

Tomi Saloranta

# Asfalttiaseman tietokannan synkronointi ja datan hyötykäyttö

Metropolia Ammattikorkeakoulu

YAMK

Automaatioteknologia

Opinnäytetyö

23.4.2015

Tekijä Otsikko	Tomi Saloranta Asfalttiaseman tietokannan synkronointi ja datan hyötykäyttö
Sivumäärä Aika	52 sivua + 1 liitettä 23.4.2015
Tutkinto	Insinööri (ylempi AMK)
Koulutusohjelma	Automaatioteknologia
Suuntautumisvaihtoehto	Automaatioteknologia
Ohjaaja(t)	Lehtori Jukka Pirinen Lehtori Markku Inkinen
<p>Amomatic-asfalttiasemat ovat panosperiaatteella toimivia siirrettäviä tuotantolaitoksia. Näitä asemia on vuosien saatossa toimitettu ympäri maailmaa ja vanhimmat toimivat asemat ovat yli 20 vuotta vanhoja. Uusimpiin asemiin on mukaan toimitettu nykyaikainen etäyhteysmahdollisuus, joka helpottaa aseman henkilökuntaa saamaan asiantuntevaa apua suoraan mobiiliverkon yli. Asemaa ohjataan Siemens PCS7-järjestelmällä ja hajautetulla logiikalla. Kaikki reseptit ja tuotannon hallinta (esim kuorma-autot) ja antureista tuleva data sekä laskennalliset tuotantodatat kerätään SQL-tietokantaan. Aseman omalla raporttityökalulla on mahdollista hakea ja tulostaa raportteja panos/päivä kohtaisesti. Tällä hetkellä aseman henkilökunta tekee raportit asemalla ja informoi päälliköitä tuotantoluvuista ja kulutetuista raaka-aineista.</p> <p>Opinnäytetyön tarkoituksena on tutkia ja mahdollistaa tietokannan siirto erilliselle palvelimelle, etsiä ratkaisuja pilvipalvelimen ja asiakasportaalien tekemistä varten. Tarkoituksena olisi, että aseman henkilökunta ja päälliköt voisivat katsoa aseman tuotantomääriä internetistä suoraan tai mobiiliohjelmasta käsin. Lisäksi tutkitaan AMOCEMS-pakokaasuanalysaattorin liittämistä omaan palvelimeen. Projekti on laaja ja dokumentista tulee osaksi asennusohje projektin vaiheista.</p>	
Avainsanat	SQL, synkronointi, CMS, DNN, Android

Author Title	Tomi Saloranta Asphalt Plant's database synchronization and data's utilization.
Number of Pages Date	52 pages + 1 appendices 23 May 2015
Degree	Master of Science in Engineering (Master's Degree Programme)
Degree Programme	Automation Technology
Specialisation option	Automation Technology
Instructor(s)	Jukka Pirinen, Lecturer Markku Inkinen, Lecturer
<p>Amomatic-asphalt plants are movable batch type production plants. Amomatic has sold these plants all over the world and the oldest working plants are over twenty years old. All the newest plants are sold with a modern remote connection system that helps the plant's crew to get remote support when they need. The plant is controlled with Siemens PCS7-system and Siemens distributed IO. All the asphalt recipes, production management information and values coming from the sensors are stored into the database. With the plant's computer it is possible to search and print reports from the database. Nowadays the plant's crew makes the reports and informs the managers about the production.</p> <p>This projects focuses on finding a way to synchronize the plant's database to Amomatic server and find a way to establish a cloud consumer portal. This could enable the possibility to examine the production figures from the cloud service or directly from the mobile application. There is also a need to connect AMOCEMS's exhaust fumes analyzer to the same server. This project is part of an extensive project and includes some installation guides.</p>	
Keywords	SQL, Synkronizing, CMS, DNN, Android

## Sisällys

1	Johdanto	1
1.1	Amomatic OY yrityksenä	2
1.2	Projektin ja ohjelmistojen vaatimuksia	5
1.3	AmoCEMS pakokaasuanalysaattori	7
2	Yhteyden muodostaminen asemalle	9
2.1	VPN-yhteyslaite Tosibox	10
2.2	Aseman tietokannan synkronointitapoja	13
2.3	Synkronointiohjelman valinta	15
2.4	Synkronoinnin automatisointi	17
3	SQL-relaatiotietokanta	18
3.1	Aseman SQL-tietokantaan tehtävät muutokset	19
3.2	Datan haku tietokannasta	23
4	Sisällönhallintajärjestelmä	25
4.1	Drupal	26
4.2	DNN	28
4.2.1	DNN asennus omalle palvelimelle	29
4.2.2	DNN Konfigurointi	29
4.2.3	Modulit ja datan esittäminen	31
5	Azure-pilvipalvelu	34
5.1	Palvelun ominaisuuksia	35
5.2	DNN Asennus Azureen	37
6	Android-sovellus	38
6.1	DNN Sharp	39
6.2	Basic 4 Android	40
6.3	Android ohjelman tekeminen	42
6.3.1	Ohjelma nyt ja suunnitelmat	45
7	Yhteenveto	49
8	Lähdeluettelo	51

## Lyhenteet

3G	Third Generation
AES	Advanced Encryption Standard
API	Application programming interface
APIKey	Application programming interface Key
BF-CBC	BlowFish-Cipher Block Chaining
BI	Business Intelligence
CMS	Content Management System
CSV	Comma-Separated Values
DHCP	Dynamic Host Configuration Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IIS	Internet Information Service
IP	Internet Protocol
JSON	JavaScript Object Notation
NAT	Network Address Translation
NDIR	Non-Dispersive Infrared
RAD	Rapid Application Development
RDBMS	Relational Database Management System
REST	Representational State Transfer
RSA	Rivest, Shamir ,Adleman
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
USB	Universal Serial Bus
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

# 1 Johdanto

Amomatic-asfalttiasemia on myyty jo usean vuosikymmenen ajan ja niitä on toimitettu useita Pohjoismaihin, Venäjälle ja muualle maailmaan. Matkapuhelinverkon ja mobiilidatan kehittymisen ansiosta on jokaiseen asemaan lisätty etäyhteyksmahdollisuus jo useita vuosia. Mobiililaitteiden ja tekniikan kehittyminen asettavat nykyisin paineita kehittää tekniikkaa niin, että tieto olisi yhä helpommin saatavissa silloin, kun sitä tarvitaan. Asiakkaat myös luottavat siihen, että vikatilanteen sattua apua on saatavissa internetin yli etäyhteydellä ja puhelimella.

Nykyään tietoa kerätään reaaliaikaisesti valtavia määriä, kuten esimerkiksi tuotantolaitoksista, suurista moottoreista ja prosesseista. Tiedon kerääminen on helpoin tapa päästä analysoimaan tehtaan tai moottorin toimintaa ja näin vaikuttaa sen toimintaan. Energiatehokkuus, tiukentuneet ympäristönormit ja tietysti tuotantokapasiteetin optimointi vaativat pohjaksi mitattua tietoa prosessista. Amomatic-asfalttiasemassa raaka-ainemäärät jokaisesta tuotetusta asfalttiannoksesta tallennetaan tietokantaan. Asfaltista otetaan aika ajoin useita laboratorionäytteitä ja verrataan asemalle tehtyyn reseptiin. Näin prosessia voidaan hienosäätää paikanpäällä.

Asfalttiasemalla tallentuu siis paljon tietoa tietokantaan, mutta ilman erikoisjärjestelyjä tietoa ei ole voinut katsella aseman ulkopuolelta. Tämän projektin tarkoituksena on suunnitella ja toteuttaa asfalttiaseman tuotantotietojen esitysportaali internettiin ja mobiiliohjelmisto. Suurena osuutena projektissa on tietojen synkronointi ensin omalle palvelimelle ja siitä pilvipalvelimeen. Datan tallentaminen palvelimelle antaa mahdollisuuden tarjota asiakkaille aivan uusia mahdollisuuksia nähdä tiedot graafisessa muodossa ”dashboardina” ja ajaa helposti raportteja etänä.

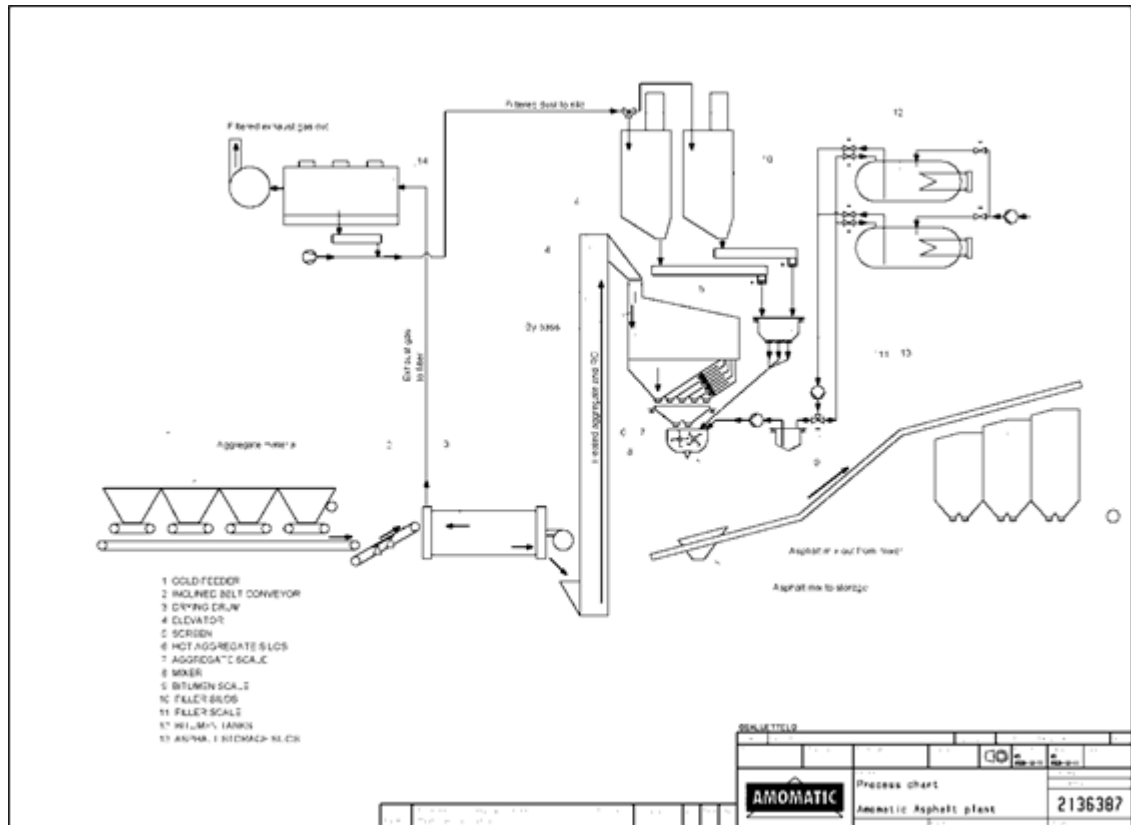
Suunnittelu on suoraviivaista, mutta usein projektin edetessä joudutaan keksimään vaihtoehtoisia ratkaisuja. Tässä projektissa suunnittelu ja tekeminen menevät jonkin verran päällekkäin johtuen monen eri ohjelmiston yhteensovittamisesta. Suunnittelun peruslähtökohtana on saada asemilta tarpeelliset tiedot siirrettyä relaatiotietokannasta omalle palvelimelle niin, ettei aseman tietokanta koskaan häiriinny. Paras mahdollinen lähtökohta olisi ettei aseman tietokantaan tarvitsisi tehdä lainkaan muutoksia.

## **1.1 Amomatic OY yrityksenä**

Yritys on Pohjoismaiden johtava asfalttiasemien valmistaja. Amomatic on perustettu vuonna 1919, joten sillä on pitkä kokemus korkealuokkaisten ja luotettavien asfalttiasemien valmistuksesta. Asemat ovat modulaarisia ja kehitetty vastaamaan asiakkaiden tarpeisiin. Luotettavuus, vähäinen huollon tarve ja kunnossapidon helppous ovat avainasemassa, kun asemalta halutaan pitkää ikää ja tuottavuutta. Tähän Amomaticilla on kehitetty useita erillaisia tapoja vastustamaan graniitin osia kuluttavaa vaikutusta. Näistä muutamia ovat muun muassa kulutusta kestävä materiaalin käyttö, vaihdettavat suojat ja kivi vasten kiveä-periaatteen hyödyntäminen esimerkiksi kananportaissa. Koviten kulutukselle joutuva asfalttimassan sekoitin uusitaan kulutusosiltaan normaalisti kerran vuodessa (100 000 - 150 000 tuotettua asfalttitonnia).

[1]

Amomatic-asfalttiasema toimii annosperiaatteella. Asfaltti sisältää kivimateriaaleja (hiekkaa, sora- ja/tai kalliomursketta), täytejauhetta (kivipöly, kalkkikivijauhe ja/tai lentotuhka), sideainetta (bitumi) sekä mahdollisesti joitain lisäaineita. Asfaltin valmistusprosessissa eri materiaaleja esikäsitellään sekoitusprosessiin sopivaksi, ja ne annostellaan ennaltamäärätyssä suhteessa sekoittimeen. Sekoitin valmistaa materiaaleista homogeenista asfalttimassaa, joka on sellaisenaan levitettävissä työkohteessa. Asfaltin valmistusprosessi Amomatic-asfalttiasemalla on esitetty alla olevassa kuvassa 1. [1]



Kuva 1: Asfaltin valmistusprosessi. [1]

Kylmäsyöttimeen tuodaan ensin eri siloihin asfalttireseptin mukaisia kiviaineksia. Kylmäsyötin suhteuttaa kiviainekset reseptin mukaan ja vinohihna tuo kiviainekset kuivausrumpuun. Pyörivässä kuivausrummussa kiviaines kuumennetaan n. 200-asteiseksi tehokkaalla kaasu-/kevytöljy-/raskasöljypolttimella. Kivistä erottuva kiviöly imetään suureen pussisuodattimeen, josta kiviölyä voidaan siirtää puhaltamalla/elevaattorilla fillerisiiloon uusiokäyttöä varten. Kivien kuumennuksen jälkeen ne nostetaan kuumaelevaattorilla seulaan, josta kivet seulotaan oikeisiin kokoihin omiin kuumasiiloihinsa. Samalla ylisuuret kivet tulevat ylivuotoputkesta ulos. Tämän jälkeen kiviä, bitumia, fillereitä, granulaattia punnitaan reseptissä annetut määrät ja tiputetaan sekoittimeen. Kun sekoitus on valmis, tyhjennetään panos joko suoraan kuorma-autoon tai vaunuun joka vie asfaltin massaradan siloihin odottamaan autoja. Sekoittimia on 3 - 5 tonnisia, joten panoksen koko on niiden välillä. Yhden panoksen tekemiseen menee noin yksi minuutti, joten esim 4000 kg panoskoolla tunnissa massaa saa tehtyä 240 tonnia. Tästä tulee Amomaticin asemien tyypit 160, 200, 240 ja 300. Taulukossa 1 on perus 240-aseman tiedot.



	standardimalli	optiot
Aseman tuotantoteho	240 tonnia tunnissa	
<b>Kylmäyöttölaite</b>		
Siilojen lukumäärä	4	vapaasti valittavissa
Yhden siilon tilavuus	12m <sup>3</sup>	9m <sup>3</sup> tai 15m <sup>3</sup>
<b>Kuivausrumpu</b>		
Halkaisija	2500 mm	
Pituus	9000 mm	
Polttimenteho	18,4 MW	
Polttoaine	kevytöljy	biokaasu, maakaasu, raskasöljy tai yhdistelmäpoltin
Rummun käyttö	4 x 18,5 kW	
Rummun pyörimisnopeus	säädettävissä taajuusmuuttajalla	
<b>Pölysuodatin</b>		
Suodatin pinta-ala	840 m <sup>2</sup>	960 m <sup>2</sup>
Savukaasupuhaltimen säätö	pyörimisnopeus säädettävissä tajuusmuuttajalla	
<b>Sekoitintorni</b>		
Lajikkeiden määrä	5	4 tai 6
Kuumasiilon tilavuus	25 m <sup>3</sup>	32 m <sup>3</sup> tai 50 m <sup>3</sup>
Sekoittimen annoskoko	4000 kg	
<b>Fillerilaitteet</b>		
Fillerisiilojen tilavuus	60 m <sup>3</sup> + 60 m <sup>3</sup>	esim. 4 x 60 m <sup>3</sup>
<b>Massasiilosto</b>		
Tilavuus	100 m <sup>3</sup>	30-300 m <sup>3</sup>
Siilojen lukumäärä	2	1-5
Siilojen poikkileikkaus	suorakulmainen	pyöreä tai suorakulmainen
<b>Granulaattilaitteet (optio)</b>		
Siilon tilavuus	3 m <sup>3</sup> tai 6 m <sup>3</sup>	
<b>Aminilaitteet (optio)</b>		
Säiliön tilavuus	1,5 m <sup>3</sup>	
<b>Bitumisäiliö (optio)</b>		
Säiliön tilavuus	50 m <sup>3</sup> , vaaka-asento	50-100 m <sup>3</sup> , vaaka- tai pystyasento
Säiliön lämmitys	sähkövastukset	
<b>Polttoainesäiliö (optio)</b>		
Säiliön tilavuus	20 m <sup>3</sup> , vaaka-asento	20-100 m <sup>3</sup> , vaaka- tai pystyasento
<b>RC-laitteet (optio)</b>		
		Kylmän kireätysasfaltin lisäys asfalttiaseman kuivausrumpuun tai erillinen kuumennusrumpu kierrätysasfaltille.

Taulukko 1: Amomatic 240 SM -aseman tiedot

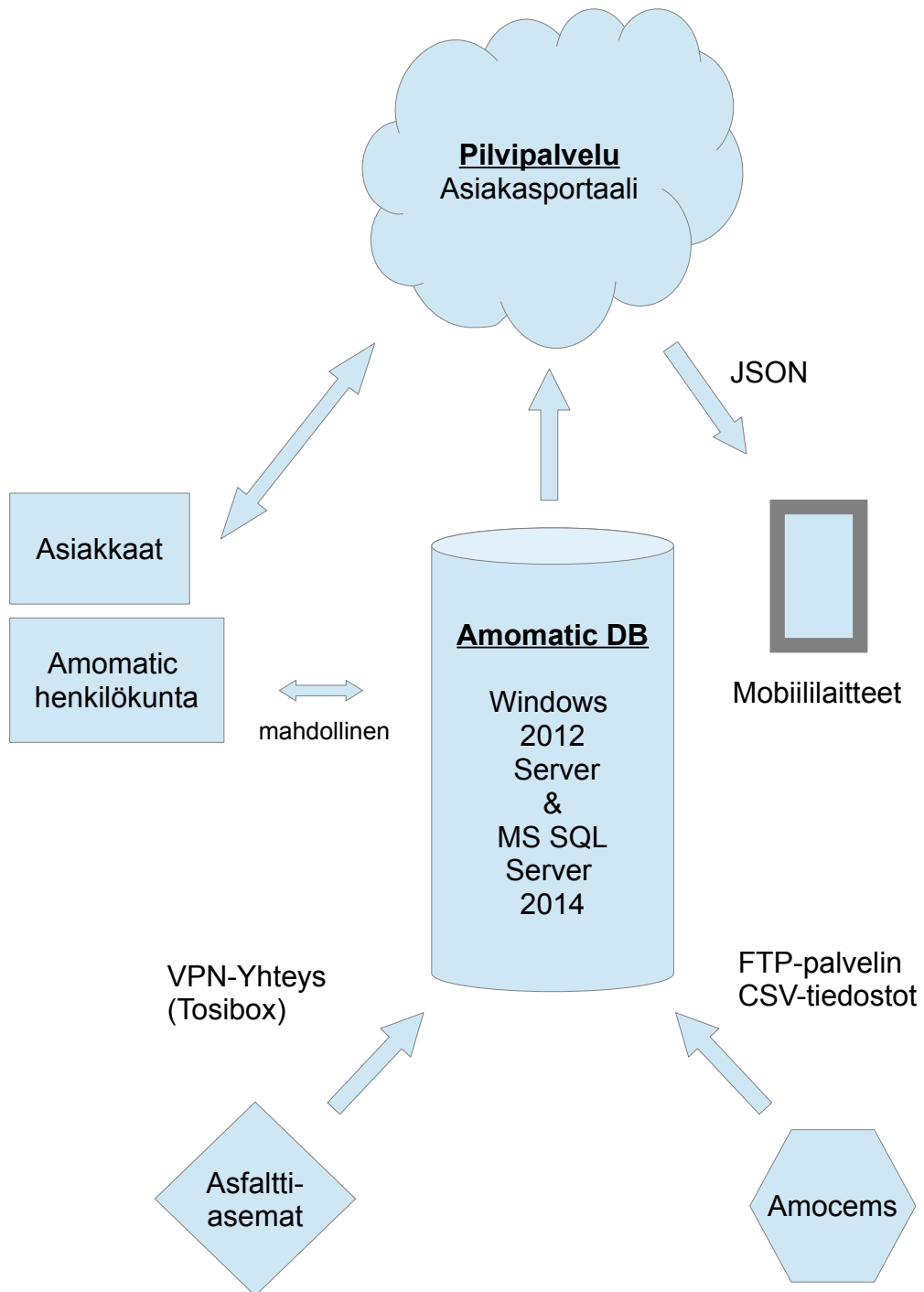
## **1.2 Projektin ja ohjelmistojen vaatimuksia**

Asfalttiasemalla on ollut jo kymmeniä vuosia Siemensin logiikka. Ohjausohjelmistona oli aikaisemmin käytössä Wonderware Intouch Windowsin eri käyttöjärjestelmillä, mutta vuonna 2006 on siirrytty Siemensin WinCC-ohjelmistoon. Nykyisin käytössä oleva Siemensin WinCC V8.1 Microsoft Windows 7 Ultimate käyttöjärjestelmällä, tuo mukanaan muun muassa Microsoft SQL Server-2008 R2, Microsoft Visual Studio , Microsoft IIS -ohjelmistot. Microsoft SQL palvelimen tietokannassa on WinCC:n tarvitsema data ja projektia varten tehty tietokanta, joka sisältää aseman anturitiedot ja tuotantodatan aina panoskohtaisesti. Asfalttiasema toimii siis panosperiaatteella, jolloin jokaisen tehdyn panoksen jälkeen anturitiedot ja punnittujen materiaalien tiedot tallennetaan tietokantaan pyydetyn asfalttireseptin kanssa. Näin nähdään mitä reseptissä on pyydetty ja mitä kone on tehnyt. Asfalttiaseman nykyiset ohjelmistot ja logiikat vaikuttavat projektissa oleellisesti valittaviin ratkaisuihin.

Tuotteiden valinnassa on vaihtoehtoina ilmaiset ja maksulliset ohjelmistot. Ohjelmistojen ilmaisuus kiinnostaa, mutta joskus se tuo mukanaan myös ongelmia, jos esimerkiksi Microsoftin SQL-palvelinohjelmisto vaihdettaisiin vastaavaan ilmaiseen ohjelmistoon. Helpointa on kuitenkin pitäytyä mahdollisimman paljon yhteensopivien tuotteiden parissa. Ohessa projektin havainnekuva (Kuva 2). Tarkoituksena on siis kerätä kaikki tarpeellinen tieto asemilta ja pakokaasuanalysaattoreilta (AmoCEMS) keskitettyyn omaan palvelimeen. Palvelimelta tieto voidaan lähettää pilvipalveluun, jossa DNN/Drupal toimii asiakasportaalina ja esittää tiedot graafisena esityksenä. Omassa palvelimessa voidaan käyttää ja testata samaa asiakasportaalia kuin pilvipalvelussakin ja näin muutokset voidaan testata etukäteen. Omaa matkapuhelinsovellusta varten pitää tutkia esimerkiksi JSON datatyypin käyttöä. Azure pilvipalvelussa on muun muassa olemassa myös sivuston kahdennus mahdollisuus, jolloin omalla palvelimella pidettävän asiakasportaalin kaksoisversio ei välttämättä ole enää kannattavaa.

Projektin vaatimuksia listattiin Amomaticin henkilökunnan ja asiakkailta tulleiden tietojen perusteella.

- Aseman normaali toiminta ei saa häiriintyä tietojen haun takia
- Tuotantotiedot nähtävissä internetissä
- Tuotantotietoa saatava omaan mobiiliohjelmaan
- Edullinen hinta
- Pilvipalvelun laajennettavuus



Kuva 2: Järjestelmän peruseriaate

Projektissa käytettäviin ohjelmistoihin vaikuttavat asemalla käytettävät Siemensin ja Microsoftin tuotteet sekä Tosibox. Tosibox toimii tällä hetkellä vain Windows ja Mac tietokoneissa, joten on järkevintä suunnitteluvaiheessa pysyä mahdollisimman paljon jo käytössä olevien valmistajien tuotteissa. Palvelimelle alustaksi valitaan siis Microsoftin Server 2012 R2, joka on uusin Microsoftin palvelinohjelmisto. Tämä tuo

mukanaan automaattisesti uusimman IIS 8.5:n, joka on erinomainen pohja julkaisujärjestelmälle eli CMS:lle. Microsoft Server 2012 R2 tuo mukanaan myös tukun palvelimelle tehtyjä ominaisuuksia, kuten erillaiset etäkäyttö ja palvelinsovellukset. Yrityksen sisäisessä intranet-käytössä sopisi vanhempikin palvelinversio, mutta uusin versio tuo paremmat mahdollisuudet käyttää uusia grafiikkaominaisuuksia datan esittelyyn. Tämä siis sillä edellytyksellä, että päädytään hyödyntämään Microsoftin Visual Studiota.

Tietokantapalvelimeksi on luontevinta valita saman valmistajan uusin versio eli Microsoft SQL-Server 2014 Standard. Vaihtoehtoina on huomattavasti kevyempi ilmainen Express-versio. Standard-versio on tähän sopiva, eikä se hidastu tai jää pieneksi datamäärän kasvaessa.[2] Ilmaisia relaatiotietokantoja on myös olemassa, mutta tässä kohtaa on järkevintä siirtää aseman data saman valmistajan tietokantaan. Jatkokäsittely on näin helpompaa ja aseman dataa voidaan varmuuskopioida palvelimelle ja joskus jopa palauttaa tarpeen vaatiessa. Pilvipalvelussa tietokanta voisi olla sama(Microsoft),ilmaisversio tai jokin muu.

Tarvitaan siis:

1. Palvelin-tietokone/virtuaalikone
2. Microsoft Server 2012 R2
3. Microsoft SQL-Server 2012/2014
4. Tosibox-avain ja ohjelmistot
5. Tiedonsiirto-ohjelma(Microsoft tai joku muu)
6. Esitysohjelmat(Drupal tai DNN).

### **1.3 AmoCEMS pakokaasuanalysointilaite**

AmoCems on Amomaticin tuotteistama pakokaasuanalysointilaite, jolla voidaan mitata asfalttiaseman pakokaasuista happea, hiilimonoksidi, hiilidioksidi, typpimonoksidi,typpidioksidi ja rikkidioksidi arvoja. Lisäksi tiedonkeruulaitteeseen voidaan liittää pakokaasun lämpötila-antureita ja erinäisiä aseman toimintaan liittyviä arvoja, jos tarvitaan. Mittauslaitteena toimii Thermo Scientificin 60i-malli. Koko mittauslinja on lämmitetty ja ennen mittausta savukaasusta poistetaan lika ja kosteus erillisellä kuivaimella ja suodattimilla. Kuvassa 3 näkyy mitattavat arvot ja tarkkuudet.

Pakokaasuanalysaattori on jatkuvatoiminen ja perustuu NDIR-tekniikkaan [1]. Tämä perustuu siihen, että kaikki kaasut absorboivat säteilyä tietyillä aallonpituuksilla ja absorption perusteella kaasujen pitoisuudet voidaan mitata [3].

Compound	O <sub>2</sub>	CO	CO <sub>2</sub>	NO	NO <sub>2</sub>	SO <sub>2</sub>
Min range	0-5 %	100 ppm	5 %	50 ppm	20 ppm	20 ppm
Full scale	0-25 %	2 500 ppm	25 %	2 000 ppm	500 ppm	10 000 ppm
Zero drift (24h)	< 0,1 %	< 1 ppm	< 0,1 %	< 1,2 ppm	< 1 ppm	< 0,5 ppm
Zero drift (7d)	< 0,1 %	< 3 ppm	< 0,5 %	< 5 ppm	< 3 ppm	< 3 ppm
Span drift (24h)	< 0,1 %	< 1 % of span	< 1 % of span	< 1 % of span	< 1 % of span	< 1 % of span
Span drift (7d)	< 0,1 %	< 1 % of span	< 1 % of span	< 1 % of span	< 1 % of span	< 1 % of span
Accuracy	+/- 0,1 % of span	+/- 2 % of span	+/- 2 % of span	+/- 2 % of span	+/- 2 % of span	+/- 2 % of span
Response time	45 sec	70 sec	70 sec	70 sec	70 sec	70 sec
Linearity	0,10 %	2 % of full scale or 5 % of measured value (whichever is smaller)				
Dimensions	2000 x 600 x 800 (H x W x D) in mm					
Operating temperature	15 – 25 °C					
Power requirements	220 – 240 VAC +/- 10%					
Outputs	7 analog outputs					
Inputs	10 digital inputs					

Kuva 3: AmoCEMS pakokaasuanalysaattorilla mitattavat arvot[1]

Tällä hetkellä mitattavat arvot kerätään Digin DR210-laitteeseen, josta ne lähetetään kolmannen osapuolen palvelimelle XML-muodossa, SOAP-protokollaa ja 3G-modeemia hyödyntäen. Arvoja voidaan tarkastella pilvipalvelusta, mutta käyttöliittymä ei ole aivan nykypäivää. DR210-laitteen ominaisuudet ovat kovin rajalliset, eikä pyynnöistä huolimatta saatu päivitysmahdollisuutta, joten vaihtoehtona on datankeruulaitteen vaihtaminen.

Datankeruulaite tulisi olla noin 6 - 8 analogiasäätuloa omaava laite, joka osaisi lähettää esimerkiksi 3G-modeemin avulla internetin yli tiedot FTP-palvelimelle. Windows Server 2012 R2 palvelimelle onnistuu FTP-palvelimen tekeminen helposti, kunhan sopiva tiedonkeruulaite löytyy. Yhdessä esimerkkilaitteessa DataTaker DT80 löytyy kaikki nämä ominaisuudet. Se tallentaa tiedot CSV-tiedostoformaattissa, jonka tallentaminen tietokantaan onnistuu pienellä vaivalla. Koska pakokaasuanalysaattorin tarve ei projektin tässä kohtaa ole akuutti, ei laitteistoa investoitu vaan jätetään tämä suunnittelu asteelle.

## 2 Yhteyden muodostaminen asemalle

Nykyisin jokaiselle uudelle asemalle asennetaan myös suomalainen Tosibox-etäyhteyslaite, jonka avulla asemalle voidaan ottaa suojattu VPN (Virtual Private Network) -yhteys milloin tahansa. Asemalle toimitettavassa kokonaisuudessa on Tosibox-lukko, 3G-modeemi, 3G-antenni ja ethernet kytkin liittämistä varten. Asetukset normaalia asfalttiaseman järjestelmää varten tehdään valmiiksi ja siihen sisältyy:

- Master-avaimen sarjoitus lukkoon
- Lukon IP-osoitteeksi 192.168.100.1 aliverkon peite 255.255.255.0
- DHCP-päälle ja alueeksi 192.168.100.150 eteenpäin
- Käytettävän 3G-verkon yhteysosoite (tämä muuttuu automaattiseksi)
- Admin salasanan vaihto.

Asfalttiaseman ja Amomaticin-palvelimen väliseen tiedonsiirtoon on olemassa monia vaihtoehtoja:

1. Asema lähettää datan internetin yli palvelimelle esimerkiksi FTP:llä, josta data ladataan SQL-tietokantaan. Tässä tavassa ongelmaksi muodostuvat häiriöiden hankala korjaus etänä aseman tietokoneella, lähettävän FTP-ohjelman toiminnallisuus ja lisenssi sekä näiden yhdistävä ohjelmointi. Dataa ei myöskään tule kuin vain ennalta määrättyinä ajankohtina, eikä sitä helposti pysty nopeuttamaan/hidastamaan. Hyviä puolia voidaan ajatella olevan nopeus ilman erillistä VPN-yhteyttä ja datan lähetys vain, kun uutta dataa on tullut.
2. Palvelin avaan yhteyden ja hakee tarvitsemansa datan aseman tietokannasta. Tämän tavan heikkouksia ovat VPN-yhteyden muodostamisen hitaus ja VPN-yhteyden hitaus. Kun yhteys on muodostettu voidaan käyttää yksinkertaisia tapoja tietokantojen synkronisointiin eikä aseman tietokoneeseen tarvita erillisiä lisäohjelmia. Yhteyden avaaminen ja synkronisoinnin aikaväli on helposti muutettavissa ja voidaan tehdä esimerkiksi ohjelman pyynnöstä. Yhteyden avaamiseen ja synkronisointiin tarvittavat ohjelmoinnit tehdään vain palvelimen päässä ja niitä on täten helppo muokata.
3. Asema lähettää datan pilvipalveluun, josta palvelin hakee datan. Tällainen tapa on useasti valmiina tietojenkeruulaitteissa. Ongelmana on vaikea datan keräys

palvelimelle sekä erittäin rajoittuneet asetukset. Ongelmaksi muodostuu myös turha viive, mikäli dataa haluttaisiin mahdollisimman reaaliaikaisesti. Lisäksi rasitteena tulevat mahdolliset ylläpitokulut laitteen pilvipalvelusta, sekä todella rajallinen laajennettavuus.

Näistä tavoista aiotaan käyttää palvelimelta päin tapahtuvaa liikennöintiä (nro.2) eli avataan VPN-yhteys ja siirretään dataa niin usein kuin tarvetta on. Amomaticin AmoCems-pakokaasuanalysaattori käyttää tällä hetkellä kolmannen osapuolen palvelua, johon tiedonkeruu lähettää dataa (tapa nro.3) ilman palvelimien välistä synkronisointia. Koska on järkevää integroida pakokaasuanalysaattori samaan järjestelmään, kannattaa vaihtaa analysaattorin tiedonkeruulaitteisto vastaamaan esimerkiksi tapaa nro. 1. Näin on mahdollista käyttää kahta erillaista tapaa datan siirtoon. Pilvipalveluun tieto siirretään suoraan omalta palvelimelta ja tehdään heti, kun aseman tiedonsiirto on toteutettu onnistuneesti.

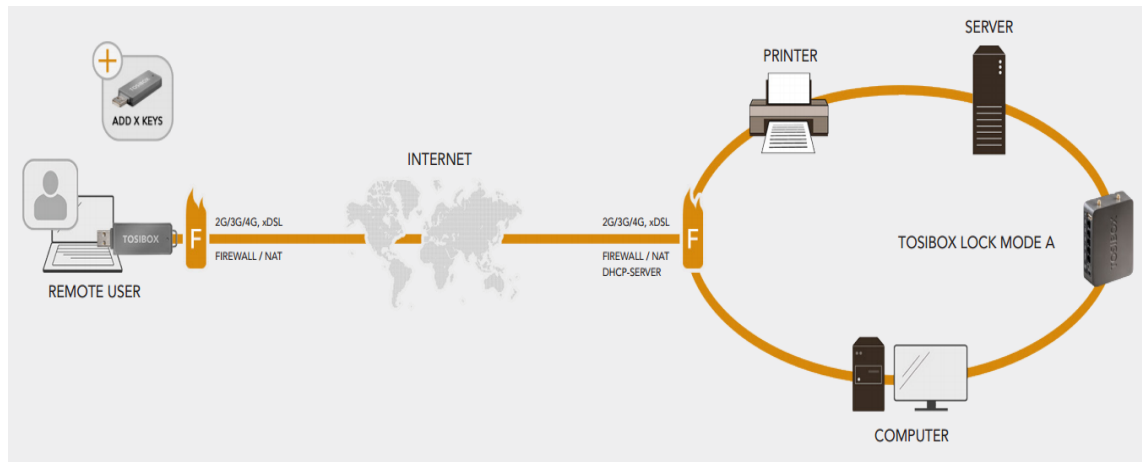
VPN-yhteyttä asemiin ei voida jättää auki, koska osoitevaruudet ovat jokaisella asemalla samat. Tähän Tosiboxilla on olemassa oma niin sanottu Central Lock, joka mahdollistaisi jatkuvat yhteydet joka asemaan. Tässä projektissa avataan kuitenkin jokaiselle asemalle yhteys ainoastaan tiedonsiirron ajaksi. Tosibox tarjoaa yhteyden muodostukseen Python-ohjelman, jolla yhteyden voi avata komentokehotteesta. Yhteyden avaukseen ja synkronisointiin tehdään powershell-scriptti, jota voidaan ajaa Windowssin omalla tehtävien ajoitus-ohjelmalla niin monta kertaa päivässä kuin on tarpeen.

## **2.1 VPN-yhteyslaite Tosibox**

Tosibox Oy on patentoinut maailman ensimmäisen Plug & Go etäyhteyslaitteen. Laitteiden myynti on alkanut Suomessa vuonna 2012. Tosibox Oy:n tuotteet valmistetaan Suomessa ja päätoimipaikkana toimii Oulu. Tosiboxia mainostetaan seuraavasti: "World's first plug & go remote access device." (Maailman ensimmäinen kytke ja käytä etäyhteyslaite). Tosibox-lukko ja avain sarjoitetaan kytkemällä avain lukon Universal Serial Bus (USB)-väylään noin kymmeneksi sekunniksi laitteen ollessa päällä. Tämän jälkeen USB-avaimella voidaan ottaa suojattu VPN-yhteys Tosibox lukkoon, kun molemmilla puolilla on internet yhteys. Avaimesta voidaan tehdä myös varmuuskopioita, jolloin jokaisella avaimella on samat oikeudet, tai vaihtoehtoisesti

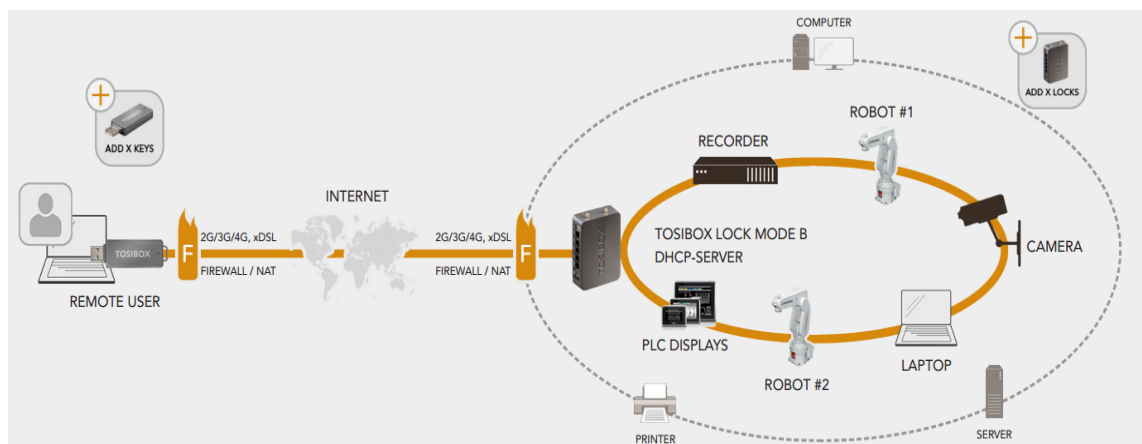
voidaan tehdä niin sanottu aliavain (Sub key), jolle voidaan asettaa oikeudet vain tiettyihin lukkoihin. [4]

Lukkoa voidaan käyttää A tai B-tilassa. A-tilassa (kuva 4) lukon LAN 3 -portti kytketään suoraan olemassa olevaan esimerkiksi yrityksen sisäverkkoon. Lukko hakee automaattisesti IP-osoitteen sisäverkon DHCP-palvelimelta. Palomuriin ei yleensä tarvitse tehdä muutoksia yhteyden muodostamiseksi.[4]



Kuva 4: Tosibox lukon käyttö A-tilassa [4]

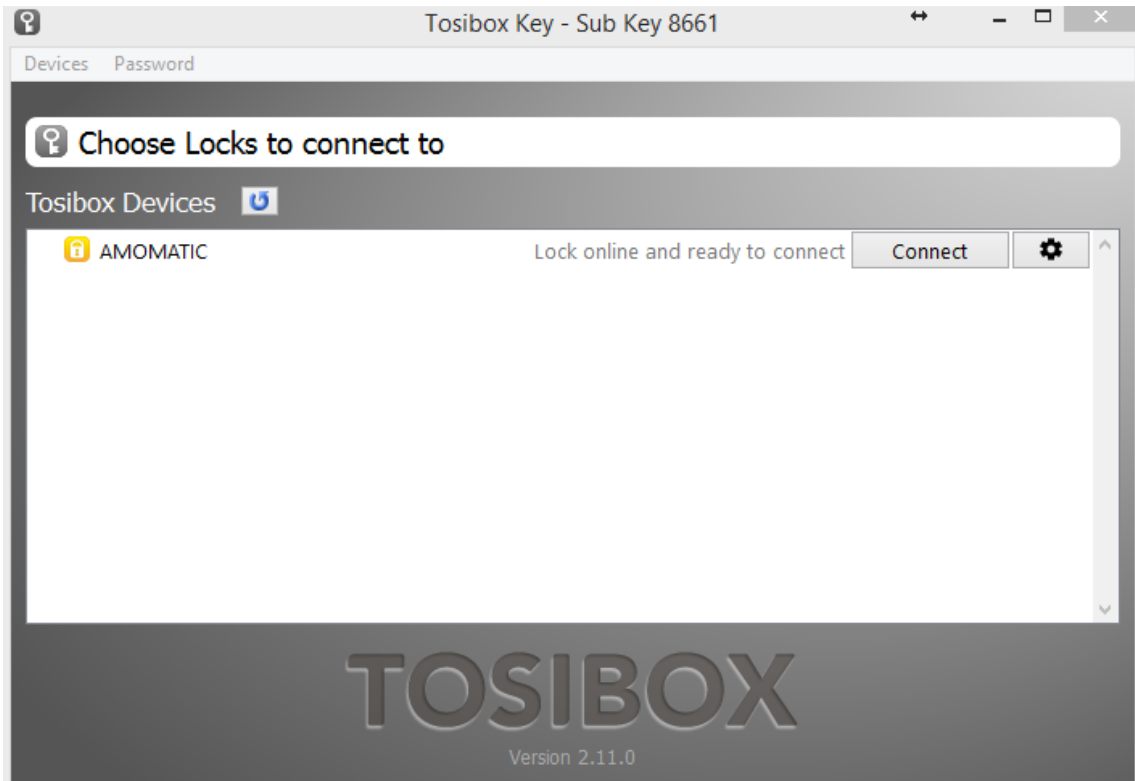
B-tilassa (kuva 5) olemassa oleva verkko kytketään WAN-porttiin tai käytetään 3G-modeemia internet yhteyden muodostamiseen. Tällöin lukko käyttää osoitteenmuunosta ja palomuria välissä, joten lukossa käytetään tällöin DHCP-palvelinta LAN-porteille ja erikseen kytkettäessä WLAN-laitteille. Tämä on usein asfalttiasemalla käytetty toimintatapa.[4]



Kuva 5: Tosibox lukon käyttö B-tilassa[4]



Tosibox-lukon asetusten määrittäminen onnistuu suoraan Service-portin kautta tai VPN-yhteyden yli. Tosibox-laitteeseen kuuluva Tosiboxin-avain toimii Mac- ja Windows-ympäristöissä. Avain pitää asentaa ensimmäistä kertaa, kun se kytketään tietokoneeseen ja tämän jälkeen yhteysikkuna avautuu aina, kun USB-avain kytketään tietokoneeseen. Yhteysikkuna näyttää ne lukot, jotka avaimeen on sarjoitettu. Keltaisella olevat lukot ovat yhteydessä internetiin ja harmaana olevat eivät kuten kuvassa 6. Yhteys voidaan muodostaa painamalla ”Connect”-nappia, jolloin lukon kuva muuttuu vihreäksi onnistuneen yhdistymisen tuloksena.



Kuva 6: Tosibox-Avaimen yhdistämispaneeli.

Tosibox perustuu OpenVPN-protokollaan, jonka VPN salauskyky on 6Mb/s (BF-CBC 128 bit). Vuonna 2015 julkistetussa uudessa Lock200:ssa VPN-salauskyky on 15Mb/s. Tosibox-lukko sisältää myös palomuurin, NATin, WLANin ja mahdollistaa kymmenen samanaikaista yhteyttä. Lukossa ei itsessään ole 3G-modeemia, vaan siinä käytetään USB 3G-modeemia. Tosiboxia voidaan käyttää myös mobiililaitteissa, joten se lisää entisestään sen joustavuutta.

Yhteyden muodostus voidaan tehdä myös käyttäen komentosarjakehoitetta. Tosibox toimittaa pyynnöstä Python-kielellä tehdyn ohjelman, jolla yhteyden avaus onnistuu suoraan komentokehoteesta tai rajapintaa voi käskyttää myös suoraan omalla ohjelmalla. Rajapinnan avaus vaatii muutoksen Windows rekisteriin sekä

komentorivikäskytystä varten tarvitaan palvelimelle Python 2.7.8 asennus. Windowsin rekisteriin tarvitaan muutos: HKEY\_CURRENT\_USER/Software/Tosibox Oy/ kohtaan new->"String Value", EnableControllInterface,1.[Virhe: Viitteen lähde ei löydy] Yhteys saadaan avattua tämän jälkeen komentokehotteessa: "python tosiboxcon.py connect-wait tb-vpnserver-\*\*\*\*\*"

Kun yhteys on muodostunut, aseman tietokone näkyy yhteysikkunassa vihreänä.

## 2.2 Aseman tietokannan synkronointitapoja

Microsoft SQL Serveriin saa lisäosina erillaisia synkronointitapoja, mutta on olemassa myös muita tapoja:

- **Transactional replication (MS SQL lisäosa)**

Tiedonsiirto aloitetaan yleensä tietokannan tallennuksesta. Heti kun alustava tietokannan tallennus on tehty, muutokset jotka tehtiin julkaisijan(publisher) koneeseen tehdään samassa järjestyksessä ja samoin ehdoin myös tilaajan (subscriber) koneeseen. Tätä käytetään yleensä palvelimien väliseen tiedonsiirtoon ja sopii muun muasa seuraaviin tapauksiin:

- Halutaan kopioida tiedot tilaajalle heti niiden tallennettua.
- Ohjelma vaatii nopean tiedonvälityksen.
- Julkaisija-palvelimessa on hyvin paljon insert,update ja delete toimintaa.
- Jompikumpi palvelimista on joku muu kuin MS SQL palvelin.[5]

- **Merge replication (MS SQL lisäosa)**

Tyypillisesti tässä tapauksessa aloitetaan myös tallentamalla tietokannan tiedot. Perättäiset datamuutokset ja struktuurimuutokset seurataan SQL-tietokannan triggereillä, kun tilaaja kytkeytyy verkkoon vain muuttunut data synkronisoidaan. Tätä tapaa käytetään usein palvelimen ja asiakkaan välillä ja esimerkiksi seuraavissa tilanteissa:

- Useat asiakkaat (subscribers) päivittävät samaa dataa, jolloin data kiertää palvelimen (publisher) kautta seuraaville.
- Asiakkaan on voitava tehdä muutoksia dataan verkon ulkopuolella ja synkronisoida jälkeinpäin.
- Tietojen sekaannusta voi tapahtua ja siihen on löydettävä korjaustoimet. [5]

- **Snapshot replication (MS SQL lisäosa)**

Tässä tapauksessa data toimitetaan juuri siinä muodossa, kuin se tietyllä hetkellä on ollut, eikä monitoroida muutoksia. Koko tallennettu tietomäärä välitetään sellaisenaan asiakkaille. Tämä tapa sopii esimerkiksi seuraavanlaisiin tilanteisiin:

- Tietokannan data ei muutu usein.
- Päivitysväli voi olla asiakkaan ja palvelimen välillä pidempi.
- Synkronisoidaan pieniä määriä dataa.
- Tietokannassa tapahtuu suuria muutoksia pienellä aikavälillä. [5]

- **Transaction Statements (Transact-SQL kuuluu vakiona)**

Tietokannan muutokset voidaan asettaa tehtäväksi niin, että ne toteutetaan vain, jos virhetilanteita ei tapahdu. Virhetilanteen sattuessa tietokannan muutokset perutaan. Tätä voitaisiin käyttää synkronisoinnissa siten, että kopioidaan taulusta kaikki rivit palvelimelle. "Begin Transaction"-komennolla aloitetaan ja jos muutokset onnistuvat ne hyväksytään komennolla "Commit Work". Jos virheitä ilmenee, muutokset perutaan (Rollback Transaction). Onnistuneen siirron jälkeen voitaisiin taulusta tyhjentää siirretyt rivit. Näin taulusta siirrettäisiin aina ainoastaan uudet rivit. [6]

- **Kolmannen osapuolen ohjelma, joka käyttää Transact-SQL tapaa synkronisointiin.**

Yhteenvetona tietokantojen siirtämiseen asfalttiasemalta on siis monia vaihtoehtoja:

1. SQL-tietokannan erikseen asennettaviin lisäosiin kuuluvat replikointi-työkalut, jotka mahdollistavat kaksisuuntaisen synkronisoinnin SQL palvelimien välillä: esimerkiksi Snapshot Replication.
2. Kolmannen osapuolen sovellukset: DB-Forge DataSync, EMS DB Comparer, SQL Maestro jne.
3. Manuaalisesti itse tehden käyttäen esimerkiksi: Transaction Statement (Transact-SQL).

Näistä ensimmäinen ei ole mahdollista, koska Siemensin toimittama SQL Server 2008 R2 Standard ei sisällä mitään lisäosia, eikä niitä saa asentaa.

Toinen tapa on tässä kohtaa kaikkein järkevin, koska projekti on melko laaja. Kolmannen osapuolen ohjelmisto sopii sellaisenaan minkä tahansa Microsoftin SQL-tietokannan (myös Microsoft SQL Express) ja Azure-pilvipalvelun tietojen siirtoon. Lisäksi valitsemalla sopiva kolmannen osapuolen ohjelmisto voidaan helposti tehdä synkronoinnista täysin automaattista.

Kolmas vaihtoehto on tulevaisuudessa mahdollinen toteuttaa niin, että virhekorjaukset huomioonotettuna voidaan tiedonsiirtotaulut kopioida ja tyhjentää aina tietojen siirron jälkeen. "Transaction" on työ, joka tehdään vain, jos kaikki datamuutokset osuuden aikana hyväksytään. Jos toiminnon aikana tapahtuu vikatilanteita, kaikki muutokset voidaan peruuttaa. Tämä tekisi periaatteessa saman, mitä kolmannen osapuolen ohjelmatkin tekevät. Haittapuolena on hidas muokattavuus ja suuri työmäärä, koska vikatilanteita voi olla monia.

Aseman tietokannasta kopioidaan siis aseman kaksi SQL-taulua (ReportDataSync ja ReportDataDetailsSync) Microsoft SQL palvelimeen käyttäen Tosibox etäyhteyslaitteen muodostamaa VPN-yhteyttä.

### **2.3 Synkronointiohjelman valinta**

Tietokannan synkronisointiin testattiin monia eri ohjelmia. Testilaitteistoon kuului:

- Täysin vastaava valvomo pc kuin asemalla (Siemens PCS7 IPC547C)
- Tosibox-etäyhteys 3G-yhteydellä
- Siemens logiikka CPU416-2
- Aseman käyttö simuloituna
- Kokeiltavat tiedonsiirtoohjelmat valikoitiin hinnan ja automatisointimahdollisuuden perusteella: SQL Maestro, EMS DB Comparer ja DB-Forge DataSync

SQL Maestro-yrityksen MS SQL Maestro ([www.sqlmaestro.com](http://www.sqlmaestro.com)) ei syystä tai toisesta löytänyt lainkaan testitietokannan kaikkia tauluja. Asiaa kysyttiin teknisestä tuesta, mutta syy jäi epäselväksi.

SQL Manager.Netin EMS DB Comparer ([www.sqlmanager.net](http://www.sqlmanager.net)) onnistui tekemään kaiken oikein, eli taulut saatiin synkronoitua sekä ohjelman visuaalisella puolella että komentokehotteesta. Synkronoinnin hitaus ihmetytti, koska noin 192 taulurivin synkronoinnissa ohjelma käytti 3G VPN-yhteydellä yli kaksi minuuttia. Tämä tuntui olevan aivan liian pitkä aika perustoimintoon, ja jos nopeampaa ohjelmaa ei olisi tullut vastaan, olisi pitänyt jo harkita muita tapoja.

Kokeilluista ohjelmista paras oli Devartin DB-Forge DataSync ([www.devart.com](http://www.devart.com)). Ohjelma toimii loogisesti ja jokaisesti synkronisoitavasta asemasta tehdään oma asetustiedosto. Ohjelman asetustiedostoa tehdessä valitaan tietokannat ja taulut mistä ja mihin kopioidaan. Lisäksi tehdään suodatusasetus niin, että voidaan valita montako päivää taaksepäin synkronoidaan: "timestamp > GETDATE()-20". Tämä vertailee timestamp-riviä taulussa ja ottaa tämän hetkisestä päivästä 20 päivää taaksepäin. Tästä johtuen ReportDataDetailsSync tauluun piti lisätä erillinen "timestamp"-sarake. Ideaalitulanteessa tuotantokaudella asfalttimassaa tehdään päivittäin. Näin ollen voidaan synkronisointi päivien määrää pienentää, jos asemalla on hyvä internet yhteys. Jos kuitenkin käy niin, että jostain syystä dataa jää jonkin päivän osalta pois, voidaan se korjata tekemällä koko kannan synkronisointi esimerkiksi kerran kuussa.

Kokeilluista ohjelmista parasta Devartin DB-Forge DataSynciä voidaan ajaa myös komentokehotteesta sen jälkeen, kun asemasta on ensin tehty asetustiedosto. Parametreina asetetaan asetustiedoston nimi ja lokitietostojen tyyppi ja paikka esimerkiksi:

```
"Datacompare.com /datacompare /compfile:"C:\Sync\asema".dcomp"
/sync /log:"C:\Sync\Logs\asema".log" /report:"C:\Sync\Logs\asema".html"
/reportformat:HTML"
```

Ohjelma tekee pyydettyä raportin synkronoinnin onnistumisesta tekstitiedostoon ja HTML-tiedostoon, kuten ylläolevassa parametreissa on määritetty. Ohjelma oli myös EMS DB Comparer-ohjelmaa huomattavasti nopeampi. Ohjelma käytti vain noin 25 sekuntia saman datamäärän siirtoon kuin EMS DB Comparer käytti yli kaksi minuuttia. Vertailun vuoksi kokeiltiin vielä ApexSQL:n Sql Server Data Compare-ohjelmaa, mutta ohjelma oli huomattavasti kalliimpi ja toimi yhtä nopeasti kuin Devartin DB Forge.

## 2.4 Synkronoinnin automatisointi

Windows ympäristössä ajastettu toiminto on helpointa tehdä käyttäen komentokehotetta. Scriptin eli komentosarjan voi tehdä monella eri ohjelmointikielellä, mutta Windowssissa yleisimmät ovat DOS6.22 tuttu batch ja uudempi Powershell.

Aluksi tehtiin normaali Dos6.22 batch-tiedosto eli sync.bat. Johtuen sen rajallisista ominaisuuksista uuden Windowssin kanssa, sitä ei saatu toimimaan toivotulla tavalla. Seuraava vaihtoehto oli tehdä sama Powershell-scriptinä. Ohessa toimiva synkronisointiskripti:

```
"$mac="tb-vpnserver-*****"
$asema="valvomo"

pushd c:\Sync\python tosiboxcon.py connect-wait "$mac"

if ($LASTEXITCODE -eq 0) { Write-Host "Connection is OK. Sync begins."

pushd "C:\Program Files\Devart\dbForge Data Compare for SQL Server\"

datacompare.com /datacompare /compfile:"C:\Sync\"$asema".dcomp" /sync
/log:"C:\Sync\Logs\"$asema".log" /report:"C:\Sync\Logs\"$asema".html"
/reportformat:HTML

pushd c:\Sync\python tosiboxcon.py disconnect "$mac"
exit
    }

else
    {
        Write-Host "Connection IS NOT OK! EXIT"
        exit
    }
}"
```

Ohjelma on tehty niin, että alkuun lisätään Tosibox VPN-yhteyden mac-osoite ja aseman nimi, jota on käytetty myös synkronointiohjelman kanssa. Kun ohjelma ajetaan, Tosibox yrittää avata VPN-yhteyttä asemalle. Jos yhteys muodostuu, aloitetaan tiedonsiirto ja, jos ei yhdistystä tapahdu, niin tulee virheilmoitus:"Connection IS NOT OK! EXIT". Ohjelma ajoitetaan Windowssin Task Schedulerilla toimimaan seuraavilla parametreilla:

```
"Program: C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe"
```

```
"Add arguments: -NoLogo -NonInteractive -File "C:\Sync\Valvomo.ps1""
```

Lisäksi pitää ajoittaa ohjelman käynnistys ajat. Valitaan ajankohdiksi klo 11 ja klo 16.

Näin scripti "Valvomo.ps1" saadaan ajettua ilman ylimääräisiä ikkunoita taustaprosessina.

Samalla synkronisointiohjelmalla voidaan tehdä synkronisointi myös pilvipalveluun, jossa on Microsoftin SQL-palvelin. Edellä olevaan scriptiin lisätään synkronointi oman palvelimen ja pilvipalvelun välille. Toiminto käynnistyy, jos aseman tiedonsiirto on onnistunut.

### **3 SQL-relaatiotietokanta**

Tietokanta on paljon enemmän kuin kanta, johon tallennetaan tietoa. Microsoftin SQL-Server on relaatiotietokanta tai oikeammin relaatiotietokannan hallintajärjestelmä (RDBMS). Tallennettava tieto tallentuu tietokantaan vain ennalta määrättyssä muodossa ja ennalta määrättyin ehdoin. Tietokanta ei ainoastaan tallenna tietoa, vaan hallinnoi sitä niin, että oikea tieto saadaan myös järjestelmään (tietokannasta ulos). Asiakasohjelmistot voivat hakea tietoa tietokannasta esimerkiksi verkon yli SQL-kielellä tehtyjen komentojen avulla. [7].

SQL Server 2014 on Microsoftin uusin pilvivalmis tietokanta ohjelmisto. Siitä yleisimmät perusversiot ovat Standard, Business Intelligence ja Enterprise sekä ilmainen Express. Näiden erot ovat lähinnä ominaisuuksissa, kuten siinä kuinka paljon muistia ne voivat käyttää ja tietokannan koossa. Kaikkien versioiden perusvertailutaulukko taulukossa 2. Express versio on kaikkein kevyin ilmainen tietokanta, joka on tarkoitettu pienille palvelimille ja pieneen käyttöön esimerkiksi tuotekehitykseen. Standard-versio on suunnattu pienten ja keskikokoisten yritysten tietokantoihin. SQL Server Business Intelligence-versio sisältää taas laajat BI-menetelmät, analysoinnin, Power Pivotin Sharepointia varten, raportointityökaluja ynnä muuta sellaista. Kaikkein laajin ja kallein Enterprise-versio on suunnattu suuria tietokeskuksia varten ja se tarjoaa suorituskykyä ja paremman tason saatavuutta tärkeille sovelluksille. Se mahdollistaa myös pilven eli servereiden jaetun käytön tietokantapalvelimen pohjana. [8]

Feature Name	Enterprise	Business Intelligence	Standard	Web	Express with Advanced Services	Express with Tools	Express
Maximum Compute Capacity Used by a Single Instance (SQL Server Database Engine) <sup>1</sup>	Operating System maximum	Limited to lesser of 4 Sockets or 16 cores	Limited to lesser of 4 Sockets or 16 cores	Limited to lesser of 4 Sockets or 16 cores	Limited to lesser of 1 Socket or 4 cores	Limited to lesser of 1 Socket or 4 cores	Limited to lesser of 1 Socket or 4 cores
Maximum Compute Capacity Used by a Single Instance (Analysis Services, Reporting Services) <sup>1</sup>	Operating system maximum	Operating system maximum	Limited to lesser of 4 Sockets or 16 cores	Limited to lesser of 4 Sockets or 16 cores	Limited to lesser of 1 Socket or 4 cores	Limited to lesser of 1 Socket or 4 cores	Limited to lesser of 1 Socket or 4 cores
Maximum memory utilized (per instance of SQL Server Database Engine)	Operating system maximum	128 GB	128 GB	64 GB	1 GB	1 GB	1 GB
Maximum memory utilized (per instance of Analysis Services)	Operating system maximum	Operating system maximum	64 GB	N/A	N/A	N/A	N/A
Maximum memory utilized (per instance of Reporting Services)	Operating system maximum	Operating system maximum	64 GB	64 GB	4 GB	N/A	N/A
Maximum relational Database size	524 PB	524 PB	524 PB	524 PB	10 GB	10 GB	10 GB

Taulukko 2: SQL palvelimen eri versiot ja rajoitteet.

[2]

### 3.1 Aseman SQL-tietokantaan tehtävät muutokset

Microsoftin SQL palvelimessa on paljon erilaisia muutoksenvalvontatyökaluja. Yleisimmät projektissa tarkastelun alla olevia ovat seuraavat:

- *Tracking changes* eli muutosten valvonta ominaisuus dokumentoi kaikki tietokannan tauluun kohdistuvat insert, update ja delete tapahtumat. Muutokset voidaan lukea ja katsoa ketä on käyttänyt dataa ja milloin.
- *Triggerit* eli voidaan esimerkiksi kopioida kaikki tiettyyn tauluun tulevat Insert-komennolla tallennettavat rivit toiseen tauluun.

Projektissa käytetään normaalia ”insert triggeriä”, koska ei ole tarpeen tietää, kuka ja milloin kopiointi on tehty.

Alunperin oli tarkoitus tehdä synkronointi täysin ilman aseman tietokoneen muutoksia. Tämä ei kuitenkaan ole mahdollista vaan aseman tietokoneen SQL-tietokantaan pitää tehdä muutoksia. Aseman tietokoneeseen tehdään kaksi uutta taulua eli



ReportDataSync ja ReportDataDetailsSync. Käyttäen "insert trigger" -komentoa tehdään kopionti niin, että alkuperäisistä tauluista kopioidaan lähes reaaliaikaisesti data näihin "Sync"-tauluihin. Näin voidaan synkronoida turvallisesti käyttäen "Sync"-tauluja eivätkä alkuperäiset taulut koskaan ole alttiina tiedonsiirrolle.

Ensimmäiseksi tehdään testilaitteiston tietokoneen tietokantaan muutokset. Lisätään kaksi uutta "Sync"-taulua, johon tiedot kopioidaan. Jokaisella taululla pitää olla yksi niin sanottu "primary key" eli tieto, joka on uniikki jokaisella rivillä. Taulujen tiedot myös lukitaan toisiinsa "foreign key" -avaimella eli ReportDataDetailsSync taulun ReportID on sama kuin ReportDataSync taulun ID. Näin koska jokaista ReportDataSync taulun riviä kohden ReportDataDetailsSync taulusta löytyy yhtä monta riviä kuin raaka-aineita on ollut käytössä. Huomiona lisäksi synkronisointiohjelmua varten lisätty Timestamp-rivi ReportDataDetailsSync-tauluun.

```

"CREATE TABLE dbo.ReportDataSync (
  ID bigint NOT NULL PRIMARY KEY,
  Timestamp datetime NOT NULL,
  RecipeName nvarchar(128) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL,
  Revision int NOT NULL,
  Customer nvarchar(128) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  Address nvarchar(128) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  Truck nvarchar(128) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
  Silo smallint NOT NULL,
  DoseError int NOT NULL,
  AsphaltTemperature real NOT NULL,
  BitumenTemperature real NOT NULL,
  Fuel real NOT NULL,
  AggregateDelay1 real NOT NULL,
  AggregateDelay2 real NOT NULL,
  FillerDelay real NOT NULL,
  BitumenDelay real NOT NULL,
  RecycleDelay real NOT NULL,
  FiberDelay real NOT NULL

```

)  
ON [PRIMARY]

```
CREATE TABLE dbo.ReportDataDetailsSync (
  ID int NOT NULL PRIMARY KEY,
  ReportDataID bigint NOT NULL FOREIGN KEY REFERENCES ReportData(ID),
  ComponentNumber int NOT NULL,
  ComponentName nvarchar(50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
  NULL,
  MeasuredValue real NOT NULL,
  RecipeValue real NOT NULL,
  MaterialTypeNumber smallint NOT NULL,
  TypeNumber smallint NOT NULL,
  Timestamp datetime DEFAULT (GETDATE())
)
ON [PRIMARY]"
```

"Create Table taulunnimi" -käskyllä pyydetään luomaan taulu, johon tulee seuraavat sarakkeet esim. "ID bigint NOT NULL PRIMARY KEY". Tämä tarkoittaa, että luodaan sarake "ID", jolle annetaan tallennusmääritelmä "bigint" eikä sarake saa jäädä tallennettaessa tyhjäksi (NOT NULL). Jokaisella oikein määritetyllä taululla on myös "primary key" eli määrite, joka on jokaisella rivillä uniikki. "Bigint" -määrite tarkoittaa, että tallennettava luku voi olla välillä:

$-2^{63}$  (-9,223,372,036,854,775,808) -  $2^{63}-1$  (9,223,372,036,854,775,807) [9].

Timestamp-rivi vaaditaan synkronointiohjelmia varten, koska muuten synkronoidaan aina koko datataulu. Johtuen 3G-yhteydestä ei ole järkevää synkronoida kuin esimerkiksi viimeiset kymmenen päivää, ja tämä voidaan tehdä asettamalla synkronointiohjelmiaan suodatus.

Seuraavaksi tehdään kaksi "insert triggeriä" eli aina kun tauluihin ReportData ja ReportDataDetails tauluihin tallennetaan tietoa, kopioidaan ne myös aiemmin tehtyihin ReportDataSync ja ReportDataDetailsSync-tauluihin. Esimerkkinä ReportDataDetails taulun repordettrig asetus:

```
"CREATE TRIGGER [dbo].[reportdettrig]
    ON ReportDataDetails
    AFTER INSERT AS
    BEGIN
    SET NOCOUNT ON;

    INSERT INTO
    ReportDataDetailsSync
    (
    ID,
    ReportDataID,
    ComponentNumber,
    ComponentName,
    MeasuredValue,
    RecipeValue,
    MaterialTypeNumber,
    TypeNumber )
    SELECT
    ID,
    ReportDataID,
    ComponentNumber,
    ComponentName,
    MeasuredValue,
    RecipeValue,
    MaterialTypeNumber,
    TypeNumber
    FROM INSERTED
    END
    GO"
```

Tietokantaan tehdään vielä käyttäjä SqlSync ja annetaan sille täydet oikeudet ReportDataSync ja ReportDataDetailsSync-tauluihin. Synkronointiohjelma ei vaadi täysiä oikeuksia, mutta tulevaisuuden mahdollisuuksia ajatellen on helpointa tehdä se nyt.

### 3.2 Datan haku tietokannasta

Jotta data saadaan diagrammeja varten sopivaan muotoon, pitää se ensin hakea sopivilla määritteillä tietokannasta. Siihen tarvitaan yhteysasetukset tietokantaan ja itse tietokantahaku, jossa data muokataan haulla sopivaksi. Riippuen siitä mitä tahdotaan näyttää, voidaan hakua rajata kuten tässä esimerkki tapauksessa. Ohessa oleva haku hakee viimeiset 30 tuotantopäivää ja laskee niistä päivittäisen tuotetun massan yhteen:

```
DECLARE @uusinDate datetime;
```

```
DECLARE @vanhinDate datetime;
```

```
SELECT @uusinDate = ( SELECT TOP 1 CONVERT(date, ReportData.Timestamp)
FROM ReportData ORDER BY Timestamp DESC );
```

```
SELECT @vanhinDate =
```

```
(
```

```
SELECT MIN(CurDate) FROM
```

```
(
```

```
SELECT TOP 30 CONVERT(date, ReportData.Timestamp) As CurDate FROM
ReportData
```

```
WHERE (ReportData.Timestamp >= @uusinDate-180)
```

```
GROUP BY CONVERT(date, ReportData.Timestamp)
```

```
ORDER BY CurDate DESC
```

```
) query_a
```

```
)
```

```
SELECT CONVERT(date, ReportData.Timestamp) AS CurDate,
ROUND(SUM(ReportDataDetails.MeasuredValue*0.001),0) AS TotalSumPerDay
```

```
FROM ReportData
```

```
JOIN ReportDataDetails
```

```
ON ReportData.ID = ReportDataDetails.ReportDataID
```

```
WHERE ReportData.Timestamp >= @vanhinDate
```

```
GROUP BY CONVERT(date, ReportData.Timestamp)
```

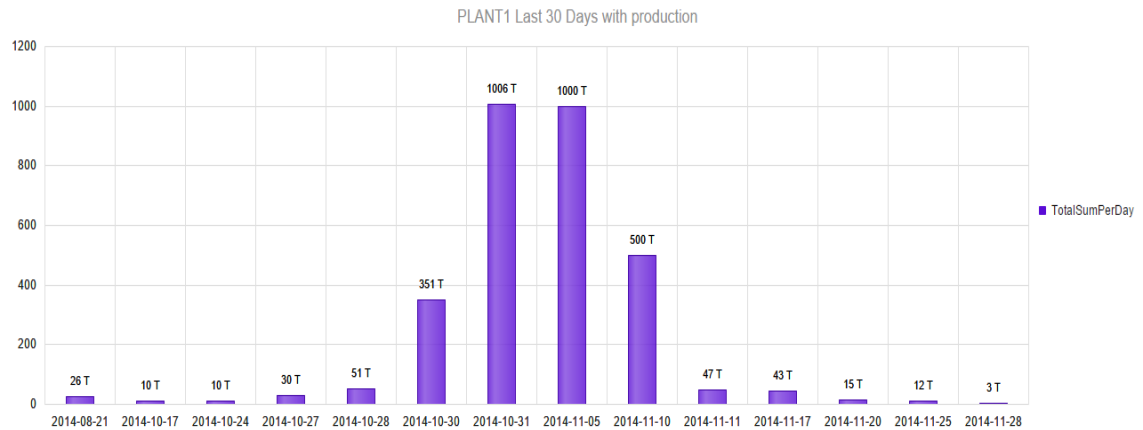
```
ORDER BY CurDate ASC”
```

SQL haun antamat tiedot taulukossa 3.

CurDate	TotalSumPerDay
21.8.2014	26
17.10.2014	10
24.10.2014	10
27.10.2014	30
28.10.2014	51
30.10.2014	351
31.10.2014	1006
5.11.2014	1000
10.11.2014	500
11.11.2014	47
17.11.2014	43
20.11.2014	15
25.11.2014	12
28.11.2014	3
23.1.2015	369
26.1.2015	190
27.1.2015	115
28.1.2015	125
2.2.2015	10
3.2.2015	5

*Taulukko 3: Haun antamat tiedot tietokannasta*

SQL-koodissa etsitään tietokannasta ensin uusin päivä, jolloin tuotantoa on ollut. Siitä valitaan 30 tuotantopäivää taaksepäin ja lasketaan niiden tuotantodatat päivittäin yhteen. Koodissa on myös ehto, ettei päiviä haeta 180 päivää kauempaa. Diagrammia varten tärkeimmät tiedot ovat: Curdate ja TotalSumPerDay. Tietokannassa painotkin ovat kiloina, joten siksi kerroin on 0.001, jotta saadaan tonneja(T). Kyseisen haun diagrammi on kuvassa 7 ja kuten kuvasta voi huomata, ei tuotantopäiviä ole ollut kolmeakymmentä viimeisen 180 päivän aikana. Tämä on selvää, koska data on testisimulaattorista.



Kuva 7: SQL-haun tuottama pylväsdiagrammi

## 4 Sisällönhallintajärjestelmä

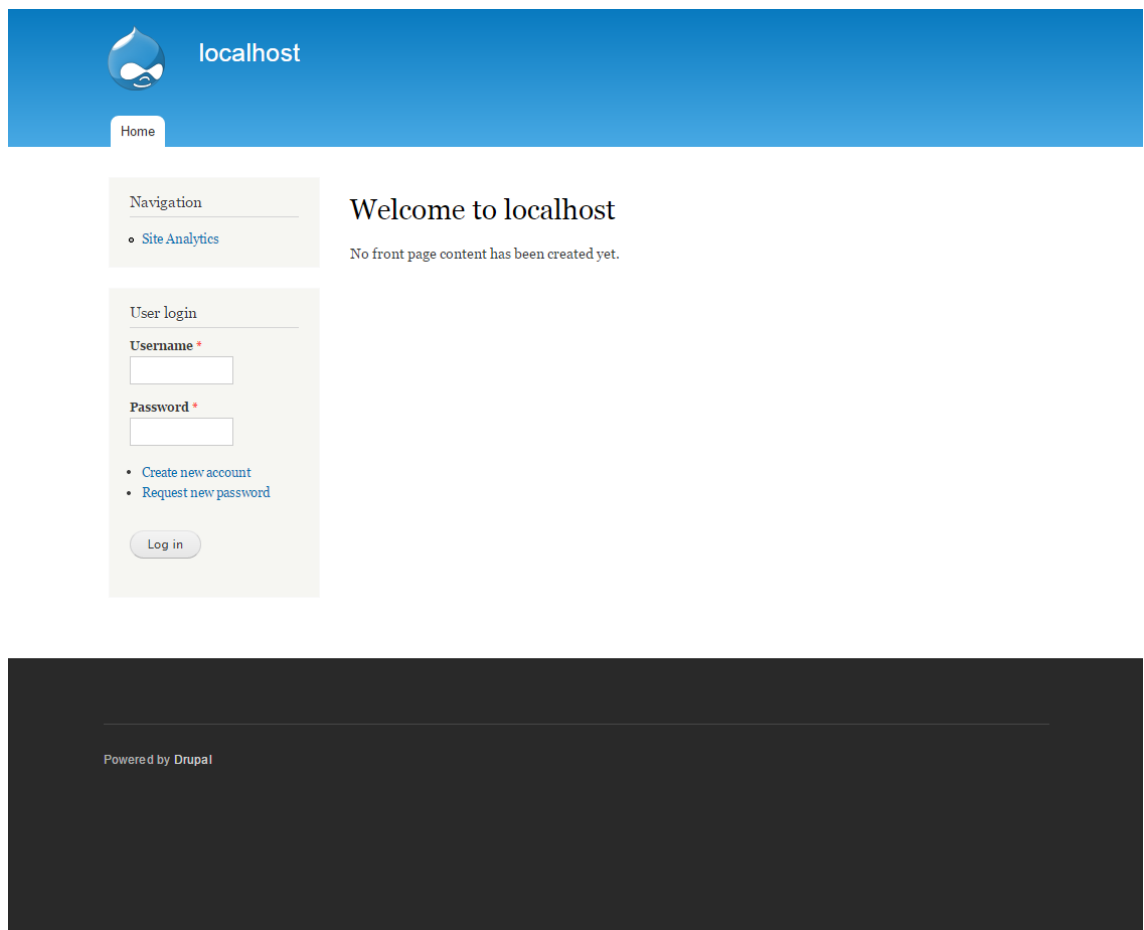
Sisällönhallintajärjestelmiä (CMS) on kehitetty useita kymmeniä viimeisten vuosien aikana internetiä varten. Osa näistä on tehty jotain tiettyä tarvetta varten ja osa on hyvinkin universaaleja ja mukautuvia. Nykyään suosituimmat CMS:t ovat Wordpress, Joomla! ja Drupal ja Microsoftin .Net pohjaisista DNN on suosituin. Yleisesti sisällönhallintajärjestelmiä on siis kymmeniä erillaisia. [10,11]

Julkaisujärjestelmän valinta tulee tapahtumaan tutkimalla näiden eri ominaisuuksia ja mahdollisuuksia. Projektin alkutietojen takia haasteena monessa julkaisujärjestelmässä on puutteellinen Microsoft-tuotteiden tuki, joka yleensä tarkoittaa myös edullisempaa hintaa. Suurennuslasin alla oli Wordpress, Drupal, Joomla, DNN ja RubyOnRails. Koska pilvipalvelin tulee olemaan osa Amomaticin sisällöntuotantoa, on luontevinta aloittaa Drupalilla. Drupal on PHP-pohjainen ja se voi käyttää sekä Microsoftin SQL että MySql-tietokantoja. Tämä ominaisuus tekee Drupalista edullisemmän pilvipalvelin sovelluksissa. Toisena vaihtoehtona pidetään DNN:ää.

Molemmissa vaihtoehdoissa on hyvät käyttäjien hallinta mahdollisuudet sekä laaja lisäosien kirjasto. Drupalin vahvuudet ovat riippumattomuus ohjelmistoalustasta ja laaja käytettävyys. Drupal on kuitenkin kovin vaikea konfiguroitava ja siitä onkin tulossa uudistunut 8-versio, josta puhutaan helpompina omaksua. DNN eli entinen DotNetNuke on täysin Microsoftin alustalle tehty eli toimii vain Microsoftin SQL-tietokannan ja IIS:n kanssa.

## 4.1 Drupal

Drupal on syntynyt alunperin Dries Buytaertin ja muiden yliopisto-opiskelijoiden lähiverkko yhteydenpitoon tehdystä Message Boardista. Vuonna 2000-2001 Message Board sai paljon huomiota osakseen ja siihen alettiin yhteisön voimin kehittämään lisäominaisuuksia. Näin Drupal syntyi ja siitä tuli avoimeen lähdekoodiin perustuva sisällönhallintajärjestelmä, jonka sisältö on näytettävissä eri käyttäjille eri muodossa. Drupalia voidaan laajentaa perusominaisuuksien lisäksi lisäosien avulla. [12] Drupal on PHP-pohjainen ja käyttää pohjana MySQL-relaatiotietokantaa. Näin ollen se on täysin riippumaton Microsoftin tuotteista ja perusosiltaan ilmainen. Kuvassa 8 on Drupalin aloitusnäky. Drupal sopisi projektiin, mutta Microsoftin SQL-palvelin saattaa tuottaa ongelmia.



Kuva 8: Drupal-sisällönhallintajärjestelmä

Drupal asennetaan aluksi testitietokoneeseen IIS:n ja Microsoftin SQL Server 2014:n kanssa. Drupal voidaan asentaa käyttäen Microsoftin Web Platform Installeria, mutta silloin on valittavana vain Acquia ja Commerce Kickstart-versiot. Asennusohjelma suostuu käyttämään vain ilmaista SQL Server Expressiä eikä löytänyt aiemmin asennettua Microsoft SQL Serveriä. Tämän takia Drupal asennetaan käyttäen perus pakettia, joka löytyy osoitteesta: <https://www.drupal.org/>. Microsoftin tarjoaman asennusohjelman avulla kannattaa kuitenkin asentaa Drupalia varten PHP5.3 ohjelmointikirjasto, URL-rewrite 2.0 ja PHP Manager for IIS.

Drupal paketti puretaan IIS 8.5 hakemistoon /inetpub/wwwroot/. Muut tiedostot voidaan hakemistosta poistaa ennen purkamista. Alikansiot ja niiden sisältämät tiedostot pitää jättää. Projektissa käytettävä kombinaatio Windows, IIS ja Microsoftin SQL Server vaativat monia lisäosia ja alkuvalmisteluja:

1. URLRewrite module
2. PHP SQL Server Driver 2.0
3. Drupal 7 driver for SQL Server
  - Tämä puretaan sites/all/modules/sqldr ja sisältä löytyvä sqldr kopioidaan lisäksi kansioon includes/database
  - Ajuri pitää laittaa päälle eli enabled-tilaan IIS Managerista löytyvällä PHP manager extensions eli lisäosat kohdasta. Samalla kannattaa valita käytettäväksi PHP versioksi 5.3.2.
4. Asettaa IUSR oikeudet kansioon inetpub\wwwroot\sites\default\files
5. Kopioida default.settings.php -> settings.php ja asettaa se muokattavaksi asennuksen ajaksi

Näiden jälkeen voidaan käynnistää itse Drupal asennus selaimella osoitteesta: /localhost/. Jos kaikki on asetettu oikein, asennus ei herjaa puuttuvista oikeuksista ja Microsoft SQL Server valinta löytyy ja asennusta voidaan jatkaa.

Drupaliin on saatavana paljon erilaisia moduleja, jotka lisäävät Drupalin toiminnallisuutta ja ominaisuuksia. Ohessa lista projektissa tarvittavista moduleista: Charts (kaaviomoduli), Ctools, Views, Schema ja Data. Drupalin asentaminen jo pelkästään on todella haastavaa, kun halutaan käyttää pohjana yhdistelmää IIS ja Microsoft SQL Server. Erillaiset modulit, visualisointi paketit ja käyttäjien hallinta toimivat hienosti. Nämä ominaisuudet olisivat projektissa tarpeen. Eri käyttäjille on mahdollista asettaa näkymiä (views), joten eri asemien data diagrammit voitaisiin



esittää näkymien avulla. Valitettavasti sqlsrv:n ajuri ei tue schema- ja datamoduleita, joita tarvittaisiin charts-kaavion datan tuomiseen Microsoftin SQL-tietokannasta. Myöskin import\_SQL-moduli alkoi antamaan virheilmoituksia sqlsrv-ajurista.

Koska erillaisia diagrammeja tarvittaisiin ja muokattavuuden pitäisi olla helppoa, näyttää siltä, että on keksittävä kiertotie. Drupal7 ja Mysql olisi pilvipalvelussa mahdollinen kombinaatio. Projektin aikataulun takia vaihtoehdoksi jää DNN.

## **4.2 DNN**

DNN Corporationin (entinen DotNetNuke Corp.) omistama DNN projekti, josta ovat eriytyneet kaupalliset versiot esimerkiksi Evoq ja ilmainen yhteisön ylläpitämä versio. DNN perustuu vapaaseen lähdekoodiin, joka tarkoittaa sitä, että se on vapaasti ladattavana internetistä ilman käyttöoikeusmaksuja ja jokaisella käyttäjällä on mahdollisuus muokata sen lähdekoodia. Shaun Walker kehitti DNN:n alunperin IBuySpy portaalin laajennusosaksi. IBuySpy taas oli Microsoftin kehittämä esimerkkisovellus .NET ympäristölle. DotNetNukun tuotekehitys lähti kunnolla käyntiin v.2004. [13]

DNN on kehittynyt viimeisen kymmenen vuoden aikana paljon. Viimeisin versio on tällä hetkellä v.2015 07.04.00 [14]. Alusta asti DNN on saavuttanut suosiota käyttäjäystävällisyydellään ja monikäyttöisyydellään. Avainominaisuuksina onkin pidetty asennuksen ja ylläpidon helppoutta, laajennettavuutta ja suorituskykyä. Avoin lähdekoodi ja yhteisön kehittämät ominaisuudet lisäävät DNN:n käyttömahdollisuuksia ja pitävät sen yhtenä suosituimpana sisällönhallintajärjestelmänä.

DNN perustuu täysin Microsoftin .NET arkkitehtuurille ja palvelimenaan käyttää sekä Microsoftin IIS-palvelinta että Microsoftin tietokantapalvelinta. Ominaisuuksia on paljon hallintaan, ulkonäköön ja laajennettavuuteen liittyviä valintoja. Turvallisuuden puolesta käyttäjähallinta jakautuu rooleihin ja käyttäjiin. Näin on helpointa lisätä tai poistaa käyttäjiä roolien alta.

Kun data on saatu siirrettyä omalle tietokantapalvelimelle voidaan dataa jo katsella SQL Managerin ikkunasta ”raakadatana”. Tässä muodossa tiedosta on vaikea hahmottaa tuotantotietoja esimerkiksi päivätasolla. Siksi tallennettu data on saatava graafiseen muotoon ja DNN on tähän hyvä pohja. DNN voi olla omalla palvelimella tai

pilvipalvelussa. DNN:lle on monia erillaisia moduleita, joilla dataa voi esittää. DNN:ää asennettiin ja testattiin aluksi omalla palvelimella, mutta lopulta tehtiin uusi asennus Azureen.

### 4.2.1 DNN asennus omalle palvelimelle

Dnn asennettiin ensiksi omalle palvelimelle testikäyttöä varten. DNN Corporation tarjoaa maksullista DNN versiota osoitteessa: <http://www.dnnsoftware.com/>. Maksuttoman versionkin saa, mutta sekin on linkitetty Microsoftin Web Platform-asennukseen. Tämä asennusohjelma ei suostu huomioimaan jo asennettua Microsoftin SQL-palvelinta, joten asennus on tehtävä manuaalisesti. DNN yhteisön asennuspaketin saa onneksi osoitteesta <https://dotnetnuke.codeplex.com/>.

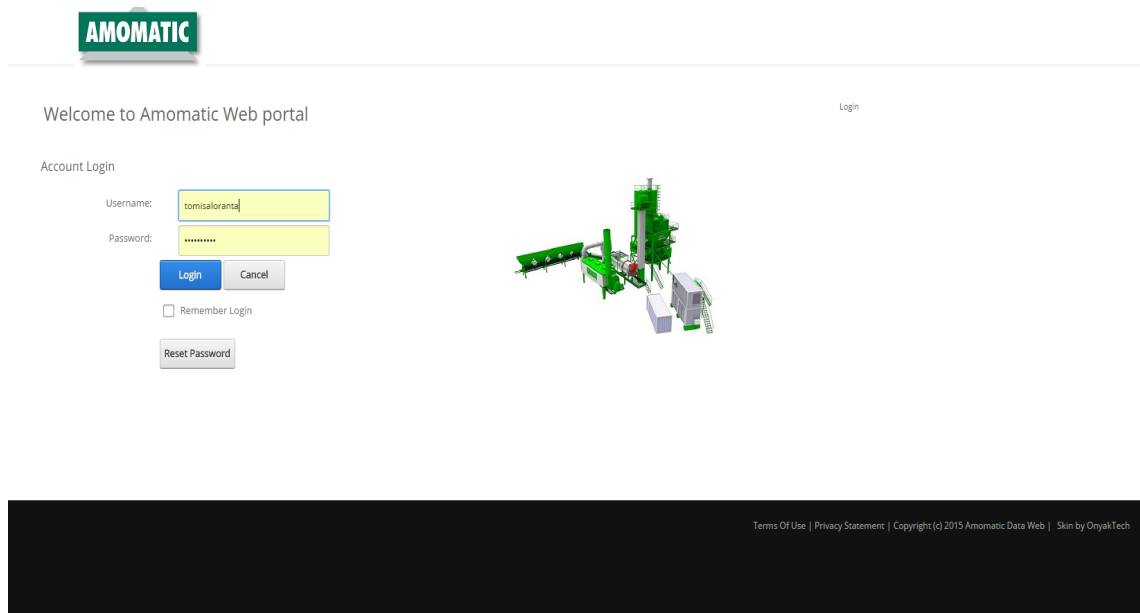
DNN asennus omalle palvelimelle testausta varten:

- Paketti puretaan esim. wwwroot/DNN ja dnn kansiolle annetaan vähintään muutosoikeudet "iis apppool\DNN"-käyttäjälle.
- Tehdään uusi web-sivu IIS Managerilla ja linkitetään se kansioon.
- Microsoft SQL Serverille tehdään Dnn-tietokanta ja tehdään käyttäjä esimerkiksi DnnAdmin, jolle annetaan dbadmin oikeudet ko. tietokantaan.
- DNN asennus voidaan tämän jälkeen aloittaa avaamalla internet selain localhost -osoitteella.
- Asennus kysyy admin salasanaa ja SQL-tietokannan tiedot.

DNN:n asennus sujuu paljon suoraviivaisemmin kuin Drupalin, vaikka lisätöitä tekeekin Microsoftin SQL Server Standard. SQL Express olisi asentunut ilman mitään suurempia ongelmia todennäköisesti suoraan Microsoftin Web Platform -asennusohjelmalla.

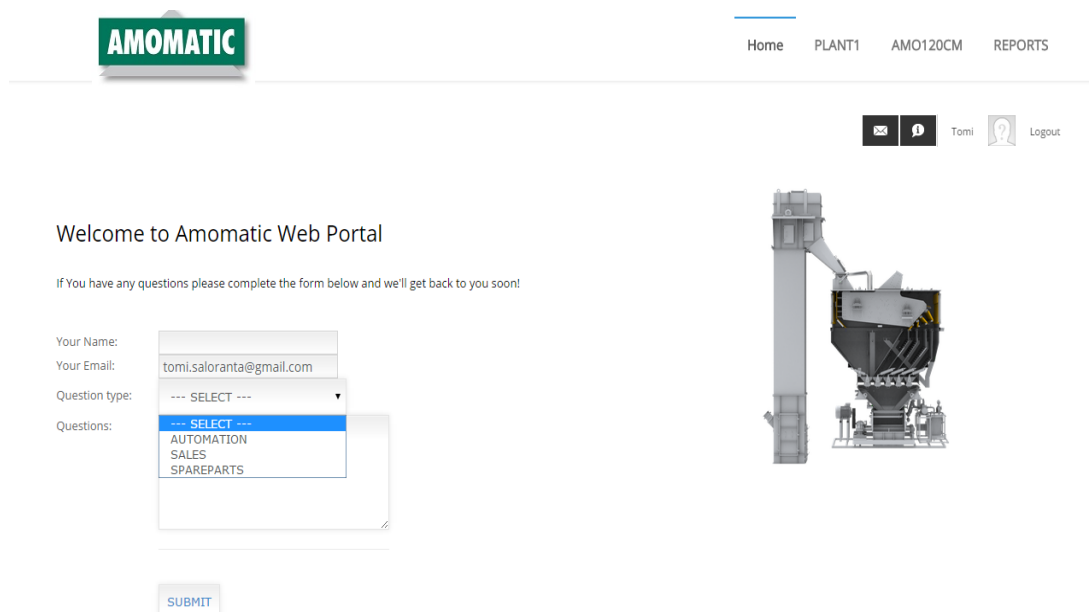
### 4.2.2 DNN Konfigurointi

Perusasennuksen jälkeen voidaan alkaa muokkamaan ulkonäköä. Ensimmäiseksi asennetaan Onyaktechin Junjii-grafiikkapaketti, joka on perus valkoinen ja pelkistetty ulkonäköllisesti. Tämä sopii hyvin laadukkaaseen business-tyyliseen sivustoon ja värejä tuovat myöhemmin erillaiset datadiagrammit. Kuvassa 9 on Amomaticin alkusivu, josta päästään kirjautumalla eteenpäin.



Kuva 9: Amomatic sivuston etusivu

Kun sivustoon kirjaututaan sisälle nähdään ensiksi palautelomake, koska tämä sivu on kaikille yhteinen ja siihen voidaan jakaa yleistä informaatiota ja/tai kuvia asemasta ja toiminnasta. Palautelomake tehtiin Onyaktech Forms-modulilla ja toimii niin, että alavetovalikosta voidaan valita mikä on yhteyspyynnön tyyppi (automaatio, varaosat vai myynti, Kuva 10). Ylhäältä oikealta klikkaamalla näkee oman aseman tuotantotiedot.



Kuva 10: Amomatic portaalin etusivu

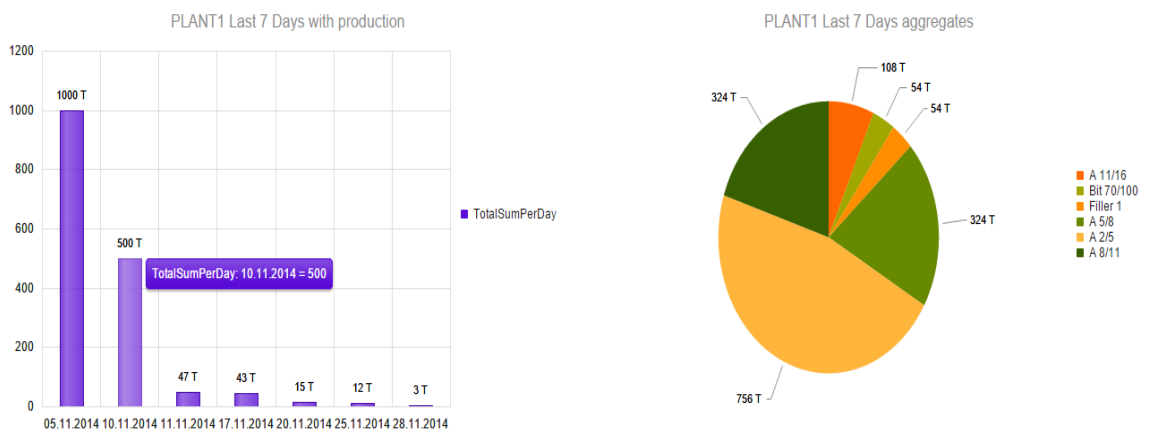
### 4.2.3 Moduilit ja datan esittäminen

DNN-alustalle on olemassa tuhansia moduleita, joista osa on ilmaisia ja osa maksullisia. Suurin osa moduleista on ostettavissa suoraan kaupasta osoitteesta:

<<http://store.dnnsoftware.com/>>

Yritys Onyaktechin päätavoitteena on tehdä joustavia yritysmoduleita DNN-sivustoille, jotka täyttävät asiakkaiden tarpeet. Yritys on tehnyt moduleja jo aivan DotNetNuken versiosta 1.1 asti. Yritys on perustettu vuonna 2004 ja modulien määrä on erittäin kattava. Moduleja on muun muassa Charts, Forms, Data Viewer, Axon Mail, Omnia, File Share Pro, Junjii, Fast Image. Moduilit voi ostaa kaikki tai erikseen ja lähdekoodin kanssa tai ilman. Asiakastuki toimii joko sähköpostin tai erillisen tukisivuston kautta. [15]

Lähes kaikki moduilit, jotka löytyivät ovat maksullisia. Onyaktech ei tässä suhteessa ole poikkeus, mutta tarjouksessa saa nelisenkymmentä modulia hyvin sopivaan hintaan. Näistä yksi on Onyaktech Charts-moduli, jolla saa tehtyä pienellä vaivalla suoraan tietokannasta näyttäviä grafikoita. Yhtenä peruseriaatteena alusta asti oli, että datan esilletuominen pitää olla nykyaikaisen näköinen ja mielellään interaktiivinen hiiren klikkauksille. Tästä kaikki ominaisuudet löytyivät. Kuvassa 11 on esimerkkinä pylväs- ja piirakka diagrammi. Hiiren osuessa palkin päälle palkin tiedot tulevat tarkemmin esille.



Kuva 11: Onyaktech modulin graafeja

Diagrammeja on monia erilaisia kuten pylväs-, piiras-, donitsi-, viiva- ja tutkadiagrammeja. Dataa voidaan myös yhdistellä monin erilaisin tavoin. Valintoja voi

tehdä myös tooltip-asetuksiin eli hiiren osuessa graafin päälle tuleviin teksteihin. Kuvassa 12 näkyy Charts-modulin pylväsdiagrammin asetuksia.

Chart Details

Chart Name:

Author:

Config Connections:   Use Selected Connection

Connection String:   Use Existing Database Connection

SQL Statement: 

```
DECLARE @uusinDate datetime;
DECLARE @vanhinDate datetime;

SELECT @uusinDate = ( SELECT TOP 1 CONVERT(date,
ReportData.Timestamp) FROM ReportData ORDER BY Timestamp
DESC );
```

\*\* When working with dynamic tokens don't include bracket identifiers "[ ] " for your fields, this may cause the parser to consider your field identifiers as tokens. Also keep the single quotes from your SQL. You can use the DEBUG feature to check for this. \*\*

Legend Options:  Display Legend

Background Color:   Transparent Background

Chart Size: Height:  Width:  (Set to 0 for dynamic chart size)

Tooltip Format:

Tooltip Template:  Example:

Chart Position:  Auto Height:  Width:  X:  Y:

Axis Grid:  X Major Grid  X Major Tick Mark  X Minor Grid  Y Major Grid  Y Major Tick Mark  Y Minor Grid

Display Options:  Enable Gradients  Display Value Labels  Display Labels Vertically

Chart Series:   Series ID 9 Loaded

Data Mapping: Series Name:   
X Value Field:   
Y Value Field:

---

Explode Field:  (A bit column, if true the chart adds additional visuals)  
Category:  (Defaults to the X Value Field if empty)  
 Render Category Labels  
 Display Labels Vertically

Format:  Example 1:  Example 2:

Opacity (1-10):

Color:

Aggregates:

Chart Type:

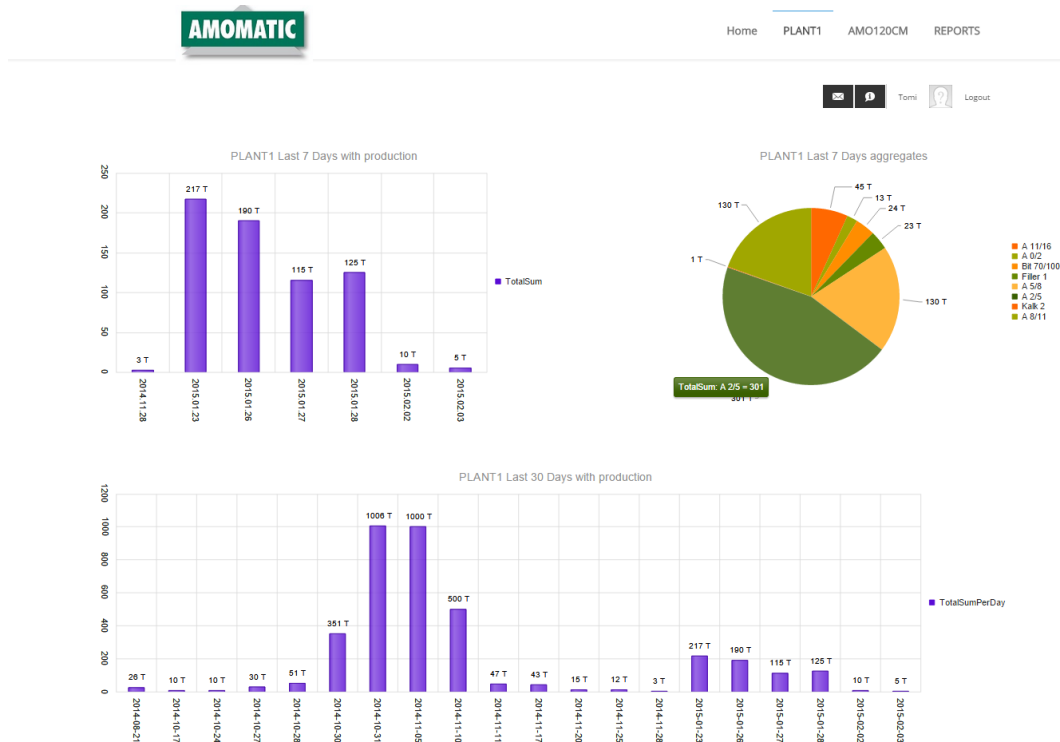
Background Color:

Kuva 12: Onyatech Charts-modulin asetuksia.

Google on tuonut myös mahdollisuuden käyttää ilmaisia grafiikoita datan esittämiseen. Diagrammeja ja mittareita on paljon erilaisia, ja kaikista saa lähdekoodit Googlen sivuilta. Googlen diagrammit käyttävät Javascriptiä, joka voidaan sisällyttää esimerkiksi internet sivuun. Javascriptin alussa on ladattava Googlen Chart-kirjastot ja scriptiin sisällytetään kaikki tieto grafiikan esittämiseksi. Grafiikat luodaan käyttäen HTML5/SVG teknologiaa, joten ne toimivat lähes kaikissa internet kelpoisissa laitteissa. Näytettävä data voidaan kirjoittaa suoraan koodiin tai syöttää sopivassa JSON muodossa suoraan.

Näiden Googlen diagrammien ainoa haaste oli, datan saaminen tietokannasta sopivaan muotoon. DNN Sharpin My Tokens-modulilla tämäkin onnistuu. Sillä voidaan tehdä tietokantakysely ja muuttaa data suoraan JSON-muotoon, joka voidaan syöttää Googlen diagrammin data kohtaan. Onyaktechin diagrammit ovat kuitenkin helpommat käyttää, joten Googlen grafiikat pidetään vaihtoehtona asiakasportaalissa. Googlen grafiikoita käytetään kuitenkin matkapuhelinsovelluksessa kappaleessa 6 Android-sovellus.

