

Excel VBA ohjelmoitu osakeportfolion analyysityökalu

Olli Eskola



Tekijä(t) Olli Eskola	
Koulutusohjelma Finanssi- ja talousasiantuntijan koulutusohjelma	
Opinnäytetyön otsikko Excel VBA ohjelmoitu osakeportfolion analyysityökalu	Sivu- ja liitesivumäärä 43 + 1
<p>Opinnäytetyön aiheena on luoda osakeportfolion analysointiin työkalu, sekä raportoida siihen liittyvistä aihealueista ja vaiheista. Käsiteltäviä aiheita ovat muun muassa osakkeet, sijoittaminen, Excel sekä Visual Basic for Applications (VBA) -ohjelmointi. Työssä kuvataan myös ohjelmistokehityksen eri vaiheita, kun ohjelmoidaan rajapintaa hyödyntävää analyysityökalua.</p> <p>Työkalun tavoitteena on tarjota käyttäjälle automatisoitu vaihtoehto osakeportfolion analysointiin. Automatisoidulla toiminnalla pyritään muun muassa vähentämään analysointiin vaadittua aikaa. Manuaalisen työn minimointi on yksi keinoista, joilla analyysityökalun tavoitteisiin voidaan päästä.</p> <p>Toiminnallisena opinnäytetyönä raportti on jaettu kahteen osaan. Ensimmäinen osa käsittelee analyysityökalun sisältämien aiheiden tietoperustaa. Näitä aiheita ovat muun muassa osakkeiden analysointiin käytettävät tunnusluvut, tuoton ja riskin teoria, sekä VBA -ohjelmointikieli.</p> <p>Toinen osa on toiminnallinen osuus, jossa käydään läpi vaiheittain analyysityökalun rakentamista alusta loppuun. Ensimmäisenä vaiheena on työkalun tavoitteiden asettaminen. Toisessa vaiheessa analyysityökalun toteuttamista suunnitellaan tavoitteiden pohjalta. Kolmas ja viimeinen vaihe käsittelee teknistä toteuttamista. Tätä on avattu lukijalle koodiesimerkkien avulla, joita on selvennetty tekstissä. Kaikki kolme vaihetta, sekä työkalun testaus, perustuvat spiraaliseen ohjelmiston kehitysmalliin.</p> <p>Lopullinen versio osakeanalyysityökalusta täytti sille asetetut tavoitteet. Sen hyödyntäminen on nopeampaa verrattuna esimerkiksi verkkosivujen selaamiseen. Tavoitteet saavutettiin seuraamalla spiraalisen kehitysmallin vaiheita.</p>	
Asiasanat osakkeet, sijoittaminen, portfolio, VBA, ohjelmistokehitys	

Sisällys

1	Johdanto	1
1.1	Opinnäytetyön tavoitteet ja rajaukset	1
1.2	Opinnäytetyön rakenne	1
2	Osakkeet ja sijoittaminen	3
2.1	Osakesijoittaminen.....	4
2.2	Osakeportfolio.....	5
2.3	Tunnusluvut	6
2.3.1	Volatiliteetti.....	7
2.3.2	Kokonaistuottoprosentti.....	7
2.3.3	P/E.....	8
3	Excel VBA.....	9
3.1	Makro.....	9
3.2	Muuttujat	10
3.3	Datatyypit.....	10
3.4	Ohjausrakenteet.....	11
3.5	UserForm.....	13
3.6	Ohjelmointirajapinta	14
3.6.1	REST rajapinta ja tiedon hakeminen	14
3.7	Kirjasto – VBA-JSON Converter.....	15
4	Portfolioanalyysityökalun tekeminen.....	17
4.1	Spiraali kehitysmalli	17
4.2	Työkalun tavoitteet.....	18
4.3	Työkalun suunnittelu	20
4.4	Toteutus.....	27
4.4.1	Työkalun käyttöliittymä.....	27
4.4.2	Datan noutaminen rajapinnasta.....	28
4.4.3	Lomakkeet	30
4.4.4	Aliohjelmat laskuihin ja päivityksiin.....	33
4.5	Toteutuksen testaaminen.....	36
5	Pohdinta.....	38
5.1	Analyysityökalun hyödyt.....	38
5.2	Oman oppimisen arviointi.....	39
	Lähteet	41
	Liitteet.....	44
	Analyysityökalun lopullinen käyttöliittymä	44

1 Johdanto

Osakesijoittamiseen liittyvän informaation määrä on kasvava. Sijoittamiseen liittyvien palveluiden tarjoamat alustat ja työkalut eivät vastaa kaikkien osakesijoittajien tarpeita. Turha tieto, monimutkaiset ohjelmistot ja maksullisuus ovat asioita, joihin osakesijoittaja saattaa helposti törmätä tiedonhaussa. Ei-ammattimaisen sijoittajan näkökulmasta tilanne ei ole edullinen. Tarpeettomia tietoja sisältävät internetsivut vievät aikaa, kun tiedon määrästä haluttaisiin nähdä vain murto-osa. Informaation määrä saattaa myös luoda aloittavalle sijoittajalle harhakuivan monimutkaisuudesta ja vaivallisuudesta, jolloin markkinoiden tehokkuus voi kärsiä.

Edelleen kasvavana trendinä on monenlaiset eri rajapinnat, joita kehittäjät voivat hyödyntää maksuttomasti ja helposti. Rajapintoja hyödyntämällä voidaan rakentaa työkalu, joka on sekä helposti muokattavissa että ilmainen. Tarjolla olevien rajapintojen, taulukkolaskennan ja talouden teorioiden avulla voidaan rakentaa työkalu, jonka avulla saadaan vähennettyä tarvittavaa aikamäärää osakkeiden analysoimiseksi. Osakesijoittajan näkökulmasta yksinkertainen ja ilmainen työkalu helpottaa sijoitusten seuraamista.

1.1 Opinnäytetyön tavoitteet ja rajaukset

Tämän opinnäytetyön tavoitteena on luoda osakeportfolion tekninen analyysityökalu, jolla pyritään luomaan käyttäjälle informatiivinen ja automaattisesti päivittyvä näkymä. Työkalu toteutetaan Excel VBA ohjelmoinnin avulla.

Tavoite jakautuu seuraaviin alatavoitteisiin:

1. Mitä tunnuslukuja portfolion analyysityökalun tulee sisältää?
2. Miten työkalulle syötetään tarvittavat tiedot analyysin tekemiseksi?
3. Miten työkalu muotoilee syötetyt tiedot?
4. Kuinka analyysityökalu tehdään?

Opinnäytetyö rajataan käsittelemään pääasiassa analyysityökalun teknistä puolta, johon liittyy muun muassa sen ohjelmointi. Teknisen puolen lisäksi opinnäytetyö käsittelee osakkeita ja niiden teoriaa, sekä näiden analysointiin käytettäviä tunnuslukuja.

1.2 Opinnäytetyön rakenne

Opinnäytetyö alkaa johdannolla, jossa kuvaillaan tämän työn tarkoitusta. Johdannon jälkeen tämä opinnäytetyö jakautuu kolmeen osaan, tietoperustaan sekä suunnittelua ja teknistä toteuttamista kuvaavaan toiminnalliseen osaan. Näiden jälkeen on pohdinta -osa,

jossa käydään läpi muun muassa opinnäytetyön tavoitteiden saavuttamista, sekä oma-kohtaista oppimista.

Tietoperusta alkaa osakkeisiin ja sijoittamiseen liittyvällä luvulla. Tässä luvussa kerrotaan yleisellä tasolla osakkeista ja sijoittamisesta, sekä näihin liittyvistä asioista, kuten portfoliosta ja tunnusluvuista. Tietoperusta jatkuu ohjelmoinnista kertovalla luvulla. Tässä luvussa on avattu ohjelmoinnin käsitteitä, jotta lukija tuntee opinnäytetyön toiminnallisen osan vaatimia teknisiä piirteitä.

Opinnäytetyön toiminnallinen osuus kertoo, miten analyysityökalu tehtiin. Tämä osuus sisältää muun muassa teorian, joka toimi koko työkalun rakentamisen perustana. Työkalun tavoitteet ja suunnittelu on myös sisällytetty toiminnalliseen osuuteen. Näiden jälkeen kuvaillaan itse toteuttamista, sekä sen testausta. Toiminnallisen osuuden eri vaiheet ovat perustettu kehitysmallin teorian mukaan.

2 Osakkeet ja sijoittaminen

Osakkeet ovat osia kohdeyrityksestä. Osakkeiden omistaminen tekee omistajasta myös itse yrityksen omistajan. Tällöin ihmiset, jotka omistavat saman yrityksen osakkeita omistavat myös palat samaa yritystä. Osakkeenomistajien lukumäärää, sekä osakkeiden lukumäärää pidetään yllä osakerekisterissä. Yrityksen osakkeilla kauppaa käytäessä, osakerekisteri muuttuu alati. (Hobson 2012, 3)

Osakkeenomistus ja yrityksen toiminta ovat kuitenkin eri asia. Omistajuus ei tarkoita yrityksessä työskentelyä tai sen päivittäisten asioiden hoitamista. Asian voi nähdä myös toisinpäin; yrityksessä toimiminen tai sen päivittäisten asioiden hoitaminen ei tee omistajaksi. On kuitenkin täysin normaalia ja sallittua toimia esimerkiksi yrityksen johdossa, sekä omistaa sen osakkeita. Osakkeenomistajilla on kuitenkin oikeus muun muassa osallistua vuosittaiseen yhtiökokoukseen, sekä äänestää hallituksesta ja tilintarkastajasta. (Hobson 2012, 4)

Osakkeenomistajan äänien määrä määräytyy omistettujen osakkeiden mukaan. Osakkeiden määrään on sidottu myös yrityksen voittojen jakaminen; mitä enemmän osakkeenomistajalla on osakkeita, sitä suuremman osuuden hän saa yrityksen voitoista. Osakekohtaista voittoa, jonka yritys jakaa osakkailleen kutsutaan osingoksi. (Hobson 2012, 4)

Yrityksen osakkeiden elinkaari alkaa, kun yritys perustetaan. Yrityksen perustajat tai sijoittajat sijoittavat varojaan, jotta yritys saa pääomaa. Usein tarvitaan pääomaa, ennen kuin yritys alkaa itse tuottamaan. Yritys voi kerätä varoja myymällä osakkeitaan. Osakkeiden määrä ei ole ennalta määrätty, jolloin yritys on oikeutettu itse päättämään osakkeiden määrästä, joita halutaan laskea liikkeelle. (Folger 2017)

Kun yrityksen osakkeilla käydään kauppaa, osakkeen arvo määräytyy pitkälti kysynnän ja tarjonnan lakien mukaan. Yhtiön nouseva kannattavuus lisää sen kiinnostusta sijoituskohteena, jolloin kiinnostuneita sijoittajia on enemmän. Tällöin yhtiön osake kerää ostajia, jolloin osakkeen hinta nousee. Huonot näkymät yhtiön tulevaisuudessa lisää osakkeiden myyjien määrää, jolloin tarjonta kasvaa suhteessa kysyntään. Myyjien määrän kasvaessa osakkeen hinta laskee. Vaikuttavia tekijöitä sijoittajien osto- ja myyntipäätöksiin on monia. Vaikutukset yhtiön osakkeen arvoon saattavat olla hyvinkin äkillisiä ja voimakkaita. Esimerkiksi öljyn hinnan muutokset saattavat vaikuttaa yritykseen, jonka liiketoiminta edellyttää kaupankäyntiä öljyllä. Vaikutukset yrityksen toimintaan vaikuttavat usein myös sijoittajien päätöksiin. (Folger 2017)

2.1 Osakesijoittaminen

Kun yritysten osakkeisiin sijoitetaan varoja, on sijoituksella jokin tarkoitus. Tavallisesti tarkoituksena on, että sijoituksesta saadaan jotakin hyötyä. Osakesijoittamisen hyödyt voidaan jakaa kolmeen osaan. Nämä ovat osingot, arvonnousu sekä osakkeenomistajan edut. (Hobson 2012, 23)

Monet tuottavat yritykset maksavat osakkeenomistajilleen osinkoja. Osinkojen maksamiseen vaadittavat varat yritys saa yritystoiminnan tuotoista. Osinkoina jaettavien varojen summa saadaan, kun lasketaan jäljelle jääneet varat verojen, sekä muiden kulujen jälkeen. Sijoittajan näkökulmasta osinkojen merkittävyys koostuu itse sijoituksen luonteesta ja tavoitteesta. Jos tavoitteena on saada tasaista kassavirtaa osakesijoituksesta, tulee sijoittajan etsiä yhtiöitä, jotka maksavat tasaista tai kasvavaa osinkoa osakkeenomistajilleen. (Hobson 2012, 23)

Osakesijoituksen arvonnousu toteutuu, kun itse osakkeen arvo kohoaa. Mikäli osakesijoituksen tarkoitus on toimia näin, sijoittajan on syytä etsiä yhtiöitä, jotka ovat aliarvostettuja muiden sijoittajien silmissä. Tulevaisuuden näkymät yhtiön kannalta tulevat olla positiiviset, jolloin osakkeen kurssinousulle on varaa. Osakkeen arvonnousun sekä kasvavat osingot kulkevat usein käsikädessä. Kasvavat tuotot johtavat osinkojen kasvuun, joka saa sijoittajat panostamaan yhtiöön. Sijoittajien laittaessa rahaa yhtiön osakkeisiin, myös sen osakkeen arvo nousee. Osakkeen arvonnousu ja osingot eivät kuitenkaan ole sidottuja toisiinsa. (Hobson 2012, 32)

Harvat yritykset tarjoavat osakkeenomistajille etuja, kuten alennuksia tuotteista tai palveluistaan. Etujen tarjoaminen on kuitenkin mahdollista, vaikka harvinaista. Esimerkkinä Ranskassa sijaitseva yritys Eurotunnel, joka operoi tunnelia Iso-Britannian ja Ranskan välillä. Eurotunnel tarjosi alkuperäisille osakkeenomistajilleen kolmea ilmaista automatkaa tunnelin läpi vuosittain. Etujen tarjoamisessa on kuitenkin huonot puolensa ja heikkoutensa. Etujen saamiseksi on usein rajoitteita esimerkiksi osakemäärissä sekä aikaväleissä, jolloin pienemmillä lyhytaikaisilla sijoituksilla näihin ei pääse käsiksi. Etuja verrattaessa osinkoihin huomataan myös, että edut ovat hyödyiltään rajoittuneempia, kun taas osingoista saatuja varoja osakkeenomistaja voi käyttää itse päättämäänsä kohteeseen. (Hobson 2012, 32-33)

Tuottojen ja hyötyjen lisäksi osakesijoittamiseen kuuluu olennaisesti riski. Osakesijoittamisen tuotot ovat olleet historiassa pitkällä aikavälillä paremmat kuin esimerkiksi talletusten. Osaketuottojen saamiseksi sijoittajien on kuitenkin siedettävä riskiä, joka koostuu mones-

ta eri tekijästä. Näihin tekijöihin kuuluu muun muassa osakkeiden arvonvaihtelut, markkinoiden epävakaus, yhtiön taloudellinen heikkous sekä konkurssi. (Finanssivalvonta 2015)

2.2 Osakeportfolio

Portfoliolla voidaan tarkoittaa yleisesti sijoitusten ryhmää tai yhdistelmää, jota hallitsee ja jonka omistaa tavallisimmin yksilö, yhdistys tai yhtiö. Osakeportfoliolla tarkoitetaan osakkeiden yhdistelmää, josta haltija pyrkii saamaan halutun hyödyn. Sijoittajan kootessa tai muokatessa osakeportfoliota tulee vertailla omakohtaista riskinsietokykyä portfolion sisältämiin osakkeisiin ja niiden riskisyyteen. (Investopedia.com, Portfolio)

Kun osakeportfoliosta halutaan tietynlainen sijoittajan tavoitteisiin sopivaksi, vaatii se usein selvitystä, ammattitaitoa ja tietämystä sijoittamisesta sekä osakkeista. Halutun portfolion kokoamiseksi sijoittajan tulee ensinnä määritellä omat tavoitteet, joihin portfolion osakkeilla on tarkoitus pyrkiä. Tavoitteisiin suhteutetaan samanaikaisesti sijoittajan toleranssi riskille, jonka merkitys on suuri. (World Finance 2013)

Portfolion hajauttaminen on yksi avaintekijöistä, kun kootaan mahdollisimman toimivaa yhdistelmää osakkeista. Jos hajautus on riittävää voi sijoittaja pysyä turvassa hyvinkin epävakailta markkinoilta. Toisaalta myös hyvin hajautettu portfolio voi auttaa sijoittajaa saamaan tasaista pääoman kasvua ja tuottoa sijoituksilleen. Osakeportfolion hajautus tapahtuu käytännössä keräämällä osakkeita eri toimialoilta. (World Finance 2013)

Osakeportfolion ylläpito vaatii sen analysointia. Mikäli portfoliota ei pidetä yllä, se ei välttämättä pääse enää sijoittajan sille asettamiin alkuperäisiin tavoitteisiin. Portfolion ylläpito saattaa tehdä suuren eron sijoitusten tuottavuuteen ja arvojen heilahteluihin. Säännöllisellä analysoinnilla pyritään erottamaan ne osakkeet, jotka parhaassa tapauksessa vasta tulevaisuudessa voisivat heikentää sijoitusten tuottoa tai kasvattaa riskiä yli halutun tason. Ennustaminen on mahdotonta, joten portfolion analysoinnilla pyritään mahdollisimman valistuneeseen päätökseen tehdä muutoksia tavoitteena turvata haluttu hyöty. (Zacks)

Harry Markowitz julkaisi vuonna 1952 Journal of Finance lehdessä kirjoituksen ”Portfolio Selection”, joka käsitteli riskin pienentämistä osakesijoittamisessa. Tästä kehittyi nykypäivän portfolioteoria, jonka tarkoituksena on käsitellä kuinka riskiä karttavat sijoittajat voivat valita portfolionsa sisällön tuoton optimoimiseksi. Portfolioteorian keskeinen ajatus on, että hajauttamalla sijoitukset erilaisiin kohteisiin, voidaan löytää yhdistelmä, jonka riski on mahdollisimman alhainen suhteessa tuottoihin. Sijoitusten valitseminen perustuu teorian

mukaan tilastollisiin tunnuslukuihin, kuten osakkeiden välisiin variansseihin, eli arvojen vaihteluiden suhteeseen toisiinsa nähden. (Investopedia.com, Modern Portfolio Theory)

Yleisellä tasolla riski tarkoittaa vaaraa menettää sijoitusten positiiviset tuotot. Riskiin luokituu kuitenkin myös negatiivisten tuottojen lisäksi tuottojen heikentyminen ja kasvaminen, jota selitetään teoksessa Yritysrahoitus (Niskanen & Niskanen 2013, 164) seuraavanlaisesti:

Vaikka riski usein mielletään puhtaasti tappionvaaraksi, sillä voidaan tarkoittaa molempiin suuntiin tapahtuvia poikkeamia tuoton odotusarvosta. Myös useimmissa riskin mittaustavoissa huomioidaan sekä negatiiviset että positiiviset poikkeamamahdollisuudet, jolloin esimerkiksi suuri myönteinen poikkeamamahdollisuus odotetusta tuotosta lisää riskiä.

Portfolion tuotto riippuu sen sisältämien sijoitusten tuotosta. Osakeportfolion tuotto riippuu sen sisältämien osakkeiden kahdesta eri tekijästä, osinkovirrasta ja pääoman kasvusta. Osinkovirta tuottaa tilanteesta riippuen osakkeenomistajalle tuloja vuosittain. Yritys maksaa osinkoja vain, jos osakkeenomistaja omistaa yrityksen osakkeita sillä hetkellä, ja yhtiö niitä päättää maksaa. Osinkojen osalta tuotot siis realisoituvat aina, kun yhtiöt niitä maksavat. Pääoman kasvusta saatu tuotto ei kuitenkaan voi realisoitua ilman, että osakkeenomistaja myy osakkeitaan. Osakkeen kasvanut hinta suhteessa ostohintaan luo pääoman kasvua. Mikäli myyntihinta on matalampi kuin hinta, jonka sijoittaja on niistä maksanut, tuotto on negatiivinen. (Niskanen & Niskanen 2013, 128-129)

2.3 Tunnusluvut

Tunnusluvut ovat matemaattisesti laskettuja arvoja sijoituksesta tai niiden yhdistelmästä, jolla pyritään saamaan parempi kuva tietyn hetken tilanteesta sijoituksissa. Tunnusluvut voivat perustua esimerkiksi viimeisen vuoden aikana kerättyihin historiallisiin arvoihin, mutta myös tämän päivän lukuihin. Osakeportfolion analysoimiseen käytetään erinäisiä tunnuslukuja. Näihin kuuluu muun muassa volatilitteetti, tuotto prosentti sekä P/E -luku. Matemaattisesti laskettujen lukujen avulla pyritään saamaan parempi kokonais käsitys yksittäisen osakkeen tai osakeyhdistelmän tilasta. Tunnuslukuja voidaan vertailla esimerkiksi indeksien tunnuslukuihin ja näin ollen vertailla suorituskykyä. Tunnusluvut perustuvat historiallisiin tietoihin, jonka seurauksena niiden perusteella ei voida tehdä varmoja ennustuksia tulevaisuuteen.

Mitä kuvataan?	Tunnusluku
Riski	Volatiliteetti
Tuotto	Kokonaistuottoprosentti
Tuotto	Arvonnousu (hinnan muutos)
Osakkeen hinnan suhde yrityksen tulokseen	P/E-luku

Taulukko 1. Tunnuslukujen kuvaamat aiheet.

2.3.1 Volatiliteetti

Volatiliteetti lasketaan osakkeen historiallisten tuottojen keskihajonnasta. Luku siis johdetaan tarkastelemalla tuottojen vaihtelevuutta tietyllä aikavälillä. Sijoittajat käyttävät historiallista volatiliteettia todennäköisimmin saatavien tuottojen arvioimiseen. Kun tuottojen keskihajonta on suurta, nähdään että ennustettavien tuottojen vaihteluväli on laaja. Suurempi vaihteluväli tarkoittaa suurempaa volatiliteettia. (Morningstar, Standard Deviation). Volatiliteetilla tarkoitetaan toisin sanoen osakkeen epävarmuutta ja riskiä. Korkea volatiliteetti käytännössä meinaa tuottojen korkeaa riskiä muuttua suuntaan tai toiseen jo lyhyellä aikavälillä. Volatiliteetti toimii myös toisinpäin. Matalana se meinaa, että kohteen tuotot eivät heilahtele suuresti lyhyellä aikavälillä, mutta muuttuvat tasaisesti. (Investopedia.com, Volatility)

Volatiliteetin luetettavuutta riskin mittarina on myös kyseenalaistettu. Sen pätevyys kärsii esimerkiksi tilanteissa, joissa tarkastellaan yritystä, jonka osakkeiden liikkuvuus on pientä. Mikäli osakkeet vaihtavat omistajaa vain vähän, suurena tekijänä sijoituksen riskillisyydessä on likviditeettiriski. Tällä tarkoitetaan, että joissakin tilanteissa näitä osakkeita voi olla vaikea muuttaa likvideiksi varoiksi eli myydä. Sijoittajien nähdessä edullisemmaksi myydä omistuksiaan vähän vaihdetuista osakkeista, riski arvonlaskulle on suuri. Volatiliteetin luotettavuus riskin mittarina on kärsinyt myös sen toiminnasta ennen pörssiromahduksia. Sen on huomattu näyttävän turhan positiivisia lukuja. (Finanssivalvonta 2.1.2012)

2.3.2 Kokonaistuottoprosentti

Kokonaistuottoprosentti kertoo sijoituksen tietyn aikavälin kaikki tuotot laskettuna yhteen. Yhdysvaltalainen rahoitusalan tutkimus ja hallinnointiyritys Morningstar laskee kokonaistuottoprosentin seuraavasti: Hinnan muutos otetulla aikavälillä lisättynä osingoista saatuihin tuloihin. Mahdolliset saadut tulot sijoitetaan takaisin. Saatu summa jaetaan lähtöhinnalla ja muutetaan prosenteiksi. (Morningstar.com, Total Return)

Kokonaistuottoprosentti tunnetaan myös nimellä sijoitusperiodin tuotto (HPR, Holding Period Return), joka voidaan laskea halutulle osakkeelle (Kaava 1.)

$$HPR = \frac{P_s - P_b + D}{P_b}$$

Kaava 1. Laskukaava sijoitusperiodin tuotolle, jossa P_s on osakkeen hinta periodin lopussa, P_b on osakkeen hinta periodin alussa ja D on periodin aikana maksetut osingot. (Knüpfer & Puttonen 2014)

Kokonaistuoton tärkeys huomataan, kun tarkastellaan eri tyyppisiä sijoituskohteita. Jotkin parhaista hyvin osinkoa maksavista osakkeista omaavat matalan kasvupotentiaalin, jolloin arvonnousu jää heikommaksi. Toisaalta sijoittamalla vain arvonnousun perusteella ei oteta huomioon yleistä hintojen nousua eikä muita tuottomahdollisuuksia. Esimerkiksi osakkeen 10% arvonnousu, sekä 2% maksetut osingot tekevät kokonaistuotoksi 12%. Näin nähdään sijoituksen todellinen kasvu, ja saadaan parempi käsitys kokonaiskuvasta. (Investopedia.com, Total Return)

2.3.3 P/E

Tunnusluku P/E saadaan, kun jaetaan osakkeen hinta yrityksen tekemällä osakekohtaisella tuloksella. Luku ilmaisee vuosimäärän, jossa yritys tekisi tulosta sen osakkeiden yhteenlasketun markkina-arvon verran (The Balance 2017). P/E-luku kuvastaa myös kuinka paljon sijoittajat ovat valmiita maksamaan ja kuinka arvokas osake on siihen nähden, että yritys tekee esimerkiksi yhden euron voittoa. Tilanteessa, jossa yrityksen osakkeen arvo on 43 euroa ja sen osakekohtainen tulos on 1,95 euroa, sen P/E-luku olisi $43/1,95 = 22,05$. Tämä tarkoittaa, että yritys tekee 1 euron voittoa 22,05 euroa kohden. (Investopedia.com, Price-Earnings Ratio)

P/E-lukua tarkasteltaessa on otettava huomioon, että tavallisimmin eri toimialoilla hyväksi P/E arvoksi nähtävä luku vaihtelee. Esimerkiksi teknologiayrityksen P/E luvun vertailu kangasteollisuuden alalla toimivan yrityksen P/E-lukuun ei ole kannattavaa, sillä ne saattavat vaihdella suuresti. Saman alan yritysten vertailu P/E luvun perusteella voi joissakin määrin olla hyödyllistä. Mikäli kahden saman alan ja saman kokoisen yrityksen osakkeiden hinnat ovat samalla tasolla, mutta toinen kykenee tekemään parempaa tulosta, huomataan että parempaa tulosta tekevä yritys on suhteessa halvempi. Tällöin myös P/E-luku pysyy matalampana. (The Balance, 2017)

3 Excel VBA

Microsoft Excel on yksi maailman eniten käytetyistä taulukkolaskentaohjelmista. Muiden taulukkolaskentaohjelmien ollessa saatavilla, Excel johtaa suosiolla. Sitä on pidetty jo pitkään oletuksena taulukkolaskentaohjelmasta. (Walkenbach 2015)

Excelin vahvuutena toimii sen monipuolisuus. Mahdollisuus toteuttaa numeerisia laskutoimituksia, sekä rakentaa erinäisiä sovelluksia tekevät siitä käyttäjälle hyödyllisen. Esimerkkinä eri tavoista käyttää sen ominaisuuksia voidaan nähdä muun muassa rahoitusanalyysit, listojen järjestäminen, taulukot, datan tuominen ulkopuolisesta lähteestä, sekä monimutkaisten tehtävien automatisointi ja havainnollistaminen. (Walkenbach 2015)

Excel sisältää ohjelmointikielen, jota kutsutaan Visual Basic For Applications –nimellä, tuttavallisemmin VBA. Se on osa Microsoftin kehittämää Office –työkalusarjaa, toimien lisämaksuttomana osana Office työkaluja. VBA: n avulla ohjelmistokehittäjät voivat rakentaa työkaluja ja applikaatioita omiin tai muiden tarpeisiin kustomoituna. (Walkenbach 2013, 11-13)

Vaikka VBA on täysiverinen ohjelmointikieli siinä missä muutkin ohjelmointikielet, sen toiminta tarvitsee ohjelman, johon se on kirjoitettu. Voidaan katsoa, että kaikki mitä VBA kielen sisältävällä ohjelmalla kyetään tekemään työkirjan puolella, on mahdollista myös automatisoida VBA koodilla. (Birnbaum 2005)

VBA on hyödyllinen, kun tehtävät asiat ovat hyvin toistuvia ja mahdollisia automatisoida. Sitä voidaan käyttää myös halutessa asettaa monipuolisia ja kustomoituja ehtoja työkirjalle sitä lukiessa, muokattaessa, tallentaessa tai tulostaessa. Myös applikaatioiden välinen tiedonsiirto ja esimerkiksi automaattinen asettelu saadaan VBA: n avulla toteutettu. (Microsoft Developer Network 2017)

3.1 Makro

Excelin makroksi voidaan laskea kaksi asiaa. Ensimmäinen niistä on Excelin itse automaattisesti kirjottamaa koodia. Tätä voidaan tallentaa, kun käyttöliittämässä tehdään eri asioita. Toisena on käyttäjän itse kirjoittama koodi saman asian tekemiseksi, jolla yleensä pyritään paremmin kontrolloituun koodiin ja tehokkuuteen. (Jelen & Syrstadt 2016)

Makro on siis kokonaisuus, joka suorittaa tietyn asian tai sarjan asioita. Makrot sisältyvät VBA ohjelmointiympäristössä moduuleihin. Yksi moduuli voi siis pitää sisällään monta makroa. Moduulin paikkana toimii projekti, joka pitää sisällään kaiken koodin ja toiminnallisuuden, jota on luotu. (Office VBA Macros You Can Use Today 2006, 7)

3.2 Muuttujat

Muuttujat ovat elementtejä, jotka pitävät sisällään tietoa. Ne ovat nimettyjä tallennuspaikkoja tietokoneen muistissa. Muuttuja voidaan määrittellä sisältämään muun muassa tekstiä, arvoja ja eri ominaisuuksien asetuksia. Muuttujien sisältämä tieto asetetaan aina erikseen. (Walkenbach 2013, 95)

Muuttujat on hyvä määrittellä VBA kielessä ennen niiden käyttöä, erillään käyttävästä lauseesta. Lause "Dim Lukumaara As Integer" luo Lukumaara -nimisen muuttujan, joka pitää sisällään kokonaisluvun. Muuttujan arvo asetetaan esimerkiksi lauseella "Lukumaara = 3". Muuttujan määrittely tietyille datatyypille nopeuttaa makron suorittamista ja käyttää vähemmän muistia, johtuen tarpeettomien tarkastusten poistumisesta. (Walkenbach 2013, 98)

3.3 Datatyypit

Aiemmin mainittu kokonaisluku on yksi VBA kielessä käytetyistä datatyypeistä. Erilaiset datatyypit vaativat tietokoneelta ja Exceliltä eri määrän työtä, jolloin ohjelman suoritusnopeus vaihtelee. Pienten VBA ohjelmien kohdalla eroja ei juurikaan huomaa, mutta siirryttäessä suurempiin kokonaisuuksiin ne tulevat tärkeiksi. VBA siis kykenee käsittelemään eri datatyypit ilman tiukkaa aiempaa määrittelyä. Määrittely on kuitenkin suotavaa ohjelmaa kirjoittaessa. (Walkenbach 2013, 96-97)

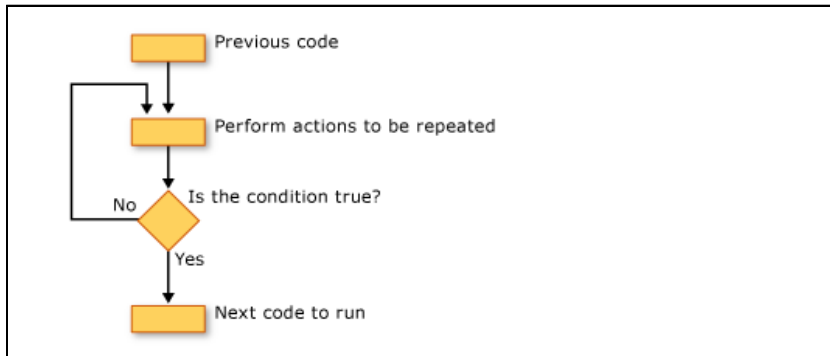
Datatyypit	Käytetyt tavut	Mahdolliset arvot
Tavu	1	0-255
Boolean	2	Tosi / Epätosi
Integer	2	-32 768 - 32 767
Long	4	-2 147 483 648 - 2 147 483 647
Single	4	-3.40 ja -1.40E-45 välin negatiiviset arvot 1.40E-45 ja 3.40 välin positiiviset arvo
Double	8	-1.79E308 ja -4.94E-324 negatiiviset arvot 4.94E-324 ja 1.79 positiiviset arvot
Currency	8	-922 337 203 685 477 ja 922 337 203 685 477 väliset arvot
Date	8	01.01.0100 - 31.12.9999
Object	4	Viittaus mihin tahansa objektiin
String	1, kirjainta kohden	Vaihtelee
Variant	Vaihtelee	Vaihtelee

Taulukko 2. Datatyypit (Walkenbach 2013, 97).

3.4 Ohjausrakenteet

Ohjausrakenne on yksi ohjelmoinnin peruskäsitteistä. Yksinkertaisimmillaan se tarkoittaa koodirivien yhdistelmää. Tarkemmin määriteltynä tämä yhdistelmä määrää kuinka, minne ja milloin koodissa edetään. Ohjausrakenne yhdistää funktioiden kutsuja, ohjeita ja lausekkeita. Yksinkertainen esimerkki ohjausrakenteesta on If...Then -ehtolause. Sen avulla määrätään ehto, jonka perusteella edetään ohjeiden mukaisesti. Etenemisohteet tulevat aina olla ennalta määrättyjä. Rakenne voi esimerkiksi ehdon täytyessä kutsua kuviteltua funktiota a. Mikäli ehto ei täyty, voidaan esimerkiksi antaa ohjelman käyttäjälle ilmoitus virheestä. (Valid-computing)

Ehtolauseiden lisäksi ohjausrakenteisiin kuuluu myös toistorakenteet. Yksi näistä on Do...Loop -rakenne. Sitä käytetään, kun halutaan suorittaa tiettyjä komentoja ennalta määräämätön määrä. Do...Loop rakenteesta on monia eri versioita, jotka toimivat hieman eri tavoin toisistaan. Jokainen niistä kuitenkin tarkastelee annettua ehtoa, jonka perusteella kiertoa joko jatketaan tai se päätetään. Ehdon tulee aina kääntyä muotoon tosi (true) tai epätosi (false). (Microsoft 2015a)



Kuva 1. Toistorakenteiden yleinen toimintaperiaate. Aiemmasta koodista siirrytään toistorakenteeseen, tarkistetaan ehdot ja edetään sen mukaisesti. Ehto voidaan myös tarkistaa toistettavia komentoja ennen. (Microsoft 2015a)

```

Dim Sum As Integer
Sum = 1

Do While Sum < 5
    Sum = Sum + 1
Loop
  
```

Kuva 2. Do...Loop rakenne. Toistorakenne jatkaa kiertoaan ja komentojen suoritusta, kunnes "While" ehto ei enää toteudu määrätyn mukaisesti.

```

Dim Sum As Integer
Sum = 1

Do Until Sum = 5
    Sum = Sum + 1
Loop
  
```

Kuva 3. Do...Loop rakenne. Erona kuvaan 2, tämän kuvan lause tarkistaa Until ehdon. Toistorakenne jatkaa kiertoaan ja komentojen suoritusta, kunnes ehto toteutuu.

For...Next -toistorakenne on myös yksi usein käytetyistä ohjausrakenteista. Sen toimintaperiaatteena on suorittaa toistuvasti sen sisään kirjoitettua koodia ennalta määrätty määrä kertoja. For...Next -toistorakenne käyttää toistojen kontrolloimiseen siihen liitettyä askelmuuttujaa. Askelmuuttuja mahdollistaa tärkeänä tekijänä toistomäärien laskemisen, jonka perusteella etenemistä hallitaan. Ohjelmoija määrittelee askelmuuttujan alku- ja loppuarvot. Näiden lisäksi ohjelmoija määrittelee, kuinka askelmuuttujaa käsitellään toistosta toiseen (Microsoft Developer Network).

```
Dim Count As Integer

For Count = 1 To 10
    MsgBox (Count)
Next
```

Kuva 4. Excel VBA ohjelmoitua koodia, jossa hyödynnetty For...Next -toistorakennetta. Kokonaislukumuuttujan "Count" halutaan kasvavan yhdestä kymmeneen, samalla määrätien toistorakenteen toistokerrat.

For Each...Next -toistorakenne on edellisten lisäksi hyvin käytetty ohjausrakenne. Tämä rakenne suorittaa sen sisään kirjoitettujen koodien toiminnot tietylle joukolle elementtejä. Ohjelmoija määrittelee askelmuuttujan, mutta sille ei aseteta alku- eikä loppuarvoa. For Each...Next -toistorakenteelle voidaan esimerkiksi asettaa läpikäytäväksi joukoksi lista kokonaislukuja, joista jokaiselle voidaan suorittaa haluttuja toimintoja (Microsoft 2015b).

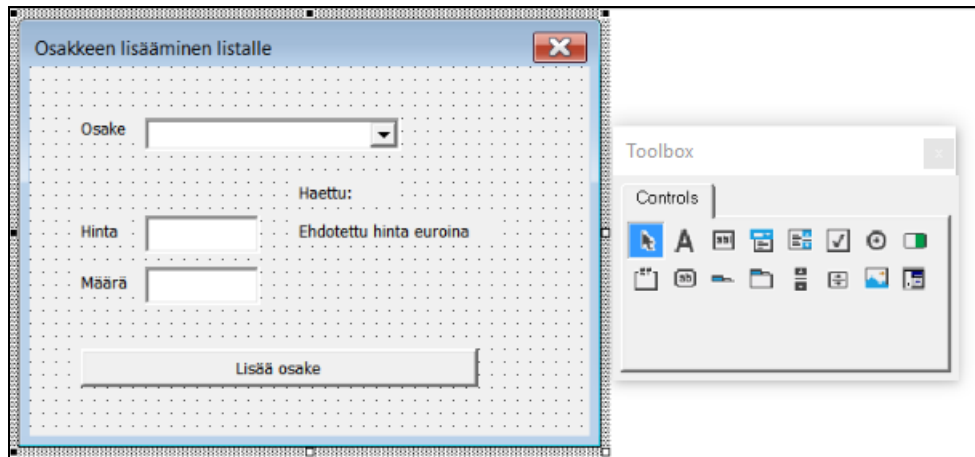
```
Dim Ws As Worksheet

For Each Ws In ActiveWorkbook.Worksheets
    MsgBox (Ws.Name)
Next
```

Kuva 5. Excel VBA ohjelmoitua koodia, joka käy läpi työkirjan välilehtiä ja tulostaa niiden nimet. Läpikäynnissä on hyödynnetty For Each...Next -toistorakennetta.

3.5 UserForm

UserForm on ohjelman tekijän luoma lomake. Sen voidaan käyttää olemassa olevan tiedon esittämiseen, uuden tiedon asettamiseen, sekä näytettävän datan muokkaamiseen. Tarkoituksena on näyttää käyttäjälle tietoa ja saada vastaus. Lomakkeen avulla saadaan luotua monipuolisia ikkunoita, joihin on mahdollista sisällyttää Windowsin normaaleja vuoropuhelupohjan elementtejä. Pohja lomakkeeseen luodaan Visual Basic Editorissa. Tyhjälle pohjalle lisätään työkalulaatikosta eri elementtejä, joihin kuuluu esimerkiksi nimikkeet, tekstikentät, pudotusvalikot, sekä komentopainikkeet. Lomakkeen tekijä kykenee vapaasti päättämään eri elementtien sijainnit, sekä niiden määrän. Jokaiselle niistä voidaan myös asettaa monia eri tapahtumia. (Green 2007, 273)



Kuva 6. Lomakkeen luominen ja muokkaaminen.

3.6 Ohjelmointirajapinta

Ohjelmointirajapinnalla (engl. Application Programming Interface tai API) tarkoitetaan käyttöliittymää, jonka avulla voidaan luoda yhteys kahden toisistaan erillisen ohjelmiston välille. Ohjelmistojen ei tule välttämättä olla saman tyyppisiä niitä linkittäessä. Ohjelmointirajapinta voi toimia esimerkiksi datan välittäjänä kahden ohjelmiston välillä. Ohjelmointirajapintojen suosio on jatkunut jo pitkään. (Uzayr 2016)

Ohjelmoijilla ja sovelluskehittäjillä rajapinnat ovat usein päivittäisessä käytössä. Tunnettuina ohjelmointirajapintojen hyödyntäjän on esimerkiksi Facebook. Sen sisältämät pelit ovat suurimmilta osin kehitetty käyttäen Facebookin luomia ohjelmointirajapintoja. Tällöin peli saadaan liitettyä saumattomasti toiseen sovellukseen. Ohjelmointirajapinnat siis avaavat ohjelmoijille tiet hyödyntämään toisten ohjelmistojen, palveluiden ja alustojen ominaisuuksia ja sisältöä omaan käyttöön. Rajapintojen avulla voidaan hallita tiedon tarjoamista, jolloin toiminta on turvallista ja halutuissa määrin rajoitettua. (Uzayr 2016)

3.6.1 REST rajapinta ja tiedon hakeminen

REST, joka tulee sanoista "Representational State Transfer", on avoin ohjelmointirajapinta, joka toimii tiedonlähteenä. REST toimii verkkorajapintana serverin ja kehittäjän välillä, jolloin kehittäjän ei tarvitse välittää serverin toimintaominaisuuksista. Sen toimintaperiaatteena toimii käyttäjän näkökulmasta asioiden tai tiedon pyytäminen. REST ei toteuta tai toimita pyytäjälle toiminnallisuutta, eli sille ei ole mahdollista lähettää käskyjä. REST rajapinnan tarkoituksena on helpottaa tiedon kulkeutumista kehittäjille, edesauttaen ohjelmistojen rakentamista. Rajapinta tekee tiedon hyödyntämisestä helpompaa juuri kehittäjien omien tarpeiden vastaamiseksi. (Hunter 2016, Chapter 1)

3.7 Kirjasto – VBA-JSON Converter

VBA-JSON Converter on avoimen lähdekoodin ohjelmointirajapinta, joka on tehty JSON (JavaScript Object Notation) -tiedostomuodon datan siirtämiseen Excel VBA ohjelmointiympäristöön. Se on maksuttomassa jakelussa GitHub (<https://github.com>) verkkosivustolla. (Hall 2017)

JSON on kevyt tekstimuotoisen datan vaihtomuoto. Sen avulla data saadaan muotoiltua helposti luettavaksi ihmisille, sekä helposti jäsennettäväksi ja tuotetuksi tietokoneille. JSON on tekstimuotona ohjelmointikielistä riippumaton, mutta sisältää ominaisuuksia joiden avulla yleisimpien ohjelmointikielten, kuten C:n, Pythonin sekä Javan osaajat kykenevät sitä helposti ymmärtämään ja hyödyntämään. (json.org)

JSON koostuu kahdesta eri rakenteesta, joista ensimmäinen on nimi-arvo parit. Monissa ohjelmointikielissä tämä tarkoittaa esimerkiksi objekteja tai hajautustauluja. Toinen rakenne on arvoja sisältävät järjestetyt listat. Nämä rakenteet nähdään universaaleina tietorakenteina, jonka seurauksena käytännössä kaikki modernit ohjelmointikieliset sisältävät niiden tuen muodossa tai toisessa. Koska JSON on tarkoitettu datan vaihtomuodoksi, on sen luomisen peruseriaatteena myös pidetty, että se perustuu näille rakenteille. (json.org)

```
▼ 53:
  id: "44729"
  isin: "FI0009013403"
  name: "KONE OYJ"
  price: 43.88
  currency: "EUR"
  date: "2017-12-08T12:47:59Z"
  performance: -0.44
  performancePercent: -0.9928
  industry: "2000"
  ▼ info:
    lastPriceEuro: 43.88
    dividend: 1.55
    marketCap: 19797238780.1
    dividendState: 2
    eps: 2
    pe: 21.94
    ep: 4.557885141294439
    dividendIncome: 3.5323609845032
    dividendExDate: "2017-02-28T23:00:00Z"
```

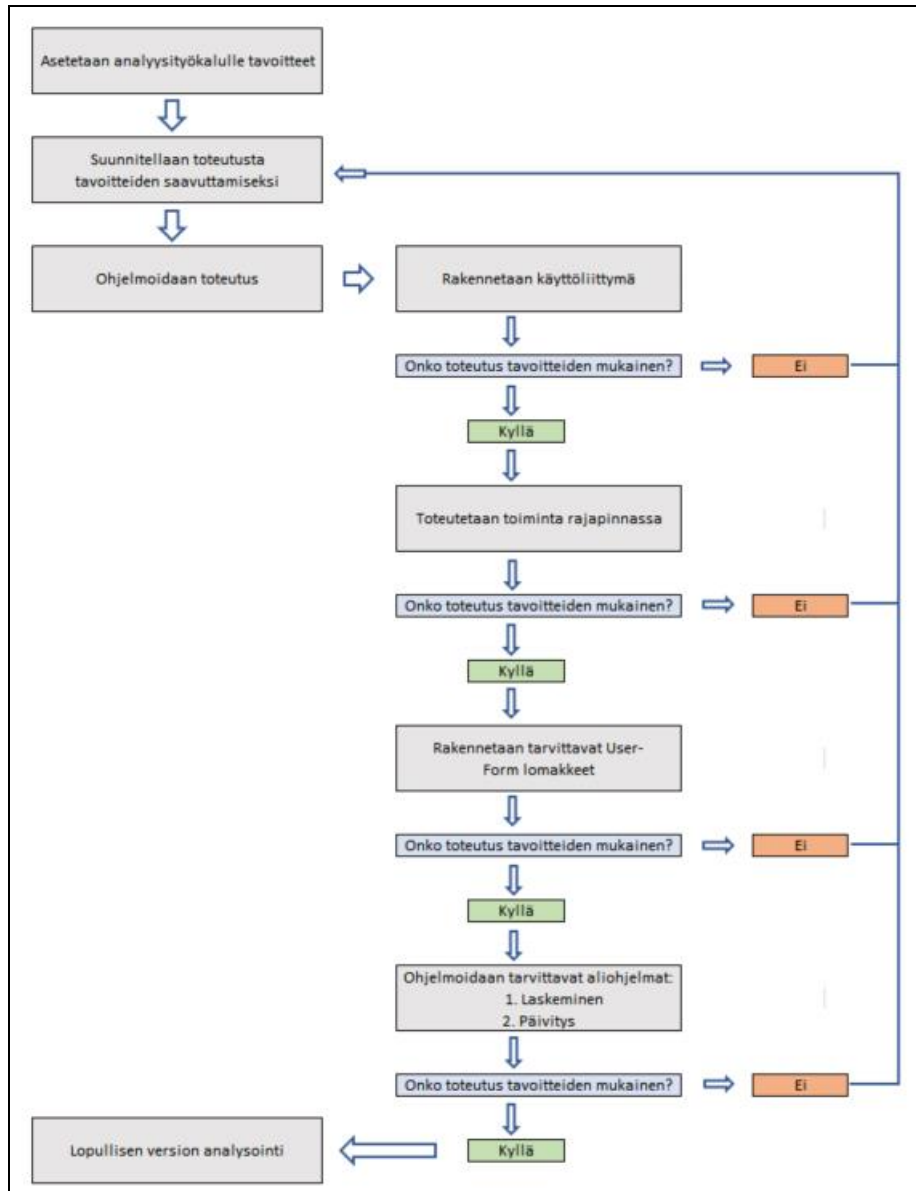
Kuva 7. JSON muotoista dataa järjestettynä. Sinisellä ilmaistuna nimi, jonka perässä siihen liitetty arvo.

```
{ "id": "44729", "isin": "FI0009013403", "name": "KONE  
OYJ", "price": 43.88, "currency": "EUR", "date": "2017-12-  
08T12:47:59Z", "performance": -0.44, "performancePercent": -0.9928, "industry": "2000", "info":  
{ "lastPriceEuro": 43.88, "dividend": 1.55, "marketCap": 19797238780.1, "dividendState": 2, "eps": 2  
, "pe": 21.94, "ep": 4.557885141294439380127620783956244, "dividendIncome": 3.5323609845032, "div  
idendExDate": "2017-02-28T23:00:00Z" } },
```

Kuva 8. Sama data kuin kuvassa 7 kuvattuna raakamuodossa.

4 Portfolioanalyysityökalun tekeminen

Tässä luvussa kuvaillaan opinnäytetyön aiheena olevan analyysityökalun suunnittelemisen, rakentamisen ja testaamisen eri vaihteita. Vaiheet perustuvat ohjelmiston spiraaliseen kehitysmalliin (Kuva 9).



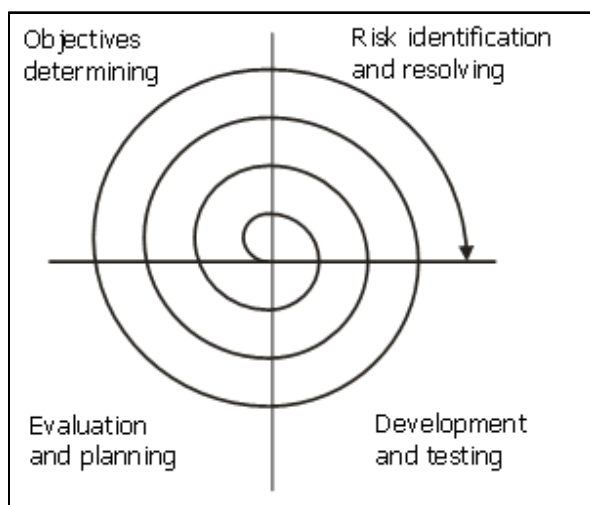
Kuva 9. Vaiheittainen kuvaus työkalun rakentamisesta

4.1 Spiraali kehitysmalli

Amerikkalainen Barry Boehm kehitti vuonna 1988 perinteisen vesiputous - ohjelmistokehitysmallin tilalle spiraalisen kehitysmallin. Sen tarkoituksena oli olla vesiputousmalliin verrattuna kokonaisvaltaisempi sekä soveltuvampi ohjelmistokehityksen tarpeisiin. Spiraalisen kehitysmallin pääajatuksena on luoda riskipohjainen lähestymistapa

ohjelmistokehitykseen. Koska kehitykseen sisältyy monia muuttujia ja riskejä, spiraalisen mallin avulla näihin pystytään vastaamaan. Yleisiä riskialueita ovat muun muassa kustannusten ylitykset, muuttuneet vaatimukset, projektihenkilöiden vaihtuminen, tarvittavien laitteiden viivästyminen, sekä teknologiset läpimurrot, jotka vaikuttavat projektiin.

Spiraalisessa mallissa ohjelmiston prototyyppiä testataan pitkin sen kehitystä. Testaamisen tarkoituksena on tehdä ohjelmistosta asteittain parempi. Jokaisessa testaamisen vaiheessa prototyypin toiminnasta tehdään riskianalyysi. Analyysin perusteella voidaan päättää, halutaanko ohjelmiston kehittyvän viimeisimmän prototyypin mukaiseen suuntaan, vai hylätäänkö muutokset ja tehdään uusi prototyyppi. (Virginia Tech, Department of Computer Science).



Kuva 10. Spiraalisen kehitysmallin vaiheet jaettuna neljään osaan. (Expertiza 2012)

4.2 Työkalun tavoitteet

Työkalun keskeisenä tavoitteena on tarjota käyttäjälle mahdollisuus tarkastaa portfolion tila, sekä tehdä siihen muutoksia samassa käyttöliittymän näkymässä. Kun käyttöliittymänä toimii vain yksi näkymä, työkalun toimintaperiaate pysyy yksinkertaisena ja helposti sisäistettävänä käyttäjälle.

Koska työkalu tarjoaa vain yhden näkymän portfolion analysointiin, tulee samaan näkymään sisällyttää myös halutut tunnusluvut jokaisesta osakkeesta. Tunnuslukujen avulla käyttäjä pystyy tekemään analysoivan katsauksen portfolioonsa. Osakkeiden ja tunnuslukujen sijoittaminen samalle näkymälle luo kuitenkin haasteita. Yhtenä suurena haasteena on, että tiedot pysyvät järjestyksessä. Kun samaan näkymään asetetaan paljon tietoa, on tärkeää, etteivät ne mene sekaisen keskenään. Tavoitteena on luoda näkymä, jonka vaih-

televuus on mahdollisimman looginen. Tämä tarkoittaa, että tunnuslukuja ei tule olla perusasetelmassa liikaa ja niiden tulee olla sijoiteltu loogisesti.

Tavoitteena on luoda työkalu, jonka avulla käyttäjä saa nopeammin kokonaiskuvan osakeportfolionsa tilasta verrattuna eri internetsivujen läpikäyntiin. Kokonaiskuvaan liitetään työkalussa muun muassa salkun arvo, yksittäisten osakkeiden arvo, eri arvojen muutokset sekä p/e-luku, volatilitteetti ja historiallinen tuotto. Näiden tietojen avulla työkalun käyttäjä saa muodostettua kuvan osakkeiden tilasta ja näin ollen myös koko osakeportfolion tilasta.

Työkalun tavoitteena on myös pitää kirjaa käyttäjän toimista. Sen tulee tallentaa kaikki lisäykset ja poistot myöhempää tarkastelua varten. Osakkeiden lisääminen työkaluun ja sieltä poistaminen kuvastaa niiden ostoja ja myyntejä. Tallennettujen tietojen perusteella on tavoitteena luoda lokikirja, jota käyttäjä pystyy hyödyntämään. Tiedot tulee olla jaettuna kahteen eri listaan, lisäyksiin ja poistoihin.

Työkalun yksi tavoitteista on pitää kaikki käyttäjän osakeportfolion osakkeiden tiedot ajan tasalla. Päivitetyt tiedot mahdollistavat paremman analyysin, jonka myötä käyttäjä hyötyy työkalusta enemmän. Tietojen tulee päivittyä automaattisesti ilman käyttäjän toimia. Automaattinen päivittäminen rajataan kuitenkin ohjelman käynnistämiseen. Kun käyttäjä avaa ohjelman, tulee tietojen päivittyä. Ajan tasaisilla tiedoilla ja automaattisella päivittämisellä on tavoitteena saavuttaa käyttäjälle nopeampi käyttökokemus, sekä korkeampi luotettavuuden taso.

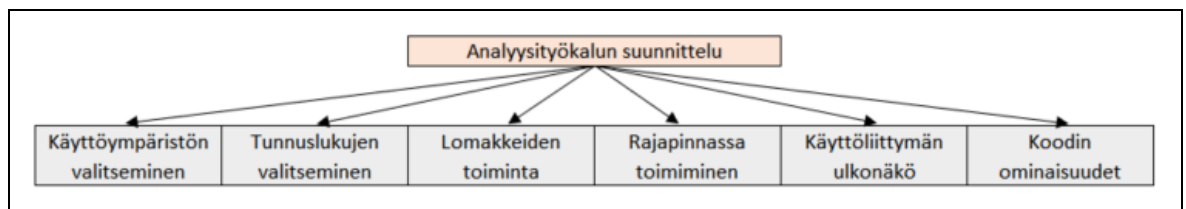
Tavoitteena on luoda käyttäjälle ilmainen ohjelma. Koska reaaliaikaiset tiedot ovat maksullisia, ohjelman tulee hyödyntää viivästettyjä tietoja, joita on saatavilla maksuttomasti. Työkalun tavoitteena ei kuitenkaan ole tarjota alustaa tai tietolähdettä nopeatempoiseen osakekauppaan, jonka vuoksi viivästettyjen tietojen hyödyntäminen ei tee työkalun käyttämisestä vähemmän hyödyllistä käyttäjälle. Hyödynnettäessä maksutonta dataa tulee varmistaa, että lähde on luotettava. Lähteen tulee myös mahdollisimman vakaa, jolloin vältytään tietokatkoksilta.

Työkalun toiminnan tulee perustua pitkälti automaattisesti tapahtuviin komentoihin. Automatisoidulla toiminnalla tavoitellaan mahdollisimman pientä riskiä inhimillisiin virheisiin. Virheitä ei tule, kun työkalu käsittelee automaattisesti kaiken tarvittavan datan. Tietovirheiden riski on kuitenkin olemassa johtuen mahdollisesta virheestä ohjelmoinnissa tai hyödynnettävässä datassa. Koska tavoite on minimoida virheet, hyödynnetään ohjelmaa rakennettaessa muun muassa spiraalista ohjelmiston kehitysmallia.

Työkalu pyrkii kokonaisuudessaan auttamaan käyttäjää seuraamaan osakeportfolionsa toimintaa ja tilaa, sekä tarjoamaan katsauksen valittuihin tunnuslukuihin ja arvoihin. Käyttäjän saama hyöty työkalusta ilmenee muun muassa säästettynä aikana ja vaivana. Kun osakkeiden ja portfolion analysoinnista ja tarkastelusta karsitaan pois mahdollisimman paljon manuaalista työtä, saavutetaan käyttäjälle välitön hyöty.

4.3 Työkalun suunnittelu

Analyysityökalun suunnitteluvaiheessa pyritään löytämään keinot, joilla tavoitteiden mukainen toteutus voidaan saada tehtyä. Spiraalisen kehitysmallin mukaisesti suunnitteluvaiheeseen palataan, mikäli testaus- ja analysointivaiheissa ilmenee asioita, joiden vuoksi suunnitelmaa on muutettava. Analyysityökalun suunnittelu on jaettu kuuteen eri pääosaan (Kuva 11).



Kuva 11. Suunnittelun eri vaiheet.

Työkalun tavoitteiden saavuttamiseksi katsotaan sopivimmaksi vaihtoehdoksi suorittaa toteutus Microsoft Excel VBA:lla, jonka käytöstä minulla on aiempaa kokemusta. Henkilökohtainen osaaminen nopeuttaa työkalun ohjelmointia ja vähentää virheiden määrää. Excel taulukkolaskennan sekä VBA:n opinnot ovat edesauttaneet hahmottamaan, kuinka kyseinen ohjelmointiympäristö toimii ja mitä sillä voi tehdä. Työkalun ohjelmoinnin ja kehittämisen halutaan myös vastaavan koulutusohjelman sisältöä, jolloin VBA:n hyödyntäminen olisi konkreettinen osoitus opituista asioista, sekä niiden soveltamisesta eri käyttötarkoituksiin. Opinnoista on jäänyt talteen myös kurssitehtävinä koodattuja ohjelmia, joita voidaan hyödyntää.

Maailman käytetyimpänä taulukkolaskentaohjelmana Excel on erinomainen alusta toteuttaa osakeanalyysityökalu. Aiempi kokemus Excelin käytöstä auttaa käyttäjää ymmärtämään työkalun toimintaperiaatteen. VBA:n hyödyntäminen analyysityökalun rakentamiseen ei tarkoita, että käyttäjän tulisi osata kyseistä ohjelmointikieltä. Myös tämän seikan vuoksi Excel luo hyvän ympäristön työkalun kehittämiseen ja käyttämiseen.

Portfolioanalyysityökalun jatkokehittäminen halutaan tehdä mahdolliseksi. Mikäli käyttöliittymä tai muu ohjelman toiminta ei vastaa käyttäjän tarpeita, on hyvä tarjota mahdollisuus omatoimiseen työkalun modifiointiin. Työkalu perustuu lähes kokonaan VBA koodiin, jonka vuoksi tehokas kehittäminen edellyttäisi sen osaamista. VBA:n taitaminen ei kuitenkaan ole välttämätöntä, mikäli muutoksia halutaan suorittaa työkirjan puolella soluissa. Excelin valmiit funktiot luovat varteenotettavia mahdollisuuksia jatkokehittämiseen ilman VBA koodia, jonka vuoksi sen valinta ohjelman kehitys- ja toimintaympäristöksi on hyödyllistä. Analyysityökalun VBA koodin, sekä solujen suojaaminen tulee olla lukitsematonta edellä mainittuja seikkoja ajatellen. Muokattavuus lisää työkalun hyödyllisyyttä mahdollisesti muuttuvissa tarpeissa.

Käyttöliittymä, lomakkeet ja tietojen listaaminen ovat tavoitteiden mukaisen portfolioanalyysityökalun toiminnan tärkeimmät ominaisuudet. Excel mahdollistaa halutun käyttöliittymän rakentamisen. Solujen, rajojen, värien ja tekstin avulla käyttöliittymän ulkonäkö saadaan halutunlaiseksi. Koska Excelin työkirjan valmiit työkalut mahdollistavat näiden ulkonäkötekijöiden vapaamuotoisen muokkaamisen, on niiden hyödyntäminen järkevää ohjelman toiminnan ja tehokkuuden kannalta.

Jotta analyysityökalu pysyisi mahdollisimman yksinkertaisena, siihen ei luoda erillistä tietokantaa, vaan tiedot tallennetaan Excel työkirjan soluihin. Tietojen noutaminen ja asettaminen soluihin on oleellista työkalun toiminnan kannalta. Sen tulee osata hakea päivitettyä tietoa itse. Avoimessa jakelussa oleva VBA-JSON kirjasto mahdollistaa json muotoisen datan kääntämisen ja hyödyntämisen maksuttomasti Excel ympäristössä käytettäväksi.

Portfolioanalyysityökalun keskeinen tekijä on sen käyttäjälleen tarjoamat tunnusluvut. Niiden perusteella käyttäjä kykenee analysoimaan osakkeita ja portfolioaan. Työkalussa näytettävät tunnusluvut on valittava huolellisesti, jotta saadaan poimittua hyödyllisimmät. Tunnuslukujen valintaa rajoittaa kuitenkin se, että niiden tulee olla tarjolla rajapinnassa, josta tietoja haetaan. Valmiiksi laskettujen tunnuslukujen etuna saavutetaan ohjelman nopeampi toiminta, sekä parempi toimintavarmuus. Valmiiksi lasketut tunnusluvut katsotaan sopivaksi portfolioanalyysityökalun käyttötarkoitukseen. Tunnuslukujen lisäksi tulee harkita, mitä muuta työkalun käyttöliittymän halutaan sisältävän. Näitä ovat muun muassa osakkeiden hintojen muutosprosentit ja summat.

Yhdeksi työkalun tarjoamaksi osa-alueeksi tunnuslukujen osalta valikoitui riski. Koska riski on olennainen osa portfolioteoriaa, sen kuvaaminen käyttäjälle on tärkeää. Työkalussa osakkeiden riskiä mitataan volatiliteetin avulla, joka on laskettu 250 päivälle.

Osakkeiden ja portfolion analysoimiseksi on olennaisena osana tarkasteltava tuottoa. Sen tarkastelemiseksi valikoitui kokonaistuotto prosentti. Tavoitteena on tarjota käyttäjälle informaatiota, josta ilmenee portfolion ja osakkeiden tuottama taloudellinen hyöty. Historiallinen kokonaistuotto prosentti on laskettu yhden vuoden aikavälillä nykypäivästä.

Portfolion sisältäessä enemmän kuin yhtä eri osaketta, tulee osakkeiden painoarvot ottaa huomioon. Tämän perusteella käyttäjän on mahdollista saada käsitys, kuinka portfolion kokonaisarvo jakautuu eri osakkeille. Painoarvon laskukaavana toimii tiettyjen osakkeiden määrä kerrottuna niiden viimeisimmällä arvolla. Tulo jaetaan koko salkun arvolla, jolloin saadaan yksilöity luku. Painoarvon perusteella voidaan esimerkiksi arvioida, onko portfolion sisältö painottunut liikaa tai liian vähän tietyn yhtiön osakkeeseen tai toimialaan. Painoarvojen perusteella lasketaan myös portfolion kokonaistuotto prosentti. Osakkeiden painoarvoilla korjataan tuotto prosentit, jolloin nähdään portfolion todellinen historiallinen tuotto vuoden ajalta. Painoarvoilla korjatulla tuotto prosentilla pyritään antamaan realistinen kuva salkun tilasta.

Osakkeiden arvonvaihtelu on yksi tekijöistä, joka vaikuttaa kokonaistuotto prosenttiin. Arvonvaihtelun tarkastelu erikseen on kuitenkin myös hyödyllistä. Tämän vuoksi työkalun käyttöliittymän tulee sisältää erikseen laskettu hinnan muutos ostohetkestä nykypäivään. Käyttäjä pystyy tämän perusteella näkemään, kuinka paljon osakkeen arvo on muuttunut prosentteina siitä päivästä lähtien, kun se on portfolioon lisätty. Arvon muutos lasketaan kaavalla: $(\text{nykyhinta} - \text{ostohinta}) / \text{ostohinta} * 100\%$. Käyttäjän saama hyöty arvonvaihtelusta riippuu sijoitusstrategiasta. Mikäli arvonnousulle annetaan enemmän painoarvoa kuin osingoille, se on hyödyllinen.

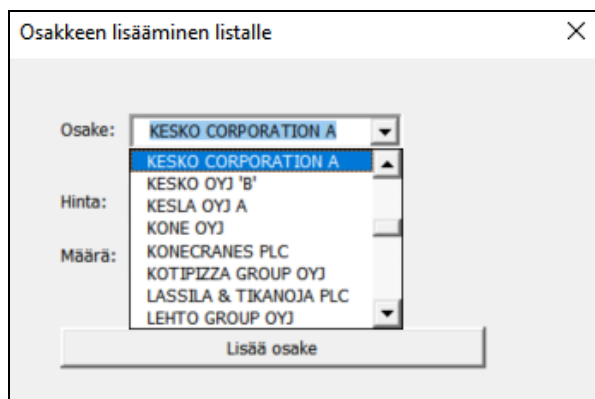
Jotta käyttäjä saa realistisen kuvan osakeportfolionsa tilanteesta, tulee työkalun käyttöliittymän sisältää koko salkun yhteenlaskettu arvo. Sen avulla käyttäjälle tarjotaan tieto, kuinka paljon portfolioon lisättyjen osakkeiden arvo on summana. Yhteenlaskettu arvo saadaan, kun summataan osakkeiden nykyarvolla kerrotut määrät. Kuten muutkin työkalun tarjoamat luvut, myös yhteenlaskettu arvo lasketaan automaattisesti, jolloin vältetään mahdollisilta laskuvirheiltä.

Osakkeiden lisäämisessä portfolioon noudatetaan työn tavoitetta automatisoida mahdollisimman moni asia työkalun käytön nopeuttamiseksi, sekä virheiden vähentämiseksi. Tämän perusteella osakkeiden lisääminen työkaluun tulee suorittaa valmiiksi määritellyn lomakkeen avulla. Ennalta määritetyt rajoitteet tiedon lisäämiselle edesauttavat tietojen järjestämistä, hallintaa ja ohjelman yleistä toimintaa. Lomakkeen avulla pystytään myös

varmistamaan, että ohjelma etenee tietojen lisäämisen suhteen ennalta määrätyllä tavalla. Lomake kootaan Excelin sisäänrakennetulla UserForm työkalulla.

Lomake osakkeiden lisäämiseksi tulee sisältää tiedon lisättävästä yhtiöstä, osakkeiden määrästä sekä osakekohtaisesti hinnasta. Käyttäjän avuksi lomakkeelle haetaan myös viimeisin hinta. Hinnan määrittäminen on kuitenkin käyttäjän vastuulla, sillä ei voida varmaksiksi määrittellä onko käyttäjä ostanut yhtiön osakkeita juuri viimeisimmällä hinnalla. Hae-
tun hinnan tarkoitus on antaa osviittaa hinnan kirjoittamiseksi lomakkeelle, välttämällä mahdollisia sekaannuksia. Koska hintaa ei voida määrittää automaattisesti tietojen saavuttamattomuuden vuoksi, on käyttäjän syötettävä luku manuaalisesti.

Käyttäjän lisätessä osaketta analyysityökalun listalle, haluttu osake tulee olla valittavissa automaattisesti haetusta pudotusvalikosta. Listan tulee sisältää kaikki Helsingin pörssin päälistan yhtiöt. Valittavissa olevien yritysten listan tulee olla päivitettyä tietoa, mikäli pörssin listalla tapahtuu muutoksia. Esimerkiksi tilanteessa, jossa uusi yritys listautuu pörssiin, tulee työkalun tarjota mahdollisuus listatun yrityksen lisäämiseen. Koska käyttäjälle ei sallita mahdollisuutta lisätä yhtiön tietoja käsin, tietojen pitää pysyä päivitettyinä ohjelman ajantasaisuuden takaamiseksi. Tiedot päivitetään, kun käyttäjä käynnistää ohjelman.



Kuva 12. Lisäyslomakkeen pudotusvalikko, joka sisältää yhtiöiden nimet.

Lomakkeen avulla tapahtuva osakkeiden lisääminen listalle mahdollistaa myös ohjelman tavoitteena olevan automaattisen tietojen hakemisen. Yksittäisen yhtiön tietoja ei tulla hakemaan yhtiölle yksilöidystä rajapinnasta sen nimen perusteella, vaan sille erikseen osoitetun tunnusluvun, eli id:n perusteella. Tämän avulla voidaan suorittaa rajapintaan pyyntö, ja saada tarvittavat tiedot ohjelman käytettäväksi. Edellä mainitulla suoritusmenetelmällä analyysityökalun toiminta on käyttäjän kannalta yksinkertaisempaa, verrattuna tietojen syöttämiseen käsin. Koska ohjelman tavoitteena on suorittaa automaattisesti mahdolli-

simman moni toiminto, myös tietojen lisääminen tulee olla automaattista. Ennalta määrätty ohjelman kulku nopeuttaa ohjelman käyttöä käyttäjän näkökulmasta.

Pörssiyhtiöiden nimet haetaan automaattisesti. Tällöin yhtiöiden nimissä ei ole mahdollisuutta kirjoitusvirheille. Väärin kirjoitettu yhtiön nimi voisi tuottaa ohjelman kulkuun vaikuttavan virheen. Virheiden minimoimiseksi on perusteltua toteuttaa työkalun koodin eteneminen ja toiminta niin, etteivät kirjoitusvirheet vaikuta siihen.

UserForm lomaketta luotaessa, sen toiminta ohjelmoidaan erilliseen pohjaan. Koodi on niin sanotusti lomakkeen takana, jonka vuoksi sitä ohjelmoitaessa on helppo havaita mitkä koodin osat kuuluvat mihinkin lomakkeeseen. Koodin järjestäminen lomakkeeseen liitettynä on myös hyödyllistä jatkokehitystä varten. Mikäli jatkokehittäjä tuntee VBA kielen, kykenee hän löytämään lomakkeeseen ohjelmoidut toiminnot. Lomakkeessa tulee olla nappi, jolla käyttäjä hyväksyy asettamansa tiedot. Napin tehtävänä on laukaista lomakkeeseen ohjelmoidut toiminnot.

Tavoitteiden mukaisen analyysityökalun ohjelmoiminen vaatii tehokkaasti hyödynnettävän datalähteen. Itse data ja sen hyödyntäminen on yksi ohjelman toiminnan avaintekijöistä. Koska ohjelman toiminta perustuu datan analysointiin, hyödynnettävän lähteen tulee olla käytettävissä kellonajasta riippumatta. Tavoitteiden mukaisiin tarpeisiin tulee hyödyntää REST-rajapintaa. Sen avulla ohjelma voi hakea dataa rajapinnasta milloin tahansa. REST rajapinnan hyödyntäminen on tehokasta sen nopeuden ja Exceliin saatavilla olevan kirjaston takia.

Dataa tuotaessa Exceliin, pyritään tavoitteiden mukaisesti minimoimaan siihen käytetty aika. Vaadittua aikaa saadaan pienennettyä muun muassa karsimalla turhien tietojen hakemista. Analyysityökalun tulee tuoda rajapinnasta dataa vain minimimäärä. Osakkeille yksilöityjä tietoja on saatavilla erillisistä REST-rajapinnoista. Yksilöityjen tietojen hakeminen suoritetaan, kun käyttäjä lisää osakkeen työkaluun.

Datan noutaminen verkosta halutaan toteuttaa luotettavalta taholta. Väärä tieto voisi aiheuttaa analyysityökalun käyttäjälle vääristyneen kuvan portfolion tilanteesta. Verkossa julkaistavan Taloussanomien REST-rajapinnan käyttäminen lähteenä katsotaan sopivaksi sen maksuttomuuden ja luotettavuuden perusteella. Kyseisen rajapinnan data on 15 minuuttia viivästettyä, mutta analyysityökalun käyttötarkoituksen nojalla tällä ei ole merkitystä. Tietojen virheettömyys nähdään kuitenkin tärkeänä tekijänä, jonka vuoksi työkalun tulee varmistaa, että päivitys tapahtuu oikein.

Rajapinnasta noudetut tiedot päivitetään kahden eri toiminnon seurauksena. Ensimmäinen näistä on työkalun käynnistäminen. Tällöin työkalu päivittää tiedot automaattisesti ilman erillistä toimenpidettä tai ilmoitusta. Toinen päivittävä toiminto on käyttäjän erikseen antama käsky. Työkalun käyttöliittymän tulee sisältää muiden painikkeiden ohessa päivityspainike. Tämän avulla käyttäjä voi halutessaan laukaista toiminnon, joka hakee rajapinnasta uusimmat tarjolla olevat tiedot. Tietojen päivittäminen painikkeella tulee kuitata käyttäjälle ponnahdusviestillä, jonka tarkoitus on ilmoittaa, että päivitys on onnistunut.

Ohjelman käytettävyyden kannalta on tärkeää, että käyttöliittymä suunnitellaan huolellisesti. Käyttöliittymää suunniteltaessa ensin tulee määritellä sarakkeet, joihin tunnusluvut sijoitetaan. Sarakkeiden sijoittelun tulee olla johdonmukaista. Mikäli sarakkeiden tiedot liittyvät läheisesti toisiinsa, kuten nykyhintaa, ostohintaa sekä hinnanmuutosta, tulee näiden olla sijoiteltuna vierekkäin.

Tietojen jaottelu sarakkeisiin mahdollistaa, että kaikkien käyttäjän asettamien osakkeiden tiedot saadaan asetettua samalle näkymälle. Mikäli käyttäjä joutuisi vaihtamaan sivua halutessaan tietoa eri osakkeista, olisi työkalun käyttö monimutkaisempaa. Yhden näkymän hyödyntämisellä pyritään myös säästämään työkalun käyttöön vaadittua aikaa. Osakkeiden listaaminen samalle sivulle luo kuitenkin haasteen, että näkymä pysyy selkeänä. Tämän vuoksi on tärkeää, että listan rakenne on pysyvä eikä vaihtelee osakkeiden kohdalla. Rivien tulee olla yhtä korkeita toisiinsa nähden, kuin myös solujen leveyden. Osakkeiden erottamiseksi toisistaan tulee hyödyntää värejä. Rivien värittäminen helpottaa rakenteen hahmottamista. Käytettävillä riveillä asetetaan ulkorajat, joiden avulla saadaan luotua laatikko, jonka sisällä käyttäjälle annettavat tiedot ovat.

Työkalun yksinkertaistaminen on suurelta osin kiinni käyttöliittymästä. Koska työkalun halutaan tarjoavan analyysiin vaaditut tiedot yhdessä näkymässä, tulee kaikki ylimääräinen karsia pois. Hyödynnettävien solujen ulkopuolelle jäävät alueet tulee värittää niin, että käyttäjän huomio ei kiinnittyisi siihen. Värinä tulee käyttää neutraalia vaaleaa, jonka avulla saadaan myös poistettua Excelin solujen väliset rajat. Kyseiselle alueelle tulee myös asettaa työkalun käyttöön vaaditut painikkeet. Painikkeiden tulee olla selvästi erottuvat ja pelkistetyt.

Työkalussa tehdyistä toiminnoista halutaan tallentaa tietoa ja koota ne yhteen historiatiedoiksi. Näiden tietojen avulla annetaan käyttäjälle mahdollisuus tarkistaa mitä on tehty ja milloin. Käyttäjän lisätessä osakkeen analyysityökalun listalle, tallennetaan tiedot päivämäärästä, osakkeiden määrästä, sekä osakekohtaisesti hinnasta. Listalta poistettaessa tallennetaan samat tiedot. Lisäyksistä ja poistoista tulee pitää kirjaa erillisinä listoina, jotta

tapahtumat saadaan eroteltua selkeästi. Historiatietoja ei tule sisällyttää samaan näkymään kuin osakelistaa, jotta ne eivät häiritse toisiaan. Historiatiedot tallennetaan automaattisesti.

Koodia järjestetään aliohjelmien (sub) avulla. Koska aliohjelmien ei tarvitse ohjelman toiminnan kannalta palauttaa arvoja, voidaan hyödyntää vain aliohjelmiä. Eri toiminnoille tulee luoda omat aliohjelmat. Näiden kutsuminen parantaa koodin luettavuutta sekä vähentää toisteisuutta. Lyhyemmällä ja yksinkertaisella koodilla ohjelman jatkokehitys on helpompaa ja virheiden löytäminen nopeampaa.

Avoimen lähdekoodin kirjasto VBA-JSON on omassa moduulissaan. Tämä selkeyttää koodin rakennetta. VBA-JSON kirjaston komentojen käyttäminen osana aliohjelmien koodia toteutetaan kutsumalla komentoja itse aliohjelmasta. Näiden komentojen avulla analyysityökalun ohjelmointiin vaadittavan uuden koodin määrä vähenee ja ohjelman rakentaminen nopeutuu.

Analyysityökalun koodi kirjoitetaan kokonaan englanniksi, jotta eri osat voidaan nimetä johdonmukaisesti ja ymmärrettävästi. Englanninkielen käyttäminen koodissa parantaa myös jatkokehitysmahdollisuuksia. Aliohjelmien, muuttujien sekä muiden koodin osien nimeäminen toteutetaan kuvaavasti. Annetun nimen tulee viitata toimintaan tai muuhun tietoon, johon nimetty asia on tarkoitettu. Esimerkiksi summaa kuvaava muuttuja nimetään tämän mukaisesti "Sum" -nimellä.

Lomakkeen koodipohjia, moduuleja, sekä työkirjan toimintaan liitettyä koodialustaa tulee hyödyntää ohjelman koodin rakenteessa. Työkalun eri toimintojen koodit tulee sijoittaa alustoihin niin, että sen toiminta vastaa sijaintia. Näin saadaan myös hyödynnettyä Excelin sisäänrakennettua ominaisuutta, jonka avulla haluttuja toimintoja voidaan ajaa ennalta määrätyissä tilanteissa, kuten työkalua käynnistettäessä. Moduulit ja pohjat sisällytetään niille asetettuihin kansioihin.

Koodin kommentoinnilla parannetaan sen ymmärrettävyyttä. Kommentit ovat lauseita, jotka kertovat muutamalla sanalla, mitä kommentin yhteydessä oleva koodi tekee. Kommenttien tarkoituksena on nopeuttaa koodin ymmärtämistä. Hyvän ohjelmoinnin kirjoitustapojen mukaisesti koodia myös sisennetään. Sisennyksien avulla pyritään parempaan luettavuuteen, joka hyödyttää ohjelman virheiden havaitsemisessa, kehittämisessä sekä rakenteiden ymmärtämisessä. Näillä keinoilla koodista saadaan helpommin ymmärrettävää ihmisiä varten.

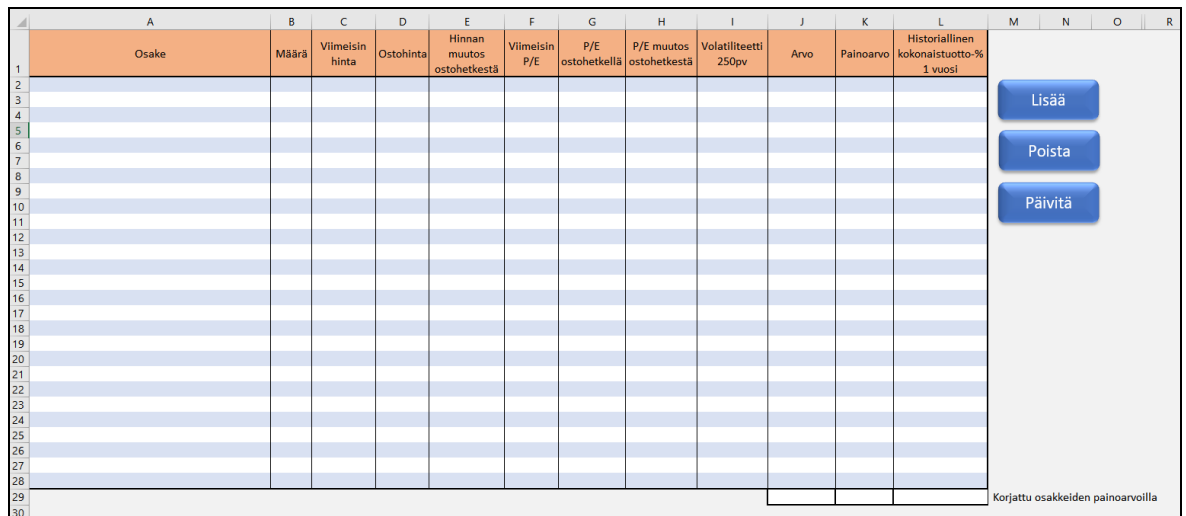
4.4 Toteutus

Tässä aluvussa kerrotaan analyysityökalun teknisen toteuttamisen eri vaiheista. Toteuttamiseen kuuluu Excel työkirjan puolella tehtävät toiminnot, sekä ohjelmointi.

4.4.1 Työkalun käyttöliittymä

Aloitin analyysityökalun rakentamisen luomalla käyttöliittymän, joka toimii työkalun päänäkymänä. Käyttöliittymä sisältää luvussa 4.3 kirjoitetut ominaisuudet. Nämä ominaisuudet toteutin Excelin sisältämällä työkaluilla.

Käyttöliittymän solujen väliset rajaukset toteutin Excelin rajaustyökalulla. Tässä hyödynnettiin eri vahvuisia rajoja. Paksulla ulkorajalla rajattiin käyttöliittymän alue, jonka sisään osakkeisiin liittyvät tiedot halutaan asettaa. Myös sarakkeita kuvaavien otsikoiden alle, sekä summalukuja varten asetettiin paksut rajaviivat. Näiden jälkeen väritin käyttöliittymän suunnitelman mukaisesti. Otsikkorivillä hyödynsin punertavaa pohjaväriä, sekä osakkeille varatuilla riveillä vaihdellen vaaleansinistä ja valkoista. Otsikkoriville kirjoitin valitut tekstit tekstityökalua hyödyntäen. Lopuksi lisäsin käyttöliittymään tarvittavat painikkeet, jotka sijoittelin irti toisistaan ja taulukon oikealle puolelle (Kuva 13).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	R
	Osake	Määrä	Viimeisin hinta	Ostohinta	Hinnan muutos ostohetkestä	Viimeisin P/E	P/E ostohetkellä	P/E muutos ostohetkestä	Volatiliteetti 250pv	Arvo	Painoarvo	Historiallinen kokonaistuotto-% 1 vuosi				
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																

Korjattu osakkeiden painoarvoilla

Kuva 13. Käyttöliittymän näkymä.

Käyttöliittymän luomisen jälkeen loin uuden välilehden historiatietoja varten. Välilehdelle loin pohjan tietojen lisäämiseen samoja tekniikoita ja työkaluja hyödyntäen kuin käyttöliittymää rakentaessani (Kuva 14).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
1			Lisäykset (Ostot)										Poistot (Myyntit)						
2	Pvm	Osake	Ostohinta	Määrä							Pvm	Osake	Myyntihinta	Määrä					
3																			
4																			
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			

Kuva 14. Historia -välilehden näkymä.

4.4.2 Datan noutaminen rajapinnasta

Seuraavaksi rupesin ohjelmoimaan työkalun toimintaa VBA koodilla. Aluksi oli haettava lista yhtiöiden nimistä ja näihin liitetyistä id numeroista. Tietojen hakemiseksi latasin GitHub verkkosivustolta VBA-JSON kirjaston, jonka asennettuani osaksi Exceliä voisin hyödyntää sen tarjoamia komentoja. Loin aliohjelman nimeltä GetData, johon kirjoitin tietojen hakemiseen tarvittavan koodin, joka rakentuu seuraavasti: url-osoitteen avulla koodi hakee http-sivulta tiedot Item nimisinä objekteina ja asettaa ne yhteen Json nimiseen objektiin. Kyseisen Json objektin sisältämien objektien määrä on ennalta tiedossa, joten hyödynsin For Each -toistorakennetta. Sen avulla käydään läpi jokainen Item objekti poimien halutut tiedot ja asettaen ne soluihin.

```

Sub GetData()

Dim I As Integer
I = 1
Dim Row As Integer
Dim MyRequest As Object
Dim Json As Object
Dim Item As Object

'tyhjennetään taulukko
Sheets("Tunnusluvut").Select
Range("A2").Select
Range(Selection, ActiveCell.SpecialCells(xlLastCell)).Select
Selection.ClearContents
Range("A2").Select

Set MyRequest = CreateObject("winhttp.winhttprequest.5.1")
MyRequest.Open "GET", "http://porssi.iltasanomat.fi/api/stocks/HEL/key_values?limit=150&sort=name%20down"
MyRequest.send
Set Json = ParseJson(MyRequest.ResponseText) 'Asetetaan Json objektiin haettu data

For Each Item In Json("data") 'Asetetaan valittu data soluihin
Cells(I + 1, 1).Value = Json("data")(I)("id") 'id
Cells(I + 1, 2).Value = Json("data")(I)("name") 'nimi
Cells(I + 1, 3).Value = Json("data")(I)("price") 'viimeisin hinta
Cells(I + 1, 4).Value = Json("data")(I)("performance") 'päivän muutos
Cells(I + 1, 5).Value = Json("data")(I)("date") 'pvm ja aika
I = I + 1
Next

Range("A:G").EntireColumn.AutoFit
Sheets("Työkalu").Select

End Sub

```

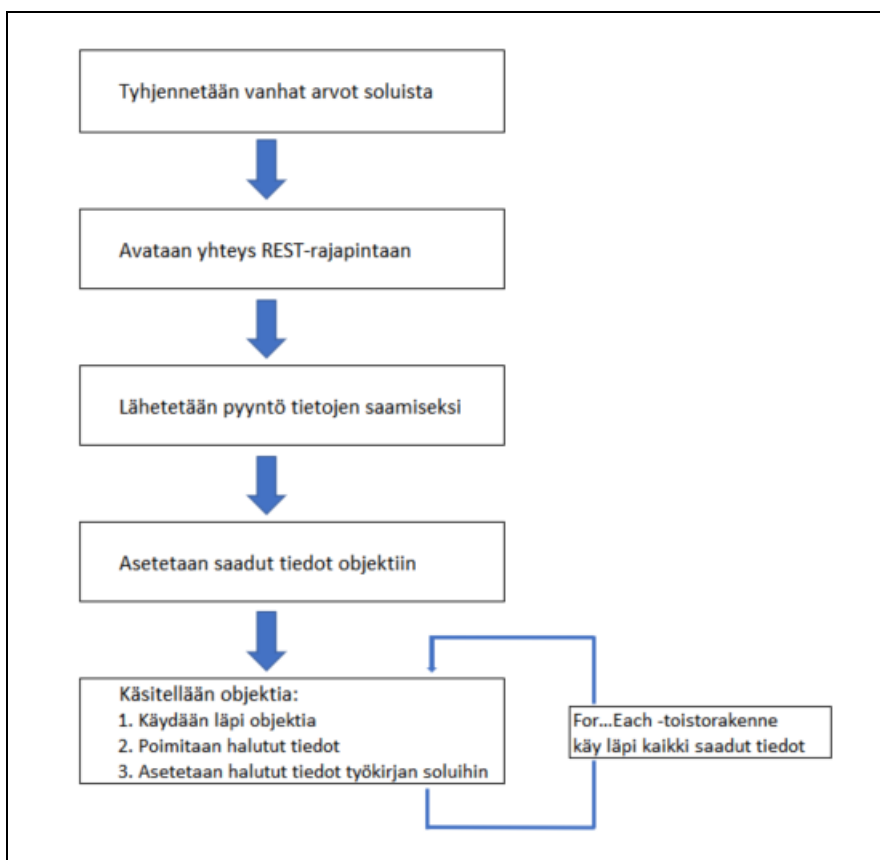
Kuva 15. GetData aliohjelman koodi kokonaisuudessaan.

Aliohjelma GetData (Kuva 15) tuottaa listan yhtiöiden nimistä, id numeroista, osakkeen viimeisimmästä hinnasta, hinnan muutoksesta sekä päivämäärästä ja ajasta (Kuva 16).

	A	B	C	D	E
1	Id	Company	Price	Päivän muutos	pvm & aika
2	44785	AFARAK GROUP OYJ	0,865	0,015	2017-12-27T11:04:44Z
3	73361930	AHLSTROM-MUNKSJÄ OYJ	18,18	0,04	2017-12-27T10:40:33Z
4	31028130	AKTIA BANK PLC A	9,1	-0,02	2017-12-27T11:03:54Z
5	31028131	AKTIA BANK PLC R	10,36	-0,14	2017-12-21T16:29:53Z
6	11697439	ALMA MEDIA CORPORATION	7,29	0,04	2017-12-27T10:02:25Z
7	194270	AMER SPORTS CORP	22,9	0,06	2017-12-27T11:07:08Z
8	44747	APETIT OYJ	14,23	-0,01	2017-12-27T11:03:05Z
9	127536162	ASIAKASTIETO GROUP OYJ NPV	23,8	0	2017-12-27T10:10:33Z
10	44780	ASPO PLC	9,39	0,05	2017-12-27T10:56:43Z
11	1544733	ASPOCOMP GROUP PLC	2,35	-0,05	2017-12-27T10:59:26Z
12	44760	ATRIA PLC A	11,81	-0,06	2017-12-27T10:56:15Z
13	44719	BANK OF ÅLAND PLC A	14,5	0	2017-12-27T10:38:52Z
14	44733	BANK OF ÅLAND PLC B	14	0	2017-12-27T10:41:51Z
15	194356	BASWARE CORPORATION	46,21	0,21	2017-12-27T10:55:41Z
16	44754	BIOHIT OYJ B	3,76	-0,07	2017-12-27T10:54:06Z
17	9245001	BITTIUM OYJ	5,67	0	2017-12-27T11:05:27Z
18	2779848	CAPMAN PLC B	1,77	0,02	2017-12-27T10:56:30Z
19	12093610	CARGOTEC OYJ	47,92	0,16	2017-12-27T11:03:23Z
20	82801610	CAVERION OYJ	5,88	-0,11	2017-12-27T11:07:11Z

Kuva 16. Esimerkki listasta, jonka aliohjelman GetData tuottaa erilliselle välilehdelle.

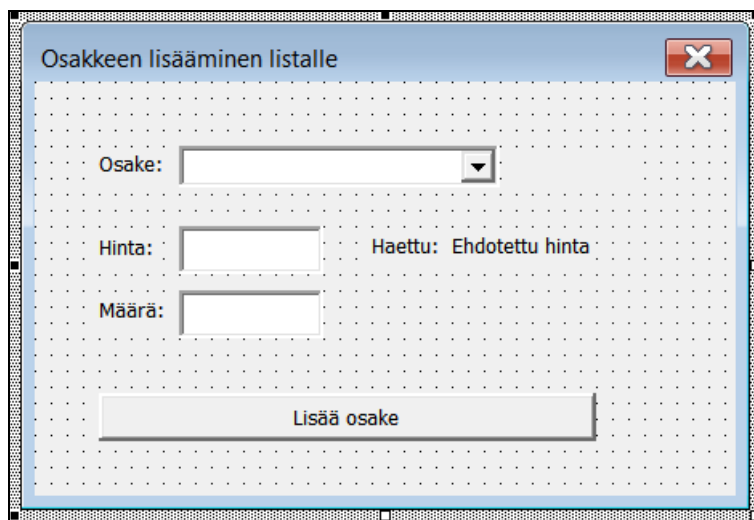
Datan noutaminen suoritetaan joka kerta saman kaavan mukaisesti ja samalla koodilla. Koodi on rakennettu interaktiiviseksi, joten sen kulkuun ei vaikuta esimerkiksi datan päivityminen (Kuva 17).



Kuva 17. Vaiheittainen kuvaus datan noutamisesta REST-rajapinnasta ja sen asettamisesta Excel työkirjan soluihin.

4.4.3 Lomakkeet

Seuraavaksi tein tarvittavat lomakkeet osakkeiden lisäämiseksi ja poistamiseksi listalta. Hyödynsin Excelin sisäänrakennetun lomaketyökalun ominaisuuksia. Lisäyslomakkeeseen lisäsin pudotusvalikon, tekstikenttiä, nimikkeitä sekä painikkeen. Kaikki edellä mainituista nimettiin suunnitelman mukaisesti, sekä aseteltiin lomakepohjaan (Kuva 18). Poistolomakkeeseen lisäsin tarpeiden mukaisesti tekstikenttiä, nimikkeitä sekä painikkeen (Kuva 19).



Osakkeen lisääminen listalle

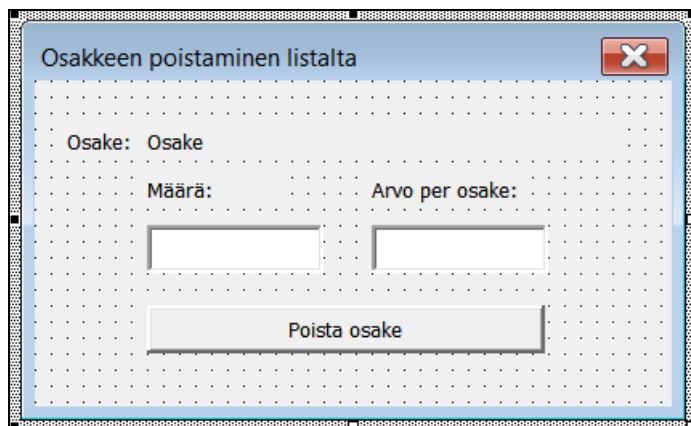
Osake:

Hinta: Haettu: Ehdotettu hinta

Määrä:

Lisää osake

Kuva 18. Lisäyslomakkeen asettelu.



Osakkeen poistaminen listalta

Osake: Osake

Määrä: Arvo per osake:

Poista osake

Kuva 19. Poistolomakkeen asettelu.

Ohjelmoin molempien lomakkeiden toiminnan niille tarkoitettuihin pohjiin. Lisäyslomakkeen toiminnan ohjelmoinnin aloitin luomalla aliohjelman nimeltä UserForm_Activate, jonka sisältämä koodi suoritetaan, kun käyttäjä avaa lomakkeen käyttöliittymässä olevasta

painikkeesta. Kyseinen aliohjelma sisältää koodin, joka asettaa lomakkeen pudotusvalikkoon työkalun käynnistyksen ohessa haettujen yhtiöiden nimet. Nimien asettaminen pudotusvalikkoon suoritetaan Do While -toistorakenteella, koska yhtiöiden määrä saattaa vaihdella (Kuva 20).

```
Private Sub UserForm_Activate()  
  
Dim Row As Integer  
Row = 1  
Dim Value As String  
  
Do  
    Row = Row + 1  
    Value = Sheets("Tunnusluvut").Range("B" & Row).Value  
    If Value <> "" Then  
        CMBShares.AddItem Value  
    End If  
Loop While Value <> ""  
  
CMBShares.ListIndex = 0  
CMBShares.Style = fmStyleDropDownList  
  
End Sub
```

Kuva 20. Aliohjelma UserForm_Activate, joka asettaa pudotusvalikkoon yhtiöiden nimet.

Lisäyslomakkeen taakse on ohjelmoitu edellä mainittujen lisäksi aliohjelma, joka käyttäjän muuttaessa pudotusvalikon valintaa, tarkastelee yhtiöiden listaa ja tekee tarvittavia toimenpiteitä. Näihin toimenpiteisiin kuuluu lisäyslomakkeen sisältämän osakkeen ehdotetun hinnan sekä id numeron noutaminen (Kuva 21).

```
Private Sub CMBShares_Change()  
  
Application.ScreenUpdating = False  
Sheets("tunnusluvut").Select  
Cells.Find(What:=CMBShares.Text, After:=ActiveCell, LookIn:=xlFormulas, LookAt _  
:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, MatchCase:= _  
False, SearchFormat:=False).Activate  
LBLSuggested = Range("C" & ActiveCell.Row).Value  
LBLId = Range("A" & ActiveCell.Row).Value  
Sheets("työkalu").Select  
Application.ScreenUpdating = True  
  
End Sub
```

Kuva 21. Pudotusvalikon toimintaan liitetyt toiminnot.

Aloitin poistolomakkeen ohjelmoinnin luomalla aliohjelman UserForm_Activate, suoritetaan kun lomake avataan. Kyseiseen aliohjelmaan ohjelmoin ehtolauseita hyödyntäen toiminnallisuuden, joka tarkistaa onko käyttäjän aloittama osakkeen poistaminen listalta mahdollista. Mahdollisuus osakkeen poistamiseen listalta tarkistetaan valitun solun perusteella. Mikäli valittu solu ei ole sallittujen rajojen sisällä tai se on tyhjä, kehoitetaan käyttäjää virheviestillä korjaamaan virhe.

```

Private Sub UserForm_Activate()

'Tarkastetaan onko solu valittu sekä onko se sallituissa rajoissa.
  If ActiveCell.Value = "" Then
    FRMRemove.Hide
    MsgBox "Valitse poistettava osake!", vbCritical
    Exit Sub
  End If

  If ActiveCell.Row > 28 Or ActiveCell.Row < 2 Then
    FRMRemove.Hide
    MsgBox "Valitse poistettava osake!", vbCritical
    Exit Sub
  End If

  If ActiveCell.Column <> 1 Then
    FRMRemove.Hide
    MsgBox "Valitse poistettava osake!", vbCritical
    Exit Sub
  End If

  LBLShareFetched.Caption = ActiveCell.Value

End Sub

```

Kuva 22. Poistolomakkeen aktivoituessa tarkistetaan käyttäjän aktivoima solu.

Mikäli käyttäjä on onnistuneesti valinnut poistettavan osakkeen, ohjelma jatkaa poistolomakkeen näyttämistä. Käyttäjältä kerätään lomakkeen tekstikenttiin tarvittavat tiedot poiston suorittamiseksi. Poistaminen suoritetaan painikkeesta ”Poista osake”. Painikkeen painaminen kutsuu aliohjelmaa nimeltä `BTNRemove_Click`. Aliohjelmaan ohjelmoin ehtolauseita hyödyntäen tarkistuksia, jotta työkalu voi suorittaa poistamisen ilman virheitä. Ehtolauseiden toimintaa on kuvattu koodissa kommentteilla (Kuva 23).

```

Private Sub BTNRemove_Click()

'Tarkistetaan onko määrä tyhjä tai onko siinä kirjaimia
  If TXTAmount.Value = "" Or IsNumeric(TXTAmount.Value) = False Then
    MsgBox "Tarkista syötetty määrä!", vbCritical
    TXTAmount.SetFocus
    Exit Sub
  End If

'Tarkistetaan onko määrä alle 0
  If TXTAmount.Value < 0 Then
    MsgBox "Määrä ei voi olla negatiivinen!", vbCritical
    TXTAmount.SetFocus
    Exit Sub
  End If

'Tarkistetaan onko määrä 0
  If TXTAmount.Value = 0 Then
    MsgBox "Määrä ei voi olla nolla!", vbCritical
    TXTAmount.SetFocus
    Exit Sub
  End If

'Tarkistetaan onko annettu määrä liian suuri
  If CLng(TXTAmount.Value) > CLng(Range("B" & ActiveCell.Row).Value) Then
    MsgBox "Määrä on liian suuri!", vbCritical
    TXTAmount.SetFocus
    Exit Sub
  End If

'Tarkistetaan onko kirjaimia tai arvo alle 0
  If IsNumeric(TXTValue.Value) = False Or TXTValue < 0 Then
    MsgBox "Arvoksi tulee antaa vain numeroita!", vbCritical
    TXTValue.SetFocus
    Exit Sub
  End If

```

Kuva 23. Poistamiseen liitetyt ehtolauseet.

Kuvan 20 ehtolauseiden lisäksi ohjelmoin aliohjelmaan BTNRemove_Click ehtolauseen, joka tarkistaa onko poistettavien osakkeiden lukumäärä yhtä suuri kuin listalla olevien kyseisten osakkeiden lukumäärä. Mikäli määrät ovat yhtä suuret, tulee ohjelman poistaa osake työkalun näkymästä. Tämän jälkeen työkalun näkymän osakkeet järjestellään tyhjiä välilyöntejä poistamiseksi. Aliohjelma BTNRemove_Click myös lisää onnistuneista poistoista tiedot Historia -välilehdelle.

```
'Tarkistetaan myydäänkö kaikki osakkeet. Jos myydään, siirretään listassa alla olevia osakkeita ylöspäin
If CLng(TXTAmount.Value) = CLng(Range("B" & ActiveCell.Row).Value) Then
    Module1.AddToHistory LBLShareFetched.Caption, TXTAmount.Value, CDb1(TXTValue.Text), True
    RemoveShare
    Dim Row As Integer
    For Row = ActiveCell.Row To 27
        Range("A" & Row).Value = Range("A" & Row + 1).Value
        Range("B" & Row).Value = Range("B" & Row + 1).Value
        Range("C" & Row).Value = Range("C" & Row + 1).Value
        Range("D" & Row).Value = Range("D" & Row + 1).Value
        Range("E" & Row).Value = Range("E" & Row + 1).Value
        Range("F" & Row).Value = Range("F" & Row + 1).Value
        Range("G" & Row).Value = Range("G" & Row + 1).Value
        Range("H" & Row).Value = Range("H" & Row + 1).Value
        Range("I" & Row).Value = Range("I" & Row + 1).Value
        Range("J" & Row).Value = Range("J" & Row + 1).Value
        Range("K" & Row).Value = Range("K" & Row + 1).Value
        Range("L" & Row).Value = Range("L" & Row + 1).Value
        Range("P" & Row).Value = Range("P" & Row + 1).Value
        Range("Q" & Row).Value = Range("Q" & Row + 1).Value

    Next

Else
    Range("B" & ActiveCell.Row) = Range("B" & ActiveCell.Row) - TXTAmount.Value
    Module1.AddToHistory LBLShareFetched.Caption, TXTAmount.Value, CDb1(TXTValue.Text), True
End If

TXTAmount.Value = ""
TXTValue.Value = ""
CalculateChanges
FRMRemove.Hide
```

Kuva 24. Aliohjelman BTNRemove_Click sisältämä ehtolause.

4.4.4 Aliohjelmat laskuihin ja päivityksiin

Loin koodin järjestämistä varten uuden moduulin nimeltä Module1, johon tulisin sijoittamaan kaikki aliohjelmat, jotka suorittavat työkalun laskutoimituksia ja päivityksiä. Ensimmäinen aliohjelma oli nimeltään UpdateOnStart, joka päivittää työkaluun valittujen osakkeiden tiedot. Päivitys tapahtuu, kun ohjelma avataan. Aliohjelma hyödyntää Do Until toistorakennetta, koska tarkasteltavien rivien määrä saattaa vaihdella. Toistorakenteen sisään ohjelmoin komentoja, jotka hakevat yhtiön id numeron perusteella rajapinnasta tietoa. Tietojen hakemisessa hyödynsin VBA-JSON kirjaston komentoja.

```

Sub UpdateOnStart ()

Dim MyRequest As Object
Dim Json As Object
Dim Item As Object
Dim RowNumber As Integer
RowNumber = 2
Dim Id As String

Sheets("Työkalu").Select
Range("A" & RowNumber).Select

Do Until ActiveCell.Value = ""
Set MyRequest = CreateObject("winhttp.winhttprequest.5.1")
Id = CStr(Range("P" & RowNumber).Value)
MyRequest.Open "GET", "http://porssi.iltasanomat.fi/api/stocks/" & Id
MyRequest.send

Set Json = ParseJson(MyRequest.ResponseText) 'Asetetaan Json objektiin haettu data

Range("C" & ActiveCell.Row).Value = Json("price") 'päivitetään viimeisin hinta
Range("F" & ActiveCell.Row).Value = Json("info")("keyValues")("pe") 'päivitetään P/E -luku
Range("I" & ActiveCell.Row).Value = Json("info")("vola250") 'päivitetään vola250
Range("L" & ActiveCell.Row).Value = Json("info")("performance")("year") / 100 'päivitetään kokonaistuotto-% 1 vuosi

RowNumber = RowNumber + 1
Range("A" & RowNumber).Select

Loop

CalculateChanges

End Sub

```

Kuva 25. UpdateOnStart aliohjelman koodi.

Seuraavaksi loin aliohjelman CalculateChanges, jota myös kuvan 25 aliohjelman UpdateOnStart kutsuu. CalculateChanges hyödyntää Do Until toistorakennetta työkalun osakkeiden läpikäymiseksi. Se laskee haettujen tietojen perusteella niistä johdettavat luvut hinnan muutos ostohetkestä, P/E muutos ostohetkestä, arvo, sekä näistä laskettavat summat. Työkalu siis hyödyntää CalculateChanges aliohjelmää pitämään lasketut luvut ajan tasalla.

```

Sub CalculateChanges ()

Sheets("Työkalu").Select

Dim RowNumber As Integer
RowNumber = 2

Range("A2").Select
Do Until ActiveCell.Value = ""
'hinnan muutos ostohetkestä
Range("E" & RowNumber).Value = (Range("C" & RowNumber).Value - Range("D" & RowNumber).Value) / Range("D" & RowNumber).Value
Range("E" & RowNumber).NumberFormat = "0.00%"
Range("H" & RowNumber).Value = Range("F" & RowNumber).Value - Range("G" & RowNumber).Value 'p/e muutos ostohetkestä
Range("J" & RowNumber).Value = Range("B" & RowNumber).Value * Range("C" & RowNumber).Value 'arvo
Range("J29").Value = Application.Sum(Range("J2:J28").Value)
Range("K" & RowNumber).NumberFormat = "0.00%"
RowNumber = RowNumber + 1
Range("A" & RowNumber).Select

Loop

CalculateWeights
CalculateWeightedPerformance

End Sub

```

Kuva 26. CalculateChanges aliohjelman koodi.

Kuvan 26 aliohjelman CalculateChanges kutsumat CalculateWeights ja CalculateWeightedPerformance aliohjelmat loin laskemaan osakkeiden painoarvot, sekä niiden avulla korjatut kokonaistuottoprosentit. Molemmat hyödyntävät Do Until toistorakennetta työkalun osakkeiden läpikäymiseen, suorittaen samalla laskutoimituksia. CalculateWeights ja CalculateWeightedPerformance aliohjelmien toimintaa on kuvattu kuvissa 27 ja 28.

```

Sub CalculateWeightedPerformance()

Dim Sum As Double
Dim RowNumber As Integer
RowNumber = 2

Range("A2").Select

Do Until ActiveCell.Value = ""
    Sum = Sum + (Range("K" & ActiveCell.Row).Value * Range("L" & ActiveCell.Row).Value)
    RowNumber = RowNumber + 1
    Range("A" & RowNumber).Select
Loop

Range("L29").Value = Sum

End Sub

```

Kuva 27. Aliohjelman CalculateWeightedPerformance koodi.

```

Sub CalculateWeights()

Sheets("Työkalu").Select

Dim RowNumber As Integer
RowNumber = 2
Range("A2").Select

Do Until ActiveCell.Value = ""
    'painoarvo
    Range("K" & RowNumber).Value = CDb1(Range("J" & RowNumber).Value) / CDb1(Range("J29").Value)
    Range("K29").Value = Application.Sum(Range("K2:K28").Value)
    RowNumber = RowNumber + 1
    Range("A" & RowNumber).Select
Loop

End Sub

```

Kuva 28. Aliohjelman CalculateWeights koodi.

Ohjelmoin Module1:n sisään myös aliohjelman nimeltä AddToHistory, jonka tarkoituksen on vastaanottaa tietoa ja käsitellä sitä niin, että tiedot tallentuvat Historia -välilehdelle. AddToHistory aliohjelma vastaanottaa Boolean -tyyppisen Sell muuttujan, jonka perusteella se tallentaa tiedot joko poisto- tai lisäyslistaan. Tietojen asettaminen soluihin Historia välilehdellä vaatii seuraavan vapaana olevan rivin etsimistä. AddToHistory löytää vapaan rivin kolmen eri askeleen avulla. Ensimmäinen askel valitsee koko työkirjan viimeisen solun tietystä sarakkeesta. Toinen askel Excelin End(xlUp) komentoa, joka etsii ylöspäin liikuttaessa ensimmäinen ei-tyhjän solun. Kolmas askel hyödyntää Offset komentoa, joka liikuttaa solun valintaa haluttuun suuntaan halutun määrän askelia suhteessa alkuperäiseen valintaa (Kuva 29).

```

Sub AddToHistory(Name As String, Count As Integer, Price As Double, Sell As Boolean)
Sheets("Historia").Select

    If Sell = True Then
        Range("K1048576").Select
        Selection.End(xlUp).Select
        ActiveCell.Offset(1, 0).Range("A1").Select
        Range("K" & ActiveCell.Row).Value = Date
        Range("L" & ActiveCell.Row).Value = Name
        Range("M" & ActiveCell.Row).Value = Price
        Range("N" & ActiveCell.Row).Value = Count
    Else
        Range("A1048576").Select
        Selection.End(xlUp).Select
        ActiveCell.Offset(1, 0).Range("A1").Select
        Range("A" & ActiveCell.Row).Value = Date
        Range("B" & ActiveCell.Row).Value = Name
        Range("C" & ActiveCell.Row).Value = Price
        Range("D" & ActiveCell.Row).Value = Count
    End If

    Sheets("Työkalu").Select

End Sub

```

Kuva 29. Aliohjelman AddToHistory koodi.

Viimeisenä module1:n sisään loin vielä aliohjelman nimeltä CalcChangesOnBtn, joka kutsuu tarvittavia aliohjelmaa tietojen päivittämiseen. Liitin kyseisen aliohjelman työkalun käyttöliittymän "Päivitä" painikkeeseen, jonka avulla käyttäjä voi päivittää tiedot halutessaan. CalcChangesOnBtn ilmoittaa käyttäjälle suoritettuaan päivitykset ilmoituslaatikkoa hyödyntäen (Kuva 30).

```

Sub CalcChangesOnBtn()
Application.ScreenUpdating = False

UpdateOnStart
CalculateChanges

Application.ScreenUpdating = True

MsgBox "Tiedot päivitetty"

End Sub

```

Kuva 30. Aliohjelma CalcChangesOnBtn koodi.

4.5 Toteutuksen testaaminen

Työkalua toteuttaessani suoritin spiraalimallin mukaisesti testauksia eri vaiheissa. Aliohjelmin kohdennetuilla testauksilla pyrin varmistamaan niiden virheettömän toiminnan. Testaukset toteutin kehitysympäristössä kaksivaiheisina. Ensin suoritin aliohjelman toiminnan ilman yksittäisiä askelia, jolloin voidaan todeta, kääntyykö koodi virheettömästi ja tuottaa-ko se halutun lopputuloksen. Tämän jälkeen suoritin aliohjelman rivi kerrallaan, jonka avulla todensin jokaisen yksittäisen rivin toiminnan. Aliohjelmissa tapahtuvat virheet kehi-

tysvaiheessa korjasin heti niiden ilmentyessä joko muokkaamalla aiemmin kirjoitettua koodia tai palaamalla aiempaan versioon ja kirjoittamalla aliohjelman uusiksi.

Suoritin testausta myös työkalun käyttöliittymässä. Tämän avulla etsin virheitä työkalun toiminnasta käyttäjän näkökulmasta. Analysoin käyttöliittymässä aiheutetut mahdolliset virhetilanteet, jonka jälkeen tein tarvittavat korjaustoimenpiteet. Korjaustoimenpiteiden jälkeen aloitin testauksen alusta, ensin kehitysympäristössä ja sen jälkeen käyttöliittymässä.

Suoritin testausta myös toisella Windows tietokoneella, jonka avulla pyrin löytämään eroavaisuuksia toiminnassa. Käsittelin erot toiminnassa muiden virhetilanteiden tapaan analysoiden ja korjaustoimenpitein. Testaamisen ja korjaustoimenpiteiden kiertokulkua jatkoin niin pitkään, että ongelmia ei enää ilmennyt normaalissa käytössä.

Osa	Määrä	Viimeisin hinta	Hinta ostohetkellä	Hinnan muutos ostohetkestä	Viimeisin P/E	P/E ostohetkellä	P/E muutos ostohetkestä	Volatiliteetti 250pv	Arvo	Painoarvo	Historiallinen kokonaistuotto-% 1 vuosi	
METSO OYJ	50	29,78 €	29,75 €	0,10 %	34,22	34,22	0,00	23,76	1489,00	39,14 %	-8,61 %	
OUTOKUMPU OYJ	50	8,03 €	8,03 €	0,02 %	22,93	22,93	0,00	38,02	401,40	10,55 %	-2,27 %	Lisää
TIETO CORPORATION	30	28,18 €	27,00€	4,37 %	19,29	19,29	0,00	20,67	845,40	22,22 %	7,76 %	Poista
NOKIAN TYRES	20	38,68 €	35,23 €	9,79 %	20,67	20,67	0,00	17,72	773,60	20,33 %	6,91 %	Päivitä
DIGIA PLC	120	2,46 €	2,89 €	-14,88 %	0,67	0,67	0,00	27,33	295,20	7,76 %	-16,68 %	
									3804,60	100 %	4,96 %	Korjattu osakkeiden painoarvoilla

Kuva 31. Analyysityökalun testikäyttöä eri yhtiöiden osakkeilla.

