Antud ressurss pole abitööd alustanud
- 3 – Antud ressurss pole seda operatsiooni alustanud

Lõpeta paus

Valitud operatsiooni või abitegevuse pausi lõpetamiseks käivitatakse protseduur:

ProdMaster.PM_ExtProcedures_Scanfurn_EndPause @OrderNo, @Release,
@Operation,
@ResourceID

Parameetrid:

OrderNo	Tootmistellimuse või koondtellimuse nr. või NULL, kui alustatakse üldise abitegevuse pausi
Release	Tootmistellimuse osa või NULL, kui koondtellimus või alustatakse üldise abitegevuse pausi
Operation	Valitud operatsioon
ResourceID	Kasutaja tunnus

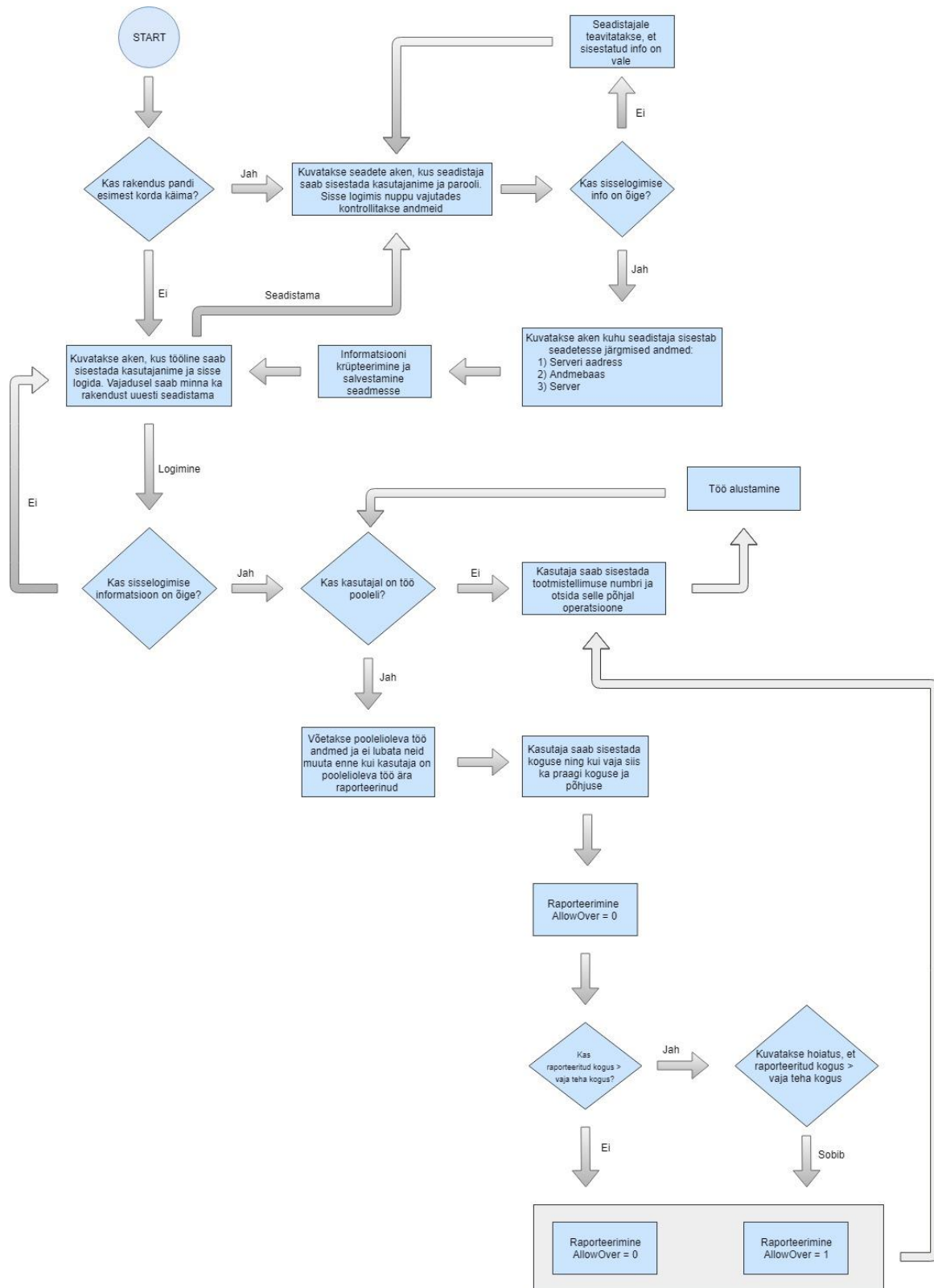
Protseduuri väljundiks on üherealine tabel:

Result	Tulemus
--------	---------

Võimalikud tulemused koos tähendusega:

- 0 – Kõik korras
- 1 – Abitööde tellimust ei leitud
- 2 – Antud ressurss pole pausi alustanud
- 3 – Ei leitud jätkatavat tööd

Lisa 2 Tootmistellimuste raporteerimise rakenduse algoritm



Lisa 3 Juhendaja hinnang lõputööle

Lõputöö teema	Laoliikumiste ja tootmise raporteerimise rakendus android operatsioonisüsteemidega seadmetele
Lõputöö autor	Andreas Kuuskaru
Suhtumine lõputöö kirjutamisse (aja planeerimine, õpilase kohusetundlikkus, koostöö õpilasega)	
Lõputöö kirjutamisse suhtus Andreas kohusetundlikult. Vajalikud andmed kogus ta õigeaegselt ja ka aja planeerimisega oli kõik korras. Töö praktiline osa valmis varakult. Seega jäik ka kirjutamiseks piisavalt aega.	
Teema aktuaalsus, rakenduslikkus	
Käesoleva töö teema on oma põhiosas pärit ühe Usesofti kliendi konkreetsest vajadusest. Töö tulemusena valminud rakendus on igati korralik ja praegusel hetkel kliendi juues igapäevases kasutuses.	
Üliõpilase võimekus (iseseisva töö oskus, järeltunde loogilisus ja põhjendatus, initsiatiivsus)	
Andreas iseseisva töö võime on väga hea. Tööd alustades oli ta üsna keerulises olukorras, kuna rakenduse jaoks vajalik arendusvahend (Delphi) oli talle täiesti tundmatu. Siiski suutis ta küllalt lühikese aja jooksul iseseisvalt vahendi endale selgeks teha ja töö korralikult ära teha. Ülesande selgitamisel näitas ta üles vajalikku initsiatiivi ja pakkus välja ka mitmeid omapoolseid huvitavaid lahendusi.	
Lõputöö vastavus vormistatusnõuetele (töö ülesehitus, viitamine, kirjutamise stiil, graafilise materjali valik)	
Töö ülesehitus vastab juhendaja arvates igati nõuetele. Vajalikud viited on olemas.	
Lõputöö tulemused (lahenduste keerukuse tase, ajakohasus, loogilisus ja põhjendatus, autori panus)	
Nagu juba eespool märgitud, vastab lõputöö tulemus igati ootustele ja on hetkel igapäevases kasutuses.	
Hinnang lõputööle (palun märkida järgmise skaala alusel: väga hea – „5“; hea – „4“; rahuldav – „3“; puudulik – „2“; nõrk – „1“)	
Juhendaja soovib hinnata tööd hindegiga „5“	

Juhendaja:

Allkiri:

Kuupäev: