



Toiminnanohjausjärjestelmän rajapintojen raportoinnin kehittäminen

Lauri Rinta-Opas

OPINNÄYTETYÖ
Helmikuu 2021

Liiketalous (päivätoteutus)
Taloushallinto

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Liiketalouden tutkinto-ohjelma
Taloushallinto

RINTA-OPAS, LAURI:

Toiminnanohjausjärjestelmän rajapintojen raportoinnin kehittäminen

Opinnäytetyö 29 sivua, joista liitteitä 0 sivua
Helmikuu 2021

Tämä opinnäytetyö toteutettiin toiminnallisena työnä suomalaiselle yritykselle. Toimeksiantajayrityksellä oli ongelma kiinteiden kulujen raportoinnin kanssa, mikä johtui toiminnanohjausjärjestelmän ja Microsoft Office Excelin välisestä rajapintaongelmasta. Aiemmin yrityksessä tiedot oli siirretty yksinkertaisesti ”kopioi ja liitä” -menetelmällä. Opinnäytetyön tarkoituksena oli luoda yritykselle mahdollisimman yksinkertainen Microsoft Excel Visual Basic for Applications -ohjelma, jolla kuluraportoinnin työmäärä laskisi automatisoimalla se mahdollisimman pitkälle.

Toimeksiantajan alkuperäiseen kiinteiden kulujen Microsoft Excel työkirjaan luotiin ohjelmakoodi, joka noutaa toiminnanohjausjärjestelmästä etukäteen Exceliin viedyistä raporteista halutut tiedot ja liittää ne halutulle solualueelle raporttiin. Lisäksi työkirjaan lisättiin visualisointia, sekä navigointipainikkeita porautumista varten.

Opinnäytetyö täytti toimeksiantajan odotukset ohjelmakoodista ja mikäli odotukset pitävät paikkaansa tulevaisuudessa ohjelmakoodi tulee säästämään useita työtunteja. Lisäksi on mahdollista, että juuri ohjelmakoodissa käytettyä automatisoitua ”kopioi ja liitä” -menetelmää jalostetaan muidenkin raporttien luomiseen, sillä se ei ole sidottu kyseisiin kiinteisiin kuluihin.

Asiasanat: excel, vba, taloushallinto, ERP, toiminnanohjausjärjestelmä,

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Administration
Financial Management

RINTA-OPAS, LAURI:
Development of ERP Interface Reporting

Bachelor's thesis 29 pages, appendices 0 pages
February 2021

This thesis was made as practice-based and its intent was to create and develop an easier solution for a big Finnish corporate to manage and create their expense reports on Microsoft Office Excel. The corporation had a problem with interfaces between their ERP and Microsoft Excel. In the past, the information had been transferred with a classic copy and paste method. The mission was to develop a solution and code it as simple as possible with Microsoft Excel Visual Basic for Applications -application. The aim with the solution was to decrease the amount of work with automatisisation.

A VBA-application was developed in the applicant's original Excel workbook. The macrocode fetches designated data from Excel workbooks which are imported from ERP beforehand and pastes it on the main workbook to a designated cell range on Microsoft Excel. In addition, visualization and navigation interfaces were added.

The thesis fulfilled the applicant's expectations about the macrocode. In the future, the application will save many hours and might be developed even further to help with other reports as is it not tied to only expense reports.

Key words: excel, vba, financial management, computation.

SISÄLLYS

1	JOHDANTO	5
2	TYÖN TAVOITE JA TOTEUTUSTAPA.....	6
	2.1 Työn tavoite	6
	2.2 Toteutustapa	6
	2.3 Rajaukset.....	6
3	YRITYKSEN ESITTELY.....	7
4	VISMA L7.....	8
5	MICROSOFT EXCEL VISUAL BASIC APPLICATION.....	9
	5.1 Microsoft Office Excel	9
	5.2 Visual Basic for Applications	9
	5.3 Esimerkki VBA-koodista	10
	5.4 Tietoturva	12
6	TYÖN TOTEUTTAMINEN	14
	6.1 Työn suunnittelu.....	14
	6.2 Toteutus	15
	6.3 Lopputulos	17
	6.3.1 "Kloonaa" komentopainike	18
	6.3.2 "Kopioi" komentopainike	18
	6.3.3 Tuhansilla jakaminen.....	21
	6.3.4 Porautuminen ja porauspainikkeiden värittäminen	23
	6.3.5 Työkirjan etuaukeaman selkeyttäminen	25
7	POHDINTA	26
	LÄHTEET	28

1 JOHDANTO

Viime vuosina sähköinen ja laajemmin automatisoitu taloushallinto on muuttanut taloushallinnon prosesseja. Uudet kehittyneet toiminnanohjausjärjestelmät ja ohjelmistorobotit rakentavat raportit itsenäisesti ja luovat muuta helposti käsiteltävää dataa käyttäjälleen. Näillä toimilla yritykset ja varsinkin yritysten taloushallinnot säästävät aikaa rutiinitehtävissä ja aikaa jää muihin tärkeämpiin tehtäviin. (Visma.fi n.d).

Yrityksen käyttäessä muita rajapintoja toiminnanohjausjärjestelmien kanssa saattaa silti syntyä työtehtäviä, jotka ovat puuduttavia sekä yksitoikkoisia, mutta vaativat silti käyttäjän tarkkaavaista keskittymistä. Pelkästään toiminnanohjausjärjestelmän luomasta raportista, tietojen siirto Microsoft Exceliin saattaa viedä aikaa, jos rajapinnat eivät keskustele toistensa kanssa. Vaikka Microsoft julkaisi Excelin jo melkein 35 vuotta sitten, se on edelleen useiden yritysten arkipäivää laskennassa. Useat yritykset käyttävät Exceliä viimeistään tiedon jälkikäsitelyssä, kuten kaavioiden luomisessa raportteihin.

Toimeksiantajayrityksellä ollut ongelma, johon tartuttiin tarjottuani opinnäytetyötäni yritykselle. Ongelma löytyi kiinteiden kulujen raportoinnista ja tietojen viemisestä Visma L7 Liinoksesta Microsoft Exceliin. Yrityksessä oli totuttu käyttämään Exceliä kiinteiden kulujen viimeisenä raporttina, sekä toteumaluku kerrallaan kopiomaan Vismasta data Excel-työkirjaan. Suorite vaati aikaa ja hermoja, eikä tämä koskenut pelkästään kiinteiden kulujen raportointia. Opinnäytetyölläni mahdollistin tämän tehtävän automatisoinnin käyttäen Microsoft Excel Visual Basic Applicationia. Raportoinnin automatisoinnin lisäksi panostin myös itse raportin ulkoasuun ja luettavuuteen.

Opinnäytetyöni teoriaosuus käsittelee toimeksiantajayritystä, Visma L7 Liinosta, sekä Microsoft Excel Visual Basic Applicationia. Opinnäytetyön loppupuolelta löytyy itse koodiosuus, jossa käydään läpi Excel VBA makron suunnittelu, toteutus ja lopputulos.

2 TYÖN TAVOITE JA TOTEUTUSTAPA

2.1 Työn tavoite

Työn tavoitteena oli luoda toimeksiantajayritykselle vaivattomampi tapa luoda toiminnanohjausjärjestelmästä tuotavista tiedoista Microsoft Excel raportti. Opinnäytetyön aloitushetkellä tiedot tuotiin Visma Liinos L7:n kiinteiden kulujen raportista ”kopioi ja liitä” -menetelmällä luku kerrallaan Exceliin. Lisäksi haluttiin panostaa taulukon luettavuuteen ja yksinkertaistamiseen. Taulukon on pysyttävä selvästi luettavana suhteessa tietoon mitä taulukosta on saatavilla.

2.2 Toteutustapa

Opinnäytetyö tehtiin toiminnallisena opinnäytetyönä Yritys X:lle, minkä lopputuloksena yrityksen kiinteiden kulujen raportointi Microsoft Exceliin nopeutuu ja täten säästää taloushallinnolta aikaa muihin tärkeämpiin työtehtäviin. Yritykseltä mukana prosessissa ja sen suunnittelussa olivat yrityksen talouspäällikkö, sekä järjestelmäasiantuntija. Käyttöön luovutettiin yrityksen tietokone, jolla pystyin etäjärjestelmää hyödyntäen tutkimaan ja kehittämään keinot luoda haluttu Microsoft Excel Visual Basic Application koodi sekä testaamaan sitä käytännössä etätyöskentelynä.

2.3 Rajaukset

Opinnäytetyö rajattiin käsittelemään ainoastaan kiinteiden kulujen tietojen siirtoa ja raportointia Visma Liinos L7:stä Microsoft Exceliin. Toimeksiantajalla oli myös toiveita lisätä Exceliin porautuvuutta, jolloin tietyn kustannuspaikan tietoihin olisi mahdollista porautua ja selvittää esimerkiksi, missä kustannuspaikassa budjetti on ylitetty. Porautuvuus pystyttiin mahdollistamaan ohjelmakoodiin. Tulevaisuudessa luomani Excel VBA ohjelmakoodi mahdollisesti sovelletaan myös muuhun tiedonsiirtoon ja raportointiin.

3 YRITYKSEN ESITTELY

Opinnäytetyössä yrityksestä käytetään nimitystä *yritys X* tai *toimeksiantajayritys* heidän pyynnöstään. Yritys X on suuri suomalainen kone- ja tuotantotekniikan perheyritys. Päätoimipaikka sijaitsee Suomessa ja neljä muuta toimipaikkaa ympäri maailmaa. Opinnäytetyöni on tehty konsernin emoyhtiölle, jonka toimialana on rahoituspalvelut. Emoyhtiön kautta toimii koko konsernin taloushallinto ja muu rahoitustoiminta. Vuonna 2019 konsernin liikevaihto oli jo useita kymmeniä miljoonia ja siten yritys on tehnyt lähivuosisikymmeninä suuria investointeja kasvuvauhdin kiihtyessä. Lähivuosina yritys X osti toisen saman toimialan yrityksen, joka avasi heille mahdollisuuksia myös kansainväliselle kasvulle. (Yritys X. n.d).

Yritys X on perustettu 1900-luvun ensimmäisellä puoliskolla niin, että yrityksen ensimmäiset harppaukset ovat sijoittuneet sodan jälkeiseen rakennusbuumiin, jota myöhemmin seurannut Suomen talouden elpyminen 60-luvulla mahdollisti yritykselle vielä paremmat mahdollisuudet kasvamiselle. 2010-luvulla yritys kasvoi kansainvälisestä lamasta huolimatta, lisäksi konsernin sisällä tehdyt muutokset ja nimeämiset siivittivät perheyrittäjäyrityksen merkittäviin investointeihin, jotka suurimmilta osin päättyivät vuoteen 2020. (Yritys X. n.d).

Yrityksen keskiössä on edelleen kuitenkin perheyrittäminen ja suomalaisuus. Järkevät investoinnit mahdollistavat yrityksen jatkuvan kilpailukykyyn ja tuotannon säilyttämisen Suomessa. Yrityksen koko toiminnan perusta on suomalainen osaaminen ja sitoutunut henkilöstö. Teollisuuden robotiikan ja 3D-tekniikan ansiosta tuotannon kustannukset tulevat todennäköisesti laskemaan seuraavina vuosikymmeninä ja silloin on hyvä, että tuotanto on siellä missä on useiden vuosikymmenten osaaminenkin. (Yritys X. n.d).

4 VISMA L7

Visma on norjalainen vuonna 1996 perustettu yritysohjelmistotarjoaja. Yrityksen tuotteiden tarkoitus on helpottaa asiakkaidensa rutiineja, työtehtäviä ja muita prosesseja. Visma saapui Suomeen vuonna 2001 ostettuaan suomalaisen Liinos Oyj:n, joka vaihtoi nimensä lopulta Visma Softwareksi. Nykypäivänä Vismalla on Suomessa yli 100 000 asiakasta. Lisäksi se työllistää Suomessa noin 1 400 työntekijää. (Visma.fi 1 n.d.)

Toiminnanohjauksen lisäksi Visma tarjoaa kymmeniä eri ohjelmistoja. Esimerkiksi ajanvaraus-, sähköisen allekirjoituksen-, työmaaratkaisu- ja opetustoimen hallinnon järjestelmiä. Ohjelmistojen lisäksi Visma tarjoaa palveluinaan esimerkiksi IT-konsultaatiota, lakipalveluita, automatisoitua maksunvalvontaa, laskutuspalvelua ja perintää. (Visma.fi 2 n.d.)

Opinnäytetyöni toimeksiantajayritys käyttää Visma L7 -toiminnanohjausjärjestelmää, talouden ja toiminnan ohjaukseen. Tämä toiminnanohjausjärjestelmä tunnetaan myös nimellä Liinos7 eli L7. Visma L7 on suunniteltu pääasiassa urakoinnin, energiatuotannon ja teollisuuden toimialoille. Visma L7 -sovellus kattaa niin taloushallinnon, projektinhallinnan, tuotannonohjauksen, varastohallinnan kuin henkilöstöhallinnankin. Yrityksen päätettävissä on, mitkä osa-alueet järjestelmään integroidaan. (Visma.fi 2 n.d.)

Toimeksiantajayritykseni käyttää siis Visma L7 -sovellusta päivittäin toiminnassaan, mutta huomioni tuotiin yksi pulma järjestelmän käytöstä. Visma L7 keskustelee huonosti muiden tiettyjen rajapintojen, kuten Microsoft Officen kanssa. Yritys pystyy järjestelmässä luomaan valmiita raportteja, analysoimaan tietoja ja valvoa toimintaa, Exceliin vienti toimii kuitenkin xls tai csv-tiedostona, mutta täten koko raportti siirtyy omaan taulukkoonsa. Ongelmana on jo olemassa olevaan Exceliin tietojen summaaminen ja vertaaminen budjettiin.

5 MICROSOFT EXCEL VISUAL BASIC APPLICATION

5.1 Microsoft Office Excel

Microsoft Excel on Microsoftin vuonna 1985 alun perin Macintoshille julkaistu, maailman laajimmin käytetty taulukkolaskentaohjelma. Tätä edesauttaa se, että useat raportointityökalut ja tietokannat mahdollistavat tiedonsiirron suoraan Microsoft Exceliin. Excel-sovelluksen helppokäyttöisyys sitoo sen niin useaan työtehtävään, kuin työntekijäänkin. (Rockoff 2014.)

Vaikka Microsoft Excelille povataan jatkuvasti seuraajaa ja työelämän digitalisaatio jatkaa nousukiitoaan, on Microsoft Excel edelleen taulukkolaskennan standardi. Vuonna 2016 Excelin saavuttaessa 30 vuoden iän oli sillä maailmanlaajuisesti 750 miljoonaa käyttäjää. Tämän perusteella onkin turvallista päätellä, että Excel ei ole vielä hetkeen väistymässä, sillä moni taulukkolaskentaa käyttävä on turvallisesti ankkuroinut itsensä Microsoft Exceliin. (Helsingin sanomat 2016.)

Excelin säilyvyyttä puoltavat myös verkkokurssien suosio. LinkedIn Learning Blogin -sivuilla on listattu vuoden 2020 LinkedInin kautta ostetuimmat verkkokurssit. Näistä kahdestakymmenestä kolme keskittyy Microsoft Exceliin. Näyttää siis siltä, että ammattilaiset jatkavat edelleen keskeisten taitojen terävöittämistä ja näihin näyttäisi kuuluvan edelleen Microsoft Office. (Brodnitz 2020.)

5.2 Visual Basic for Applications

Microsoft Excel Visual Basic for Applications, (VBA) sekä Visual Basic (VB), ovat toisiaan muistuttavia termejä, mutta ero on huomattava. Visual Basic on oma erillinen ohjelmansa ohjelmointia varten, eikä se ole riippuvainen Microsoft Excelistä, kuten Visual Basic for Applications (VBA) puolestaan on. VBA:n tarjoamat ominaisuudet on mahdollistettu käytettäväksi Microsoft Office sovellusten päällä. Näitä ominaisuuksia kutsutaan makroiksi. Makrot

mahdollistavat useiden rutiinitöiden automatisoinnin ja nopeuttamisen Microsoft Excelissä. Opinnäytetyöni toimii tässä esimerkkinä. (Shepherd 2006, ix.)

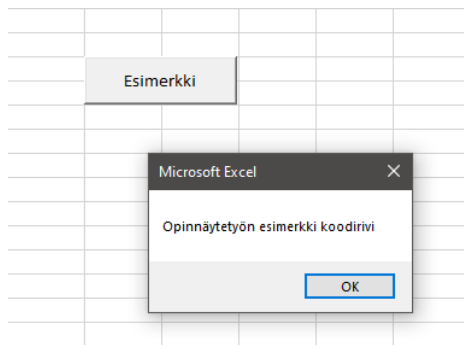
Uuden Excel makron luomiseen päädytään usein siitä syystä, että säästetään rutiinitöissä käytettyä aikaa, eikä koodin loppukäyttäjälle jää parhaassa tapauksessa muuta tehtävää kuin ohjelman käynnistäminen. Toinen yleisin syy Excel VBA -projektin aloittamiselle on se, että Excelistä ei itsessään löydy omaa toiminnallisuutta tiettyjen toimintojen suorittamiselle, mikä taas on mahdollista saavuttaa VBA-koodilla. (Merensalmi 2007, 2.) Nämä molemmat syyt ovat konkreettisia tilanteessa, minkä toimeksiantajayritys minulle antoi ratkaistavaksi.

VBA-kielessä on myös etuna se, että tietokoneelle ei tarvitse erikseen asentaa mitään. Jos tietokoneesta löytyy Microsoft Excel, siitä löytyy myös Visual Basic For Applications. Tästä syystä VBA:lla ei ole myöskään mahdollista luoda erillisiä sovelluksia. Visual Basicin käyttö vaatii aina Microsoft Excelin ja aktiivisen taulukon, josta ohjelmakoodin suoritus käynnistetään. (Gan 2019, 4.)

5.3 Esimerkki VBA-koodista

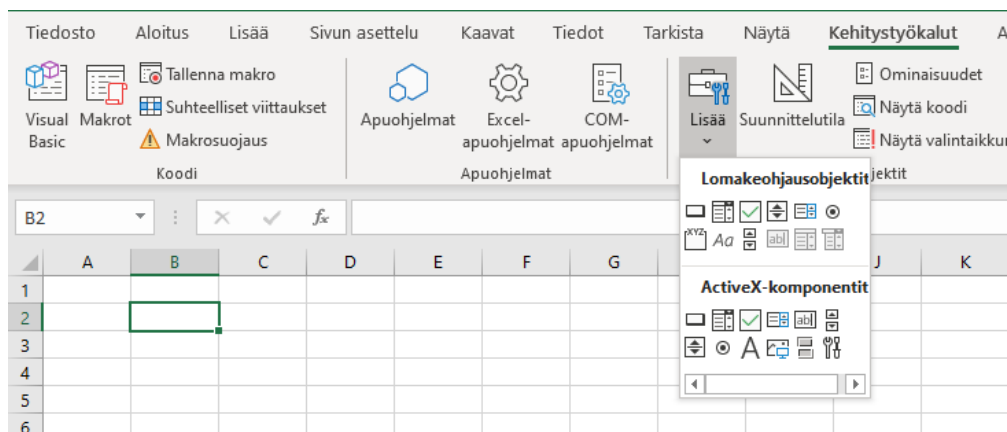
Päästäkseen alkuun Microsoft Excel Visual Basic for Applicationin käyttämisessä täytyy käyttäjän ensin asettaa Kehitystyökalut-välilehti näkyviin. Tämä onnistuu helpoiten Tiedosto -välilehdeltä. Polku on seuraava Tiedosto → Asetukset → Valintanauhan mukauttaminen → Kehitystyökalut. Kehitystyökalujen päälle kytkennän jälkeen, VBA on valmis käytettäväksi ja sen saa käyttöön ALT + F11 näppäinyhdistelmällä. Työkirja, jossa makroja aikoo käyttää, on kuitenkin oltava ”makrot käytössä” -tilassa.

VBA:lla on mahdollista luoda helposti esimerkiksi painike Excelin taulukkoon, joka luo viestikentän, johon ilmestyy kirjoittajan haluama teksti. Tein kuvan 1 mukaisen koodin, jossa Esimerkki -niminen näppäin luo viesti-ilmoituksen ”Opinnäytetyön esimerkki koodirivi”.



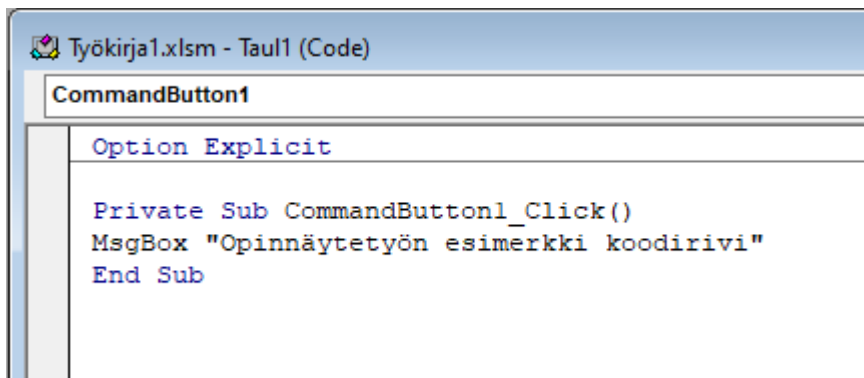
KUVA 1. Yksinkertainen VBA-suorite

Koodi on itsessään hyvin yksinkertainen. Ensiksi täytyy lisätä taulukkoon itse komentopainike kehitystyökalu-välilehdeltä, käyttäen Lisää-painiketta. Valinnan täytyy olla ActiveX-valikosta toimiakseen. (Kuva 2).



KUVA 2. VBA-komponenttivalikko

Tämän jälkeen, painikkeelle voidaan hiiren oikealla näppäimellä vaihtaa painikkeen oletusnimike käyttäen Ominaisuudet-valikkoa. Koodissani oletusnimike on "Esimerkki". Tästä päästään eteenpäin tuplaklikkaamalla painiketta, jolloin Excel avaa kyseiselle painikkeelle valmiin koodirivin VBA-editoriin. Tässä vaiheessa ei tarvitse, kuin lisätä koodiriville teksti "MsgBox "Opinnäytetyön esimerkki koodirivi"", kuten kuvassa 3. Tämän jälkeen koodi on suoritettavissa ja Esimerkki-painikkeella saadaan kuvan 1 mukainen tilanne.



```
Option Explicit

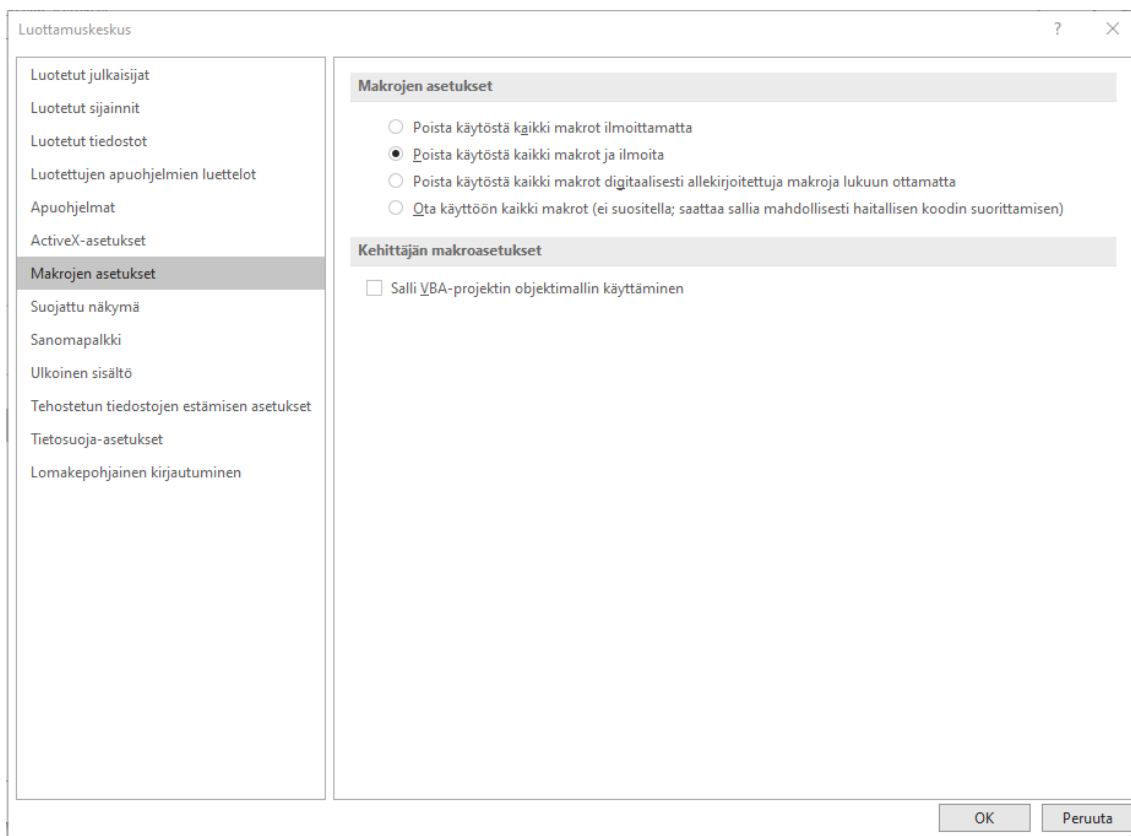
Private Sub CommandButton1_Click()
MsgBox "Opinnäytetyön esimerkki koodirivi"
End Sub
```

KUVA 3. Esimerkki koodirivit

5.4 Tietoturva

Microsoft Excelin Visual Basic Applicationilla ollessa omalla tavallaan hyvin yksinkertainen, sillä on mahdollista myös saada tietoturvaongelmia aikaan. Tuntemattomasta lähteestä ladatun Excel-makron avaaminen ja suorittaminen on riski ja pahimmassa tapauksessa se käyttäytyy kuin tietokoneen haittaohjelma. Makrokoodilla on nimittäin mahdollista avata sovelluksia kysymättä tai varastaa dataa käyttäjän tietokoneelta. Pelkästään makron käyttöönottaminen saattaa riittää sillä ohjelmakoodin suoritus voidaan käynnistää esimerkiksi pelkällä hiiren liikkeellä. (National Cyber Security Centre 2019.)

Suojautuminen tämän kaltaisilta Microsoft Excel -makro hyökkäyksiltä on helppoa. Tärkeimpänä asiana on pitää Microsoft Excelin suojausasetukset kunnossa. Asetukset täytyy asettaa siten, että Microsoft Excel poistaa aina tiedostoa avatessa makrot käytöstä ja ilmoittaa siitä, tällöin luotetusta lähteestä ladattu Excel-makro on silti suoritettavissa. Käyttäjä pääsee tarkastamaan asetuksen valitsemalla Microsoft Excelissä Tiedosto → Asetukset → Luottamuskeskus → Luottamuskeskuksen asetukset → Makrojen asetukset → Poista kaikki makrot käytöstä ja ilmoita (kuva 4). Tietoturvan takaamiseksi asetuksen täytyy olla lähes poikkeuksetta, tässä tilassa. (Microsoft Support n.d.)



KUVA 4. Makrojen käytön tietoturva-asetukset

6 TYÖN TOTEUTTAMINEN

6.1 Työn suunnittelu

Tämä kappale käsittelee ratkaistun ongelman alkuperäistilannetta ja sitä miten tekemäni VBA-ohjelma toimii. Lisäksi se miten, miksi ja missä järjestyksessä eri komentoja käytetään VBA-koodissani ja mikä on näiden komentojen päällimmäinen tarkoitus.

Ongelma tuli ilmi opinnäytetyön kartoituskeskustelussa yrityksen yhteyshenkilön kanssa. Tehtäväkseni valikoitui kuukausittaisen kiinteiden kulujen raportin kehittäminen, joka vaatii useita klikkauksia ja työläytensä vuoksi raportti saattaa jäädä vajavaiseksi. Yrityksen puolelta haluttiin, että työtehtävään saataisiin nopeutta, raportista helposti luettavan, sekä kustannuspaikkojen soluihin jossain määrin porautuvuutta, jotta raporttia olisi entistä helpompi tarkastella sen välittäessä myös mahdollisimman paljon tietoa.

Tuloslaskelma tileittäin (13 saraketta)/1000 €

Hallinto kiinteät kustannukset

Tulostusvaluutta	EUR			
	2020/7 Toteutunut	2020/7 Budjetti	2020/1 - 2020/7 Toteutunut	2020/1 - 2020/7 Budjetti
Tili	Hallinto	Hallinto	Hallinto	Hallinto
5720		1,00	1,00	1,00
5730		1,00	1,00	1,00
5735		1,00	1,00	1,00
Markkinointi		1,00	1,00	1,00
5800	1,00	1,00	1,00	1,00
5810	1,00	1,00	1,00	1,00
5815	1,00	1,00	1,00	1,00
5816	1,00	1,00	1,00	1,00
5821	1,00	1,00	1,00	1,00
5826	1,00	1,00	1,00	1,00
5830	1,00	1,00	1,00	1,00
5840	1,00	1,00	1,00	1,00
5841	1,00	1,00	1,00	1,00
5842	1,00	1,00	1,00	1,00
5855	1,00	1,00	1,00	1,00
5857	1,00	1,00	1,00	1,00
5865	1,00	1,00	1,00	1,00
5870	1,00	1,00	1,00	1,00
5880	1,00	1,00	1,00	1,00
5887	1,00	1,00	1,00	1,00
5890	1,00	1,00	1,00	1,00
5891	1,00	1,00	1,00	1,00
5892	1,00	1,00	1,00	1,00
5893	1,00	1,00	1,00	1,00
5894	1,00	1,00	1,00	1,00
5896	1,00	1,00	1,00	1,00
5897	1,00	1,00	1,00	1,00
5898	1,00	1,00	1,00	1,00
5899	1,00	1,00	1,00	1,00
5990	1,00	1,00	1,00	1,00
Hallinnon kulut	1,00	1,00	1,00	1,00
5905	1,00	1,00	1,00	1,00
Edustus	1,00	1,00	1,00	1,00
KIINTEÄT KULUT	99,99	100,00	1 999,99	1 999,99

KUVA 5. VISMA L7 -kiinteiden kulujen raportti

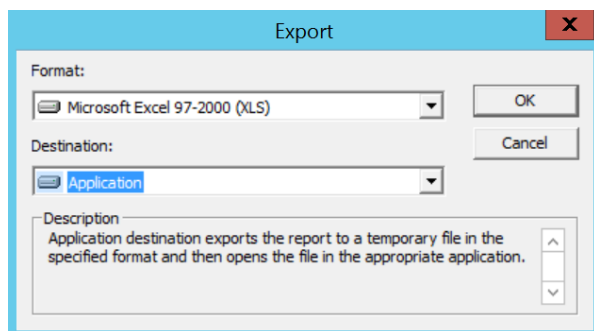
Alkuperäistilanteessa työntekijä haki ensimmäisenä Visma Liinos L7 -toiminnanohjausjärjestelmästä raportin yksittäin, jokaista raportin neljäätoista eri kustannuspaikkaa varten, etsi raportista loppusumman, kopioi ja liitti sen Excel raporttiin. Toiminnanohjausjärjestelmän raportti on nähtävissä kuvassa 5 ja Excel -työkirja kuvassa 6.

Fixed costs compared to budget					
Yritys X		Koja Oy			
		PERIOD		CUMULATIVE	
		Actual 6.2020	Budget 6.2020	Actual 1-6.2020	Budget 1-6.2020
FIXED COSTS	Kustannuspaikka 1	10	20	200	300
	Kustannuspaikka 2	11	21	201	301
	Kustannuspaikka 3	12	22	202	302
	Kustannuspaikka 4	13	23	203	303
	Kustannuspaikka 5	14	24	204	304
	Kustannuspaikka 6	15	25	205	305
	Kustannuspaikka 7	16	26	206	306
	Kustannuspaikka 8	17	27	207	307
	Kustannuspaikka 9	18	28	208	308
	Kustannuspaikka 10	19	29	209	309
	Kustannuspaikka 11	20	30	210	310
	Kustannuspaikka 12	21	31	211	311
	Kustannuspaikka 13	22	32	212	312
	Kustannuspaikka 14	23	33	213	313
	TOTAL	231	371	2 891	4 291

KUVA 6. Kiinteiden kulujen Microsoft Excel -raportti

6.2 Toteutus

Toteuttamisessa ensimmäinen asia oli saada raportti ulos toiminnanohjausjärjestelmästä, Microsoft Excelille käsiteltävässä muodossa. Raportin ollessa avattuna Visma L7 -sovelluksesta, ikkunan oikeasta ylälaidasta tiedoston vienti halutulla tavalla onnistui ”Export Report” -valinnalla. Valinta avaa valikon, jossa valintaoletuksina ovat Format: Adobe Acrobat (PDF) ja Destination: Disk File. Viedäkseen raportin Exceliin täytyy valita seuraavasti Format: Microsoft Excel 97–2000 Data Only (XLS) ja Destination: Application. Tämä valintaketju muokkaa tiedot Excel -tiedostoksi ja avaa sen sovelluksessa. Vienti-ikkuna nähtävissä kuvasta 7.



KUVA 7. Visma L7 raportin vienti-ikkuna

Tällä tavalla raportin sisältämät tiedot siirtyvät väliaikaiseen, tallentamattomaan Excel-taulukkaan. Raporttipohjan käyttäjä suorittaa tämän toimenpiteen, jokaisen kustannuspaikan raportille. Valmiiseen Microsoft Excel -tiedostoon on tehty Visual Basic Applicationilla ohjelmakoodi, joka hakee ja kopioi näistä edellä mainituista raportin sisältävistä Excel-työkirjoista halutun tiedon ja liittää ne raporttityökirjaan.

Automatisoitu ”kopioi ja liitä” -menetelmä mahdollistetaan käyttämällä Visual Basic Applicationin Workbooks-kokoelmaa. Workbooks-kokoelman objekteilla on mahdollista luoda ohjelmakoodi, joka viittaa yksi kerrallaan aukinasiin Excel-työkirjoihin. Esimerkiksi Workbooks(1) viittaa valmistelemani työkirjaan, joka sisältää suoritettavan ohjelmakoodin. Seuraava Workbooks(2) viittaa jo kiinteiden kulujen raportin ensimmäiseen kustannuspaikkaan. Tästä syystä on hyvin tarkkaa, missä järjestyksessä raportit tuodaan Microsoft Exceliin Visma Liinos L7:sta, sillä ohjelmakoodi ei lue työkirjojen nimiä vaan toimii pelkästään Workbooks-kokoelmaa hyödyntäen. (Merensalmi 2007, 224.)

Lisäksi Excel-työkirjaan hyödynnettiin hyvin paljon ActiveX-komponenttien komentopainiketta. ActiveX-komponentit ovat hyvin yleisiä, sillä ne toimivat niin verkon, kuin tietokoneenkin kautta. Esimerkiksi juuri ActiveX-komentopainikkeen taakse on mahdollista luoda koodi, joka laukaisee minkä tahansa exe-tiedoston tietokoneelta. (Dodge,Stinson 2010.) Ohjelmakoodissani komentopainikkeita käytetään vain koodirivien suorittamiseen. Työkirjassa on selkeästi esillä kaksi komentopainiketta, niistä molemmat toimivat klikkauksella eli CommandButtonX_Click -komennolla. Ensimmäinen komentopainike kloonaa taulukon seuraavaa kuukautta varten, jolloin käyttäjälle jää tässä tehtäväksi vain

välilehden nimen vaihtaminen. Toisen komentopainikkeen tehtävä on monimutkaisempi, se tyhjentää nykyisen taulukon, kopioi tiedot taustalle avatuista työkirjoista, värittää osan komentopainikkeista sekä jakaa tiettyjä solualueita tuhannella. Kuvan 8 oikeassa ylälaidassa luomani komentopainikkeet. Loput työkirjan noin 30 komentopainiketta, on tarkoitettu työkirjan kohdistamiseen tiettyyn soluun ja nämä mahdollistavat työkirjan porautuvuuden. (Shepherd 2006, 107,165.)

		PERIOD		CUMULATIVE	
		Actual	Budget	Actual	Budget
		9.2020	9.2020	Since January	Since January
FIXED COSTS	<input type="checkbox"/> Kustannuspaikka 1				
	<input type="checkbox"/> Kustannuspaikka 2				
	<input type="checkbox"/> Kustannuspaikka 3				
	<input type="checkbox"/> Kustannuspaikka 4				
	<input type="checkbox"/> Kustannuspaikka 5				
	<input type="checkbox"/> Kustannuspaikka 6				
	<input type="checkbox"/> Kustannuspaikka 7				
	<input type="checkbox"/> Kustannuspaikka 8				
	<input type="checkbox"/> Kustannuspaikka 9				
	<input type="checkbox"/> Kustannuspaikka 10				
	<input type="checkbox"/> Kustannuspaikka 11				
	<input type="checkbox"/> Kustannuspaikka 12				
	<input type="checkbox"/> Kustannuspaikka 13				
	<input type="checkbox"/> Kustannuspaikka 14				
	TOTAL				

Kloonaa taulukko seuraavaa kuukautta varten
HUOM! Muista vaihtaa nimi

Tyhjennä taulukko, kopioi tiedot aukiolevista Visman Excel -raporteista

KUVA 8. Raportin uudistettu ulkoasu

6.3 Lopputulos

Tässä kappaleessa esittelen raporttityökirjan lopullisen ulkoasun ja ohjelmakoodin, jotta tulevaisuudessa ohjelmakoodin korjaaminen on tarvittaessa mahdollista jonkun muun toimesta. Ohjelmakoodin kehitysvaiheessa törmättiin useisiin ongelmiin, jolloin työn korjaamisen kului tunteja. Tämä osio käsittelee siis pelkästään ohjelmakoodin ja Excel-taulukon lopputulosta.

Ohjelmakoodi on kirjoitettu kokonaan eksplisiittisen määrittelyn eli Option Explicit -koodin alle. Tämä vaikuttaa ainoastaan siten, että jokainen muuttuja on määriteltävä etukäteen, kuten kuvassa 9. Tätä määrittelyä käytetään ohjelmakoodin selkeyttämiseksi, sekä muuttujien välisten ristiriitojen eliminoimiseksi. (Shepherd 2006, 13.)

KUVA 9. Eksplisiittinen määrittely

6.3.1 ”Kloonaa” komentopainike

Raportin helppokäyttöisyyden vuoksi on taulukkoon ensimmäisenä luotu ”kloonaa”-komentopainike. Painike luo uuden välilehden taulukkoon, samalla nimellä kuin edellisen kuukauden välilehti kopioiden ohjelmakoodin mukanaan. Esimerkiksi tammikuu – heinäkuu 2020 -välilehden viereen ilmestyy tammikuu – heinäkuu 2020 (2) -välilehti. Käyttäjälle jää jäljelle vain nimenvaihto seuraavaksi kuukaudeksi, kuten tammikuu – elokuu 2020. Tällä komentopainikkeella ei ole siis muuta toiminnallisuutta siitä syystä, mikäli tähän ohjelmakoodiin olisi lisätty suoritteita olisi ne tapahtuneet edellisen kuukauden laskentataulukossa. Tämä puolestaan olisi sekoittanut kuukausien järjestystä. ”Kloonaa”-komentopainikkeen koodi kuvassa 10.

```
Private Sub CommandButton1_Click()  
  
    ThisWorkbook.ActiveSheet.Copy Before:=ThisWorkbook.ActiveSheet  
  
End Sub
```

KUVA 10. ”Kloonaa” -painikkeen ohjelmakoodi

6.3.2 ”Kopioi” komentopainike

Seuraava ohjelmakoodin osa eli CommandButton2_Click pitää sisällään, lähes kaikki toiminnot ja tapahtumat, mitä koko Excel-ohjelmakoodi sisältää, pois lukien lopussa käsiteltävät useat yksinkertaiset komentopainikkeiden Click-toiminnot. Kuvasta 11 selviää ”Kopioi”-painikkeen ensimmäisenä suoritettava vaihe.

```

Private Sub CommandButton2_Click()
    Range("F8:I21").Clear
    Range("BC1:BM600").Clear
    Range("F7").value = Range("F7").value + 1
    Range("G7").value = Range("G7").value + 1

    Application.Wait (Now + TimeValue("0:00:01"))

```

KUVA 11. Osa "Kopioi" -painikkeen ohjelmakoodia

Range-objektilla on mahdollista kohdistaa suorittaminen tietylle solualueelle. Ilman Range-objektia ei ole mahdollista muokata useita soluja kerrallaan tai solualueita. Clear-menetelmä kuuluu Range-objektin ominaisuuksiin ja nimensä mukaisesti tyhjentää solualueen sisällön sekä se poistaa myös kommentit ja muotoilut solualueelta, jotteivät edellisen kuukauden merkinnät siirry seuraavalle kuukaudelle, mikäli sellaisia on tehty. (Shepherd 2006, 161–163). Solualue F8:I21 kattaa raportin tiedot, jotka voi nähdä kuvasta 8. Solualue BC1:BM600 puolestaan kattaa oikeammalla taulukossa olevat kunkin kustannuspaikan raportit, kuten kuvassa 12.

Kustannuspaikka				
Kust.paikan osa 1	54.00	3 873.00	3.00	4 444.00
Kust.paikan osa 2	545.00	3.00	833.00	6.00
Kust.paikan osa 3	857 676.00	3.00	12.25	65 635.00
Kust.paikan osa 4	545.00	78.00	4.67	8.68
Kust.paikan osa 5	445.00	87.00	3.00	8.00
Kust.paikan osa 6	66 576.00	78.00	78 373.00	6 546.00
Kust.paikan osa 7	378 363.00	3.00	3.49	45.00
Kust.paikan osa 8	3 378.00	33.00	3.00	63.00
Kust.paikan osa 9	83 783.00	0.25	6 454 654.00	546.00
KIINTEÄT KULUT	645.00	456 456.00	6 456.00	46 546.00

KUVA 12. Kustannuspaikan raportti solualueelta BC1:BM600

Tämän vaiheen jälkeen komentopainike suorittaa kopioinnin jokaiselle kustannuspaikalle erikseen Workbooks-objektia hyödyntäen. Kuvan 13 ohjelmakoodi viittaa Workbooks(2)-työkirjassa sijaitsevaan välilehteen nimeltä Taul1. Suomenkielisessä Excelissä välilehden nimi on oletuksena aina Taul1, koska raportteja ei tallenneta tietokoneelle Sieltä ohjelmakoodi kopioi solualueen A3:E12 ja liittää sen soluun BB92.

Kuvassa 13 toisessa koodikappaleessa on Application.Wait-komento, kuten useat VBA:n ominaisuudet, myös Application.Wait on hyvin yksiselitteinen. Se käskee ohjelmakoodia odottamaan ennalta määrätyn ajan. Tähän samaan sarjaan kuuluvat myös Pause- ja Loop -ominaisuudet. Wait-ominaisuudella on mahdollista odottaa ohjelmakoodia, joko aikamääreitä tai kellon aikoja. Jos ohjelmakoodissani olisi "0:00:01" tilalla esimerkiksi "22:00:00" pysäyttäisi se toiminnan iltakymmeneen asti. Kuitenkin olen asettanut useaan kohtaan koodissani sekunnin odotuksen, jotta välttyttäisiin virheiltä, joita on mahdollista tapahtua usean eri toiminnon käydessä samaan aikaan. (Microsoft 2019.)

```
Workbooks(2).Worksheets("Taul1").Range("A3:E12").Copy _
ThisWorkbook.ActiveSheet.Range("BB92")

Application.Wait (Now + TimeValue("0:00:01"))

Range("BC101").Copy _
Range("F8")

Range("BD101").Copy _
Range("G8")

Range("BE101").Copy _
Range("H8")

Range("BF101").Copy _
Range("I8")
```

KUVA 13. Tiettyä kustannuspaikkaa koskeva koodi

Seuraavissa koodinkappaleissa käytetään Copy-ominaisuutta. Koko luomani ohjelma koodin ydin on Range-objektin Copy-ominaisuus. Kuvasta 13 voidaan huomata Copy-ominaisuuden löytyvän useilta objekteilta, kuten myös Workbooks-objektilta. Kuvan 13 neljä viimeistä osaa kopioivat solun tiedot taulukon oikeasta reunasta, kustannuspaikkojen omista raporteista ja liittävät ne taulukon kustannuspaikkojen koontisivulle. Tällä mahdollistetaan mahdollisimman tarkka raportointi lukujen ollessa suoraan toiminnanohjausjärjestelmän raportista ilman pyöristyksiä tai yhteenlaskuja. (Merensalmi 2007, 348.)

Kustannuspaikkojen kopiointi taulukoista raportointitaulukkoon noudattaa tätä samaa ohjelmakoodia. Ainoastaan solualueet muuttuvat. Poikkeuksena tästä on kaksi kustannuspaikkaa, joiden raportit koostuvat useammista osista ja ne ovat

toiminnanohjausjärjestelmässä jaettu kuukausi- ja vuosiraportteihin. Tästä syystä ne avataan toiminnanohjausjärjestelmän kautta kahteen eri Excel-työkirjaan. Se on myös syy sille, että ohjelmakoodissa Workbooks-objektilla viitataan yhteensä 16 eri työkirjaan, vaikka kustannuspaikkoja on 14. Kuvan 14 koodi käsittää siis yhden kustannuspaikan raportit, vaikka se sisältää kaksi eri Workbooks-objektin viittausta.

```
'kk
Workbooks (6) .Worksheets ("Taul1") .Range ("A3:A8") .Copy _
ThisWorkbook.ActiveSheet.Range ("BB176")

Application.Wait (Now + TimeValue ("0:00:01"))

Workbooks (6) .Worksheets ("Taul1") .Range ("J3:K8") .Copy _
ThisWorkbook.ActiveSheet.Range ("BC176")

Application.Wait (Now + TimeValue ("0:00:01"))
'vuosi-
Workbooks (7) .Worksheets ("Taul1") .Range ("J3:K8") .Copy _
ThisWorkbook.ActiveSheet.Range ("BE176")

Application.Wait (Now + TimeValue ("0:00:01"))

Range ("BC181") .Copy _
Range ("F12")

Range ("BD181") .Copy _
Range ("G12")

Range ("BE181") .Copy _
Range ("H12")

Range ("BF181") .Copy _
Range ("I12")
```

KUVA 14. Poikkeava ohjelmakoodi kahden erityisen kustannuspaikan raportille

6.3.3 Tuhansilla jakaminen

Visma L7 -toiminnanohjausjärjestelmästä Microsoft Exceliin viedyssä kustannuspaikkaraportissa kuukauden luvut ovat osa täsmällisiä lukuja ja osa valmiiksi tuhansilla jaettuja eli luvussa yksi euro vastaa tuhatta euroa. Tämän takia osa kustannuspaikkaraporttien tiedoista on jaettava tuhannella, jotta luvut saadaan täsmäämään Microsoft Exceliin luomaani raportti- ja makrotyökirjaan.

Makrotyökirjassani on jokaiselle kustannuspaikalle yksilöllinen koodi tuhansilla jakamiselle, sillä koodi jakaa tuhannella tietyt ennalta määrittelemäni solualueet,

koska kustannuspaikkojen tiedot liitetään eri soluriveille ja ne sijaitsevat eri sarakkeissa. Lisäksi ennen kustannuspaikkojen tuhansilla jakoa on ohjelmakoodi, joka jakaa työkirjan pääraporttiosuuden luvut, sillä lukujen kopioiminen kustannuspaikkojen yksilöllisistä raporteista tapahtuu ennen tuhansilla jakamista yksilöllisissä raporteissa.

Ohjelmakoodin alussa määritetään jokaiselle kustannuspaikalle oma muuttujakirjain. Kuvassa 15 ensimmäisen kustannuspaikan muuttuja on y ja seuraavan d. Ensimmäisellä koodirivillä muuttuja y määritetään kokonaisluvuksi käyttäen numeerisena tietotyyppinä Integer-tyyppiä.

Tietotyyppiä valittaessa lukujen koolla on väliä. Pienemmät luvut määritetään yleisesti Integer-tyyppisiksi ja pidemmät taas Long-tyyppisiksi. Raportin lukujen ollessa yksinkertaisia, pienellä määrällä desimaaleja Integer-tyyppi sopii koodin suorittamiseen tässä tapauksessa hyvin, sekä laskutoimitukset onnistuvat ohjelmakoodin suorituksessa nopeammin. (Shepherd 2006, 12, 19.)

```
'lukujen jakaminen tuhannella

Dim y As Integer

For y = 8 To 21
    Cells(y, 6) = Cells(y, 6).value / 1000
    Cells(y, 8) = Cells(y, 8).value / 1000
Next y

'Kustannuspaikka X
Dim d As Integer

For d = 8 To 17

Cells(d, 55) = Cells(d, 55).value / 1000
Cells(d, 57) = Cells(d, 57).value / 1000
Next d
```

KUVA 15. Kahden eri kustannuspaikan tuhansilla jakamisen ohjelmakoodi

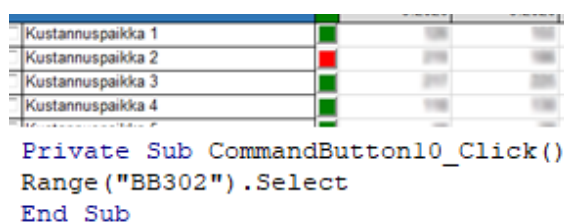
Seuraavaksi ohjelmakoodissa käytetään niin sanottua kiinteää silmukkaa. Kustannuspaikan yksilöllisen raportin rivimäärä määrittää kierrosten lukumäärän, jolloin kiinteän For – Next -silmukan käyttö on mahdollista. Kuvan 15 ylemmässä ohjelmakoodissa For-silmukka aloittaa riviltä 8 ja lopettaa riville 14. Ohjelmakoodi käy läpi siis neljätoista eri riviä ja siten silmukan kierrokset määrittävät mihin soluun silmukka kohdistuu. (Merensalmi 2008, 87.)

Jotta silmukkakoodi osaa lukea muuttujaa soluarvona, täytyy sarakkeen kirjaimet muuttaa numeroiksi. Tässä tapauksessa kuvan 14 alemman ohjelmakoodin numero 55 viittaa sarakkeeseen BC ja numero 57 sarakkeeseen BE. (Vishal Monpara 2016.)

Ohjelmakoodissa soluun viitataan siis (d, 55) lauseella. Ensimmäisellä kierroksella kyseinen solu on BC8 ja seuraavalla taas BC9 ja niin edelleen. Ohjelmakoodi määrittää solun arvoksi solun arvon jaettuna tuhannella. Tämä tapahtuu jokaisella silmukan kierroksella niin sarakkeelle BC, kuin myös sarakkeelle BE. Solujen tuhannella jakamisen jälkeen silmukka siirtyy seuraavalle kierrokselle komennolla Next d. Silmukka jatkaa kierroksiaan, kunnes se saavuttaa viimeisen kierroksen ja jatkaa seuraavan kustannuspaikan tuhansilla jakamisen ohjelmakoodiin. (Merensalmi 2007, 87–88.)

6.3.4 Porautuminen ja porauspainikkeiden värittäminen

Yhtenä toimeksiantajan toiveena oli lisätä heidän Microsoft Excel -raporttiin porautuvuutta. Tämä mahdollistettiin kopioimalla jokaisen eri kustannuspaikan tiedot omalle solualueelleen päätyökirjaan ja lisäämällä työkirjan etusivulle komentopainikkeita, jotka keskittävät Excelin haluttuun soluun. Kyseiset komentopainikkeet näkyvät kuvassa 16 jokaisen kustannuspaikan vasemmalla puolella ja ovat väriltään vihreitä tai punaisia.



KUVA 16. Porautumisen ohjelmakoodi sekä porauspainikkeet

Komentopainikkeen takana oleva ohjelmakoodi on erittäin yksinkertainen. Koodissa käytetään jälleen CommandButtonX_Click-komentoa, sekä kohdistamiseen käytettävää Select-objektia. CommandButtonX_Click:n tarkoitus on, että ohjelmakoodin suoritus alkaa painettaessa komentopainiketta. Select-objekti taas suorittaa kohdistamisen ennalta määriteltyihin soluihin, jotka ohjelmakoodimme tapauksessa löytyvät Excel-työkirjan oikeasta laidasta ja

sisältävät kustannuspaikkojen yksilölliset raportit. Jokaisen kustannuspaikan yksilöllisen raportin alalaidassa on täysin samalla periaatteella toimiva komentopainike, joka vie käyttäjän takaisin työkirjan etusivulle ja on nimetty Takaisin-otsikolla. (Shepherd 2006, 107,165.)

Työkirjan etuaukeamalla olevat väritetyt komentopainikkeet kertovat kuukauden budjetin, joko alittuneen tai ylittyneen verrattuna kuukauden toteumaan. Värittäminen mahdollistetaan If-Then-Else-End If -ehtolausekkeella käyttäen hyödyksi RGB-koodeja värittämiseen. (Merensalmi 2007, 81.)

If-Then-Else-End If -ehtolauseke on hyvin yksiselitteinen. Jos jokin If on tosi, sitten koodi toteuttaa toiminnon Then, muussa tapauksessa Else-ohjelmakoodi suorittaa vaihtoehdoisen toiminnon. Yksinkertaistettuna värityskoodi tarkastaa, onko toteuma suurempi kuin budjetti ja mikäli näin on, se värittää komentopainikkeen punaiseksi. Toteuman alittaessa budjetin ohjelmakoodi värittää komentopainikkeen taas vihreäksi. (Shepherd 2006, 38.)

Ohjelmakoodin Activesheet-objekti viittaa aktiiviseen taulukkosivuun, joka on Excelin raporttityökirjassa sen kuukauden välilehti, joka on kyseisellä hetkellä työn alla. Ohjelmakoodi siis muuttaa aktiivisen työkirjan komentopainikkeen numero 3 taustaväriä hyödyntäen värikoodeja (Merensalmi 2007, 262.)

RGB-värikoodilla eli red-, green-, blue-koodeilla on mahdollista määrittää tietyt värit halutuille objekteille. Koodi (255, 0, 0) määrittää värin mahdollisimman punaiseksi sekoittamatta värejä, kun taas (0, 255, 0) määrittäisi sen vihreäksi ja (0, 0, 255) taas siniseksi. Koodissani käytin siis täysin punaista (255, 0, 0), sekä tummemman vihreää (0, 128, 0) värikoodia. (Microsoft, 2018). Kuvassa 17 näkyy porauspainikkeiden värittämisen valmis ohjelmakoodi.

```
'Nappien väritys
    'Kustannuspaikka 1
    If [F8] > [G8] Then
        ActiveSheet.CommandButton3.BackColor = RGB(255, 0, 0)
    Else
        ActiveSheet.CommandButton3.BackColor = RGB(0, 128, 0)
    End If
```

KUVA 17. Komentopainikkeiden värien määrittämien ehtolausekkeella

6.3.5 Työkirjan etuaukeaman selkeyttäminen

Toimeksiantaja halusi myös mahdollisuuksien mukaan, että työkirjan ulkoasua parannettaisiin sekä selkeytettäisiin. Ohjelmakoodin laatimisen yhteydessä muutin työkirjasta fonttia ja sarakkeiden kokoja, sekä lisäsin yleisiä ohjeita sekä lisäsin ActiveX-valintaruutuja. Näin raportteja avatessa toiminnanohjausjärjestelmästä pysyy selkeänä, mitä raportteja taustalle on avattu esimerkiksi keskeytyksen tapahtuessa.

Alkuperäisen ja uudistetun ulkoasun välisen eron voi nähdä kuvasta 18. Uudistetussa eli kuvan 18 alemmassa työkirjassa kustannuspaikkojen oikealta puolelta löytyvät edellä mainitut valintaruudut. FIXED COSTS -valintaruudun muistiinpanoihin on lisätty toiminnanohjausjärjestelmästä tuotavien raporttien avaamisjärjestys. Työkirjaan ylälaitaan on lisätty punaisella tekstillä huomautus ohjelmakoodin käytöstä, ja sitä edeltävien toimenpiteiden tekemiseen. Lisäksi ”kloonaa”- ja ”kopioi”-painikkeiden yläpuolelle on kirjoitettu painikkeiden tarkoitus.

Fixed costs compared to budget					
Yritys X	Kustannuspaikka	Koju Oy		CUMULATIVE	
		PERIOD			
		Actual 6.2020	Budget 6.2020	Actual 1-6.2020	Budget 1-6.2020
	Kustannuspaikka 1	10	20	200	300
	Kustannuspaikka 2	11	21	201	301
	Kustannuspaikka 3	12	22	202	302
	Kustannuspaikka 4	13	23	203	303
	Kustannuspaikka 5	14	24	204	304
	Kustannuspaikka 6	15	25	205	305
	Kustannuspaikka 7	16	26	206	306
	Kustannuspaikka 8	17	27	207	307
	Kustannuspaikka 9	18	28	208	308
	Kustannuspaikka 10	19	29	209	309
	Kustannuspaikka 11	20	30	210	310
	Kustannuspaikka 12	21	31	211	311
	Kustannuspaikka 13	22	32	212	312
	Kustannuspaikka 14	23	33	213	313
	TOTAL	231	371	2 891	4 291

Yritys X	Kustannuspaikka	PERIOD		CUMULATIVE	
		Actual	Budget	Actual	Budget
		9.2020	9.2020	Since January	Since January
	Kustannuspaikka 1				
	Kustannuspaikka 2				
	Kustannuspaikka 3				
	Kustannuspaikka 4				
	Kustannuspaikka 5				
	Kustannuspaikka 6				
	Kustannuspaikka 7				
	Kustannuspaikka 8				
	Kustannuspaikka 9				
	Kustannuspaikka 10				
	Kustannuspaikka 11				
	Kustannuspaikka 12				
	Kustannuspaikka 13				
	Kustannuspaikka 14				
	TOTAL				

Kloonaa taulukko
 seurauksena kuitkaulla varien
 HUOM! Muista valita nimi

Tyhjennä taulukko, kopioi
 tiedot aiioleimista Visman
 Excel-raportista

Kloonaa

Kopioi

KUVA 18. Alkuperäinen, sekä uudistettu työkirja.

7 POHDINTA

Toimeksiantajan antamat tavoitteet pystyttiin täyttämään, kiinteiden kulujen raportointi helpottui huomattavasti ja Excel -tiedoston sisältämä data raportoinnillisessa mielessä lisääntyi. Yhden painikkeen avulla tiedosto hakee tiedot avatuista työkirjoista, luo niistä omat pienet taulukot porautuvuuteen, muuttaa raportin kuukaudet kopioituun välilehteen, sekä porautuvuuspainikkeet kuvaavat väreillä toteuman ja budjetin välistä suhdetta. Ohjelmallani sain ratkaistua kaikki pyynnöt, joita toimeksiantaja minulle alkuvaiheessa esitti.

Opinnäytetyön raportoinnissa lukijalle selviää olennaiset asiat Microsoft Excelin Visual Basic Applicationista, muun muassa tietoturvasta ja VBA:n yleisestä käytämisestä. Opinnäytetyössä ohjelmakoodin kirjoittamiseen kului myös hyvin paljon aikaa. Tavoitteenani oli kirjoittaa ohjelmakoodi mahdollisemman yksinkertaisesti, siten että sitä on mahdollista korjata ja lukea perehtymättä syvällisemmin VBA:n toimintaan. Koska tämä opinnäytetyö ja ohjelmakoodin luominen oli minulle henkilökohtaisesti ensimmäinen itseluomani Excel-makrokokonaisuus, halusin sen olevan kaikkea mitä toimeksiantaja pyysi ja siten, että olen luonut sen omalla osaamisellani.

Joulukuussa 2020 pidetyssä tapaamisessa toimeksiantajan kanssa esittelin valmiin Excel -makrotyökirjan ja sen toiminnan. Tapaamisessa oli minun ja yrityksen yhteyshenkilön lisäksi kaksi muuta osallistujaa etäyhteydellä. Ohjelmani nähneet yrityksen työntekijät pitivät näkemästään ja uskoivat tästä olevan hyötyä pitkällekin eteenpäin ja muihin raportteihin, mahdollisesti jopa järjestelmiin jalostettuna. Lisäksi selvisi, että yrityksellä käytössä oleva ohjelmistorobotti olisi mahdollista ohjelmoida avaamaan jokaisen kustannuspaikan raportti itsenäisesti. Tällöin raportoinnin aikaakuluttavin vaihe saataisiin myös automatisoitua.

Koen, että opinnäytetyöni on hyvä esimerkki siitä, että vaikka Microsoft Excel on vanha sovellus, jolle povataan jatkuvasti seuraajaa, on siitä edelleen paljon hyötyä käytettynä oikein, oikeisiin asioihin. Kiinteiden kulujen raportointi ei suinkaan ole ainoa käyttökohde Excel-ohjelmoinnille. Kirjoittamani ohjelmakoodin tapaisella ratkaisulla voi luoda lisää makroja, jotka noutavat tietoa

muista työkirjoista ja luovat niistä haluttuja kokonaisuuksia. Opinnäytetyöni on yksinkertainen esimerkki Microsoft Officen Visual Basic For Applications -sovelluksen käyttömahdollisuuksista.

Toivon, että lukemalla opinnäytetyöni yrityksen työntekijät pystyvät tulevaisuudessa rakentamaan kuvan itselleen siitä, mitä toimintoja olisi mahdollista automatisoida ja mistä yritys hyötyisi niin ajallisesti kuin taloudellisestikin.

LÄHTEET

Brodnitz, D. 2020. The 20 Most Popular LinkedIn Learning Courses of the Year. Luettu 7.10.2020. <https://www.linkedin.com/business/learning/blog/top-skills-and-courses/fy20-most-popular-courses-of-the-year-hero>

Dodge, S. & Stinson, C. 2013). Microsoft Excel 2013 Inside Out. 1st edition. Boston, USA: Microsoft Press.

Gan, G. 2017. An Introduction to Excel VBA Programming: with Applications in Finance and Insurance. 1st edition. [Online]. Boca Raton: CRC Press.

Helsingin Sanomat. 2016. Taulukkolaskenta-ohjelma Excel täytti 30 vuotta, mutta mistä löytyisi uuden ajan Excel?. Luettu 2.10.2020. <https://www.hs.fi/paakirjoitukset/art-2000002877978.html>

Merensalmi, J. 2007. Excel VBA yrityskäytössä. Jyväskylä: Docendo.

Microsoft Docs. 2018. RGB Function. Luettu 22.12.2020. <https://docs.microsoft.com/en-us/office/vba/language/reference/user-interface-help/rgb-function>

Microsoft Docs. 2019. Application.Wait method (Excel). Luettu 5.12.2020. <https://docs.microsoft.com/en-us/office/vba/api/excel.application.wait>

Microsoft Support. N.d. Change macro security settings in Excel. Luettu 17.1.2021. <https://support.microsoft.com/en-us/office/change-macro-security-settings-in-excel-a97c09d2-c082-46b8-b19f-e8621e8fe373>

National Cyber Security Centre. 2019. Macro Security for Microsoft Office (2019 Update). Luettu 17.1.2021. <https://www.ncsc.gov.uk/guidance/macro-security-for-microsoft-office>

Rockoff, L. 2015. Microsoft Excel. Boston: Cengage Learning.

Shepherd, R. & Lammi, O. 2006. Excel-ohjelmointi: tehokas hallinta. Helsinki: Readme.fi.

Vishal Monpara. 2016. Excel Column Letter to Number Quick Reference. Luettu 11.12.2020. <https://www.vishalon.net/blog/excel-column-letter-to-number-quick-reference>

Visma.fi. 1 N.d. Ohjelmistoyhtiö Visma. Luettu 15.10.2020. <https://www.visma.fi/tietoa-vismasta/>

Visma.fi. 2 N.d. Toiminnanohjauksen järjestelmä Visma L7. Luettu 15.10.2020. <https://www.visma.fi/ohjelmistoratkaisut/visma-l7/>