Lisäykset (Ostot)				Poistot (Myyntit)			
Pvm	Osa	Ostohinta	Määrä	Pvm	Osa	Myyntihinta	Määrä
10.1.2018	METSO OYJ	29,75	100	10.1.2018	ROVIO ENTERTAINMENT OYJ NPV	9,98	60
10.1.2018	OUTOKUMPU OYJ	8,026	50	10.1.2018	METSO OYJ	28,75	50
10.1.2018	ROVIO ENTERTAINMENT OYJ NPV	9,92	60				
10.1.2018	TIETO CORPORATION	27	30				
10.1.2018	NOKIAN TYRES	35,23	20				
10.1.2018	DIGIA PLC	2,89	120				

Kuva 32. Analyysityökalun testikäytössä tallennettuja historiatietoja.

5 Pohdinta

Tavoitteena oli luoda analyysityökalu, jonka automaattinen toiminta ja muut ominaisuudet hyödyttävät osakesijoittajaa portfolion analysoinnissa. Tässä pyrittiin hyödyntämään avointa rajapintaa sekä ohjelmointia. Aikaansaatu analyysityökalu täyttää nämä kriteerit. Työkalun aikaansaaminen edellytti montaa eri vaihetta. Ennen kuin työkalua pystyttiin suunnittelemaan, oli tunnettava rahoituksen teoriaa. Tämän jälkeen oli tunnettava ohjelmoinnin ja ohjelmistokehityksen teoriaa, sekä osattava hyödyntää näitä.

Koulutukseni aikana Haaga-Heliassa olen saanut kattavasti tietoa rahoituksesta, joten tiedon hankkimista ei tarvinnut aloittaa tyhjästä. Valmiiden tietojen lisäksi keräsin osakesijoittamiseen liittyvää uutta tietoa, joiden avulla onnistuin löytämään tekijöitä, jotka ovat hyödyllisiä osakkeiden analysoinnissa. Näihin kuului muun muassa tunnusluvut ja rahoituksen teoriat. Tiedonhankinnan edetessä varmistui, minkälainen osakeanalyysityökalun tulisi olla. Haasteena oli kuitenkin poimia työkaluun vain hyödyllisimmät tiedot. Näistä tiedoista onnistuin rajaamaan tärkeimmät, jotta työkalun sisältö vastaisi tehokasta osakkeiden analysointia.

Analyysityökalu saavutti sille asetetut tavoitteet. Näihin tavoitteisiin päästiin hyödyntämällä spiraalista ohjelmiston kehitysmallia. Erityisen tärkeää onnistumisessa oli runsas testaaminen. Pienten osien testauksella havaittiin monia virheitä ohjelman toiminnassa, jotka olisivat voineet vaikuttaa suuresti käyttäjän saamaan hyötyyn työkalusta. Testauksen lisäksi yksi onnistumisen tekijöistä oli eri vaiheiden tarkastelu tavoitteiden pohjalta. Vastaamalla kysymykseen ”Vastaako työkalun toiminta tässä vaiheessa sille aiemmin asetettuja tavoitteita?” voitiin havaita, jos ohjelman toteutus poikkesi halutulta polulta.

5.1 Analyysityökalun hyödyt

Työkalun automatisoidulla tiedonhaulla saavutettiin ajallinen hyöty osakkeita analysoidessa. Sijoittajan näkökulmasta säästynyt aika helpottaa sijoitusten seuraamista. Työkalun avulla sijoittaja voi karsia mahdollisen internet sivujen selaamisen, kun hakee tietoa osakkeiden tilasta. Ajallisen hyödyn mittaaminen on kuitenkin ongelmallista. Vaikuttavia tekijöitä on muun muassa tietokoneen suoritusteho, käyttäjän henkilökohtainen osaaminen, sekä internetyhteyden nopeus. Nämä tekijät eivät kuitenkaan kumoa työkalusta saatavia hyötyjä. Esimerkiksi hidas internetyhteys vaikuttaa molemmissa tilanteissa, oli sitten verkkosivujen selaajana käyttäjä tai työkalu. Työkalun etuna verkkosivuja selatessa kuitenkin on, että sen hyödyntämät sivut eivät sisällä mainoksia tai muuta ylimääräistä, jonka laa-
taamiseen menee aikaa.

Osakesijoittaja voi hyödyntää työkalua Helsingin pörssin päälistan osakkeiden kohdalla. Työkalun suorittamat automaattiset laskutoimitukset antavat hyödyllistä tietoa portfolioon tilasta. Sijoittajan ei siis tarvitse itse laskea mitään, vaan ainoastaan valita minkä osakkeiden tietoja työkalun halutaan näyttävän. Kyseiset tiedot päivittyvät automaattisesti, joten sijoittajan ei tarvitse itse niitä miettiä. Työkalun avulla sijoittaja voi muuttaa toimintatapansa, jolla analysoi osakkeita. Tehokkaampi ja varmatoimisempi toimintatapa hyödyttää sijoittajaa.

Analyysityökalun testikäytön aikana huomasin, kuinka yksittäisten osakkeiden tiedot saattavat vaikuttaa yllättävästi portfolioon kokonaisuutena. Esimerkiksi tilanne, jossa portfolio sisältää merkittävän määrän pienen kokonaistuottoprosentin omaavaa yhtiötä ja samaan aikaan pienemmän määrän suuren tuottoprosentin yhtiötä. Heikosti tuottava yhtiö vaikuttaa kyseisessä tilanteessa hyvin paljon portfolioon yhteensä laskettuun tuottoprosenttiin (Kuva 33).

Osake	Määrä	Viimeisin hinta	Hinta ostohetkellä	Hinnan muutos ostohetkestä	Viimeisin P/E	P/E ostohetkellä	P/E muutos ostohetkestä	Volatiliteetti 250pv	Arvo	Painoarvo	Historiallinen kokonaistuotto-% 1 vuosi
CAVERION OYJ	500	6,21 €	6,52 €	-4,75 %	-24,84	-24,80	-0,04	27,06	3105,00	50,11 %	-22,55 %
FORTUM CORPORATION	180	17,18 €	15,00 €	14,50 %	30,67	30,66	0,01	19,26	3091,50	49,89 %	15,67 %
									6196,50	100 %	-3,48 %

Korjattu osakkeiden painoarvoilla

Lisää

Poista

Päivitä

Kuva 33. Eri kokonaistuottoprosentin omaavia yhtiöitä portfolioissa. Osakkeiden yhteenlaskettu arvo molempien yhtiöiden kohdalla on melkein sama.

5.2 Oman oppimisen arviointi

Tiedonhaku työtä tehdessä oli sujuvaa. Opintojeni avulla pystyin nopeasti löytämään tarvitsemani tiedon. Opin kuitenkin paljon lisää asioita liittyen osakesijoittamiseen ja analysointiin. Näiden aiheiden tunteminen on mielestäni hyödyllistä henkilökohtaisten sijoitusten ja raha-asoiden hoidossa.

Analyysityökalun ohjelmoiminen oli haasteellista, mutta opin siitä paljon. Tietojenkäsittelyn monet eri aiheet vaativat syvällistä ymmärtämistä, jota työn myötä saavutin. Huomasin myös hyödylliseksi eri vaiheiden raportoimisen. Asioihin palaaminen jälkeensä ja niiden pohtiminen auttoivat ymmärtämään laajemmin niiden vaikutusta. Esimerkiksi monimut-

kaisten ohjausrakenteiden hahmottaminen helpottui, kun aiheesta raportoi. Saamiani tietoja on mahdollista hyödyntää myös tulevaisuudessa eri ohjelmistoprojekteissa.

Tietoperustaa kirjoittaessani opin paljon uutta ohjelmointiin liittyen. Käsitteiden ymmärtäminen auttaa ymmärtämään kokonaiskuvia, joita ohjelmoinnissa tarvitaan. Jotta on mahdollista rakentaa ja kehittää laadukkaita ratkaisuja eri ongelmiin, on ymmärrettävä mistä eri aihealueet ongelmaan liittyen johtuvat. Työtä tehdessäni opin monia uusia ratkaisuja eri ohjelmointiin liittyviin ongelmiin, joka on varmasti eduksi tulevaisuudessa.

Opinnäytetyön kirjoittaminen oli opettavaista. Ymmärrän työn myötä, miten raportin rakenne koostuu ja mitä sen on hyvä sisältää. Myös opinnäytetyön kirjoitusprosessi oli opettavaista. Tekstin tuottaminen ja lähteiden oikeaoppinen käyttö paranivat opinnäytetyötä kirjoittaessani.

Lähteet

The Balance 2017. Using the Price-to-Earnings Ratios as a Quick Way to Value a Stock.

The New Investor's Guide to Understanding the P/E Ratio. Luettavissa:

<https://www.thebalance.com/using-price-to-earnings-356427>. Luettu: 16.11.2017.

Birnbaum, D. 2005. Microsoft Excel VBA Programming for the Absolute Beginner, Second Edition. Birnbaum, Duane. Course Technology PTR 2005.

Expertiza 2012. The Spiral Model. Luettavissa:

http://wiki.expertiza.ncsu.edu/index.php/CSC/ECE_517_Fall_2012/SM_sm. Luettu:

4.12.2017.

Folger, J. 2017. How are share prices set? Jean Folger. Luettavissa:

<http://www.investopedia.com/ask/answers/12/how-are-share-prices-set.asp>. Luettu:

26.9.2017.

Finanssivalvonta 2.1.2012. Volatiliteetti kertoo historiallisesta arvonheilahtelusta. Luetta-

vissa: http://www.finanssivalvonta.fi/fi/Finanssiasiakas/Artikkelit/Pages/05_2011.aspx. Lu-

ettu: 16.11.2017.

Finanssivalvonta 2015. Osakesijoituksiin liittyvät riskit. Luettavissa:

<http://www.finanssivalvonta.fi/fi/Finanssiasiakas/Tuotteita/Sijoittaminen/Osakkeet/Riskit/Pages/Default.aspx>. Luettu: 7.1.2018.

Green, J. 2007. Excel 2007 VBA Programmer's Reference. Green, John. 2007.

Hall, T. 2017. VBA-JSON. JSON conversion and parsing for VBA. Hall, Tim. Luettavissa:

<https://github.com/VBA-tools/VBA-JSON>. Luettu: 25.9.2017.

Hobson, R. 2012. Shares Made Simple: A Beginner's Guide to the Stock Market. Hobson Rodney. Harriman House.

Hunter, K. 2016. Irresistible APIs: Designing web APIs that developers will love. Hunter,

Kirsten L. Manning Publications 2016.

Investopedia.com, Modern Portfolio Theory. Luettavissa:

<https://www.investopedia.com/terms/m/modernportfoliotheory.asp>. Luettu: 8.1.2018.

Investopedia.com, Portfolio. Luettavissa:

<https://www.investopedia.com/terms/p/portfolio.asp>. Luettu: 8.11.2017.

Investopedia.com, Price-Earnings Ratio. Luettavissa:

<https://www.investopedia.com/terms/p/price-earningsratio.asp>. Luettu: 15.11.2017.

Investopedia.com, Total Return. Luettavissa:

<https://www.investopedia.com/terms/t/totalreturn.asp>. Luettu: 15.11.2017.

Investopedia.com, Volatility. Luettavissa:

<https://www.investopedia.com/terms/v/volatility.asp>. Luettu: 15.11.2017.

Jelen, B & Syrstadt, T. 2016. Excel 2016 VBA and macros. Jelen, Bill & Syrstadt Tracy. Apress. 2016.

Json.org. Introducing JSON. Luettavissa: <http://www.json.org/>. Luettu: 8.11.2017.

Knüpfer, S. & Puttonen, V. 2014. Moderni rahoitus. Knüpfer, Samuli & Puttonen, Vesa. Talentum Media 2014.

Morningstar.com, Standard Deviation. Luettavissa:

http://www.morningstar.com/InvGlossary/standard_deviation.aspx. Luettu: 15.11.2017.

Morningstar.com, Total Return. Luettavissa:

<http://www.morningstar.com/InvGlossary/total-return.aspx>. Luettu: 15.11.2017.

Microsoft 2015a. Loop Structures (Visual Basic). Control Flow. Microsoft .NET Documentation. Luettavissa:

<https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/control-flow/loop-structures>. Luettu: 20.11.2017.

Microsoft 2015b. For Each...Next Statement (Visual Basic). Microsoft .NET Documentation. Luettavissa:

<https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/statements/for-each-next-statement>. Luettu: 20.11.2017.

Microsoft Developer Network. Loop Structures. Visual Basic Concepts. Luettavissa: [https://msdn.microsoft.com/en-us/library/aa263312\(v=vs.60\).aspx](https://msdn.microsoft.com/en-us/library/aa263312(v=vs.60).aspx). Luettu: 16.11.2017.

Microsoft Developer Network 2017. Getting Started with VBA in Office, 2017. Luettavissa: <https://msdn.microsoft.com/vba/office-shared-vba/articles/getting-started-with-vba-in-office>. Luettu: 12.9.2017.

Niskanen & Niskanen 2013. Yritysrahoitus. Niskanen, Jyrki & Niskanen, Mervi. Edita Publishing Oy 2013.

Office VBA Macros You Can Use Today 2006. Office VBA Macros You Can Use Today: Over 100 Amazing Ways to Automate Word, Excel, PowerPoint, Outlook, and Access. González, Juan Pablo & Meister, Cindy & Suat M. Ozgur & Bill Dilworth & Anne Troy & Nico Altink. 2006.

Uzayr, S. 2016. Learning WordPress REST API. Uzayr, Sufyan bin. Packt Publishing. 2016.

Valid-computing. Control Structures. www.valid-computing.com.
Luettavissa: <http://www.valid-computing.com/control-structures.html>. Luettu: 16.11.2017.

Virginia Tech, Department of Computer Science. The Spiral Model. Luettavissa: <http://courses.cs.vt.edu/~csonline/SE/Lessons/Spiral/index.html>. Luettu: 4.12.2017.

Walkenbach, J. 2015. Excel 2016 Bible. Walkenbach, John. John Wiley & Sons cop. 2016.

Walkenbach, J. 2013. Walkenbach, John. For Dummies. 2013.

World Finance 2013. Stock Portfolio. Luettavissa: <https://finance.mapsofworld.com/portfolio/investment/stock.html>. Luettu: 8.11.2017.

Zacks. Performance Analysis for a Stock Portfolio. Luettavissa: <https://finance.zacks.com/performance-analysis-stock-portfolio-5133.html>. Luettu: 8.11.2017.

Liitteet

Analyysityökalun lopullinen käyttöliittymä

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	R	S	T	U
1	Osake	Määrä	Viimeisin hinta	Hinta ostohetkellä	Hinnan muutos ostohetkestä	Viimeisin P/E	P/E ostohetkellä	P/E muutos ostohetkestä	Voiliteetti 250pv	Arvo	Painoarvo	Historiallinen kokonaismuutos-% 1 vuosi							
2	CAVERION OVI	500	6,21 €	6,52 €	-4,75 %	-24,84	-24,80	-0,04	27,06	3105,00	24,54 %	-22,53 %							
3	FORTUM CORPORATION	180	17,18 €	15,00 €	14,50 %	30,67	30,66	0,01	19,26	3091,50	24,43 %	15,67 %							
4	ARETTI OVI	200	15,00 €	14,30 €	4,90 %	78,95	78,95	0,00	15,37	3000,00	23,71 %	10,95 %							
5	VALMET OVI	200	17,29 €	17,90 €	-0,06 %	31,44	31,44	0,00	24,18	3458,00	27,33 %	23,55 %							
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			
21																			
22																			
23																			
24																			
25																			
26																			
27																			
28																			
29										12654,50	100 %	7,33 %	Korjattu osakkeiden painoarvoilla						
30																			
31																			
32																			
33																			
34																			
35																			
36																			
37																			
38																			