Aseman tietosivua klikkaamalla eli tässä tapauksessa virtuaalista "PLANT1"-asemaa avautuu seuraavanlainen näkymä (Kuva 13). Tarkoituksena oli alusta asti saada "dashboard" tyylinen ulkoasu, josta nähdään yhdellä vilkaisulla mahdollisimman helposti paljon tietoa. Tässä versiossa näkyy nyt viimeisen seitsemän tuotantopäivän tuotantomäärät sekä kulutetut raaka-aineet. Lisäksi alla pidemmän aikavälin eli viimeisen 30 päivän tuotantomäärät. Tästä puuttuu resepteittäin tehdyt tilaukset, jonka tarpeellisuus ilmeni myöhemmin. Samoin seuraavalle sivulle on tehtävä tietokanta haku, josta voidaan hakea raportteja tietyltä aikaväliltä.



Kuva 13: Virtuaaliaseman tuotantotiedot

## 5 Azure-pilvipalvelu

Pilvipalvelut ovat nykyään tärkeässä roolissa, ja lähes kaikkien ihmisten käytössä joko tietoisesti tai tietämättään. Pilvipalveluita ovat mm. Azure, Google Cloud, Dropbox ja niitä käyttävät esimerkiksi mobiiliohjelmat, kuten Google Apps, SoundHound ja monet muut. Puhekielessä pilvipalveluna voidaan tarkoittaa yksittäistä levytilaa pilvessä, mutta yrityskäyttöön tarkoitettu pilvipalvelu tarkoittaa useimmiten internetiin kytkettyä ohjelmistoalustaa, joka on skaalatuva konesalissa ja käyttäjän konfiguroitavissa. Pilvipalvelu on aivan kuin oma palvelin, johon voidaan asentaa ohjelmia ja ominaisuuksia, mutta on skaalautuvampi käytettävän prosessori-, muisti- ja kiintolevykapasiteetin suhteen kuin tavallinen palvelin. Usein pilvipalvelussa veloitetaan käytön, tarvittavan tehon ja ulos ladattavan tietomäärän mukaan. Näin pilvipalvelun ostajan ei itse tarvitse huolehtia kalliin palvelimen ostamisesta, päivittämisestä ja ylläpidosta.

Amomaticin asiakasportaalia varten pilvipalveluja vertailtiin Microsoft-pohjaisina, koska pilvipalveluportaalin pohjaksi valittiin aiemmin DNN. DNN:n asennus ja kaikki moduulit toimivat koepalvelimella loistavasti, joten seuraava askel oli saada samanlainen

pilvipalveluun. Listalla olivat suomalainen Nebula, muutama ulkomainen toimija ja Microsoftin Azure-pilvipalvelu. Hinnoittelu ja ilmainen kokeilukuukausi käänsivät valinnan Microsoftin Azure-pilvipalvelun hyväksi.

Microsoftin perustama Azure pilvipalvelu tuli kaupalliseen käyttöön vuonna 2010. Vuonna 2012 tuotteisiin lisättiin alusta internet sivuille ja virtuaalikoneet esimerkiksi Linuxille. Nykyään Azuresta saa skaalautuvia alustoja internet sivuille, tietokantapalvelimia, virtuaalikoneita, tallennustilaa, mobiilipalveluita ja paljon muuta. Ominaisuus jolla Azurea myydään on se, että asiakas maksaa vain käytöstä. Palvelu on hinnoiteltu eri tasoisiiin ryhmiin ja niistä maksetaan ulospäin menevän datamäärän mukaan tai kulutetun prosessoriajan mukaan. Kaikki lisenssiasiat kuuluvat hintaan Microsoftin puolelta. Virtuaalikoneet ja niihin asennettavat sovellukset ovat tietenkin asia erikseen. Azureen on helppo rekisteröityä luottokortilla ja tutustumisetuna saa 150e/1kk ajan kokeilla sen toiminnallisuutta.[16]

## **5.1 Palvelun ominaisuuksia**

Azureen sisäänkirjautumisen jälkeen voidaan valita lukemattomia erilaisia toimintoja, kuten kuvasta 14 näkyy. Esimerkiksi web-sivuston käyttöönotto tai relaatiotietokannan tekeminen on tehty hyvin helpoksi Azuren portaalin kautta. Painamalla "NEW" voidaan valita toiminnon tyyppi, määritetään salasanat ja tarvittava tietokannan koko tai asennetaan esimerkiksi CMS suoraan kirjastosta.

Tietokantaan saadaan suoraan yhteysosoite, jota voidaan käyttää esimerkiksi Microsoft SQL Managerin kanssa tai web-sivuston moduleiden yhteysosoitteissa. Tietokannan muutokset voidaan siis tehdä joko portaalin kautta tai Microsoft SQL Managerin avulla omalta tietokoneelta.



The screenshot shows the Microsoft Azure portal interface. On the left is a navigation sidebar with categories like WEB APPS, VIRTUAL MACHINES, MOBILE SERVICES, CLOUD SERVICES, SQL DATABASES, STORAGE, HDINSIGHT, MEDIA SERVICES, SERVICE BUS, VISUAL STUDIO ONLINE, CACHE, BIZTALK SERVICES, RECOVERY SERVICES, CDN, AUTOMATION, and SCHEDULER. The main area displays a table of resources under the heading 'all items'.

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
AmomaticDataSite	Web app	Running	Pay-As-You-Go	North Europe
Default Directory	Directory	Active	Shared by all Default Dir...	Europe, United States
DNND8	SQL Database	Online	Pay-As-You-Go	North Europe
Plant1	SQL Database	Online	Pay-As-You-Go	North Europe
AMO120CM	SQL Database	Online	Pay-As-You-Go	North Europe
RAPORTS	SQL Database	Online	Pay-As-You-Go	North Europe

At the bottom of the portal, there is a toolbar with icons for NEW, BROWSE, STOP, RESTART, MANAGE DOMAINS, DELETE, and WEBMATRIX.

Kuva 14: Azuren toimintapaneeli

Tässä listattuna muutamia Azure-pilvipalvelun eri ominaisuuksia:

- Windows ja Linux virtuaalikoneet saa nopeasti toimintakuntoon.
- Azure antaa pohjan pilvipalvelusovelluksille, kuten ohjelmistorajapinnoille (PHP,Java jne.)
- SQL-tietokanta, dokumenttikanta
- Käytettävä ja skaalattavissa oleva pilvitallennustila
- StorSimple hybriditallennustila yrityksille
- Redis Cache-välimuistipalvelu kun vaatimuksena on pienet viiveet
- Azure Search-hakupalvelu
- Analysointityökaluja
- Verkon käyttöjen muunneltavuus ja erilliset valokuituyhteydet Azureen
- Median suoratoistoa suoraan skaalattuna
- Palvelimen varmuuskopiointi Azureen
- Visual Studio Online-suunnitteluohjelmisto
- Pilvipalveluhallinnan automatisointia ja ajastamista [16]

## 5.2 DNN Asennus Azureen

Azure-pilvialusta DNN:n pohjaksi on skaalautuva ihan perus käytöstä aina suurien yhtiöiden vaatimiin suuriin portaaleihin asti. Hinnoittelu perustuu siis käyttöön, jolloin siitä maksetaan prosessoriajan ja tiedonsiirtomäärän (Azuresta ulospäin) mukaisesti. Azureen rekisteröinnin jälkeen DNN:n asennus voidaan toteuttaa käyttäen asennusta suoraan Azuren kirjastosta. Tämän jälkeen voidaan valita SQL-tietokannan tyyppi: Basic, Standard, Premium tai poistuvista Web, Business kuten kuvassa 15. Mielenkiintoista oli se, että Basic-valinta ei toiminut. Teknisestä asiakaspalvelusta kehoitettiin vaihtamaan ilmainen kokeilujakso maksulliseksi. Liittymätyyppi konvertoitiin, mutta tuloksetta, koska järjestelmä odottaa ilmaisjakson loppuun ensin. Muutaman päivän odottelun jälkeen kokeiltiin poistuvaa Web-tyyppistä SQL-tietokantaa, joka toimi täysin ongelmitta. Usean viikon keskustelun jälkeen ainoa tapa, jolla Microsoftin tuki olisi saanut Basic-tietokannan toimimattomuuden korjattua, olisi ilmeisesti ollut ainoastaan ilmaisen kokeilujakson välitön muuttaminen maksulliseksi.

i Web and Business Service tiers will be retired. ▼

x

NEW SQL DATABASE - CUSTOM CREATE

### Specify database settings

NAME !

SUBSCRIPTION

SERVICE TIERS                      RETIRED TIERS

BASIC
STANDARD
PREMIUM

WEB
BUSINESS ?

MAX SIZE

 ?

COLLATION

 ?

SERVER

✓

Kuva 15: Azure tietokannan valinta

Azureen DNN:n asentaminen onnistuu siis periaatteessa varsin mutkattomasti, kunhan muutamasta ongelmasta päästään ylitse. Kun DNN paketti on ladattu ja purettu, voidaan normaali DNN asennus aloittaa samalla tavalla kuin aiemmin kohdassa: DNN:n asennus omalle palvelimelle. DNN asennettiin siis ensiksi omalle palvelimelle testausta varten, jonka jälkeen tehtiin Azure-pilvipalveluasennus oikeaa jaettavaa käyttöä varten.

DNN-asennuksen jälkeen tehtiin vielä kaksi uutta tietokantaa asemia varten. Ensimmäinen tietokanta oli simulaatiokonetta varten ja toinen tulevaa konttimallista asemaa varten. DNN Sharpin DNN API End-point mahdollisti nyt toimivan pohjan mobiilisovelluksen kyselyjä varten. Lisäksi asennettiin DNN Sharpin My Token ja Onyaktechin Charts-modulit. Azuren tietokanta synkronoidaan aina onnistuneen aseman tietokannan synkronoinnin jälkeen, joten oma palvelin toimii näin tiedon jakajana pilvipalvelulle.

## 6 Android-sovellus

Internet ja sitä kautta lähes kaikki data on mahdollista saada nykypäivänä älylaitteisiin, kuten matkapuhelimiin ja tabletteihin. Amomaticilla on kauan ollut halu saada oma matkapuhelinsovellus. Sovellus on tarkoitettu myyntitarkoitukseen ja aseman päälliköille, jotka haluavat tietää mitä asemilla tapahtuu. Sovellukseen olisi hyvä saada aseman tuotantotietoja, raaka-ainetietoja ja muuta tarpeellista. Ohjelmaan syötettäisiin aseman tunnistetiedot, jonka jälkeen tietoja voisi katsella. Tämä vaatii kuitenkin pohjakseen palvelimen, josta datat haetaan. Tätä ei aiemmin ole ollut, joten perusprojektin valmistuttua voidaan matkapuhelinsovellusta miettiä tarkemmin. Sovellus vaatii kuitenkin API-rajapinnan palvelimelle, josta tiedot voidaan noutaa. Tämä siksi, että on turvallisempaa käyttää rajapintaa kuin yrittää suoraa yhteyttä Androidista tietokantaan.

Pilvipalvelun ollessa toiminnassa voidaan yhteys ottaa matkapuhelinohjelmasta ja pyytää dataa. DNN Sharpilta löytyy DNN-API Endpoint-moduli, jolla kyseinen API-rajapinta voidaan tehdä. Tietokantahaku voidaan tehdä suoraan ohjelmassa, jonka jälkeen se voidaan muuttaa JSON-muotoon. Ohjelmalla voidaan asettaa kyselylle APIKEY eli avain, ja modulin ID-koodi joiden avulla kysely identifioidaan. Näinollen

matkapuhelinsovelluksessa pitää siis asettaa kaksi asiaa: APIKEY ja module ID, jonka jälkeen pyydettyä dataa voidaan saada.

Alustavasti on tarkoitus pyytää ennalta määritettyä dataa eli 7 viimeisen tuontantopäivän tuontantodata sekä samojen päivien raaka-ainemäärät. Näin kysely ei alussa tarvitse olla molemminpuolista vaan palvelin antaa datan mikäli kirjautumistiedot ovat oikein.

## **6.1 DNN Sharp**

Ohjelmistoyritys DNN Sharp tekee ja myy ohjelmistomoduleita DNN alustalle. Tavoitteenaan heillä on tehdä laadukkaita tuotteita asiakaslähtöisesti ja takaavat hyvän asiakaspalvelun. Yrityksen päätoimipaikka on Singaporessa ja moduleita on muun muassa DNN API Endpoint, My Tokens, NavXP, DNN Sharp. [17]

Api-Endpoint on moduli, jolla voidaan tehdä niin sanottu REST API rajapinta, joka mahdollistaa ulkopuolisten ohjelmien toiminnan rajapinnan kautta. Tietoa voidaan hakea esimerkiksi relaatiotietokannasta ja muuntaa tiedot JSON-muotoon. Kun tiettyyn osoitteeseen tehdään pyyntö (GET) saadaan data JSON-muodossa. Saman REST-rajapinnan kautta voidaan siis käyttää seuraavia komentoja:

- GET: haetaan tietoja ennalta määrätystä lähteestä
- POST: lähetetään tietoa prosessoitavaksi
- PUT: korvaa kaiken datan määrätystä sijainnissa.
- DELETE: poistaa määrätyn tiedon

Tällä modulilla voidaan siis tehdä lukematon määrä erillaisia ohjauksia ja kyselyitä sivuston ulkopuolelta esimerkiksi mobiiliohjelmista. [17]

My Tokens modulilla voidaan tehdä dynaamista sisältöä sisältävä tunnistetunnus (token), jota voidaan käyttää staattisissa/muuttumattomissa paikoissa. Voidaan tehdä esimerkiksi token, jota kutsumalla saadaan JSON-muodossa suoraan tietokannasta haetut tiedot. Näin voidaan tehdä automatisoituja toimintoja DNN sivustolla. Google Charts diagrammit saadaan helposti sivuille käyttämällä datassa My Tokens-tunnistetta, joka hakee sopivan datan tietokannasta ja muuttaa sen JSON muotoon. [17]

JSON on kevyt, tekstiperustainen tiedonsiirtoformaatti, joka toimii muiden ohjelmointikielien välillä ja on hyödyllinen useissa profiileissa ja ohjelmissa. JSON tarjoaa helpon merkintätavan ilmaista sarjan nimiä ja arvoja. Koska JSON on yksinkertainen formaatti, on todennäköistä ettei se koskaan muutu. Tämä tekee siitä varman merkintätavan ja varmistaa hyvän stabiiliuden. Formaatti on ensimmäistä kertaa esitelty maailmalle osoitteessa JSON.org vuonna 2001. JSON arvot voivat olla objekti, rivi, numero, jono.[18]

## 6.2 *Basic 4 Android*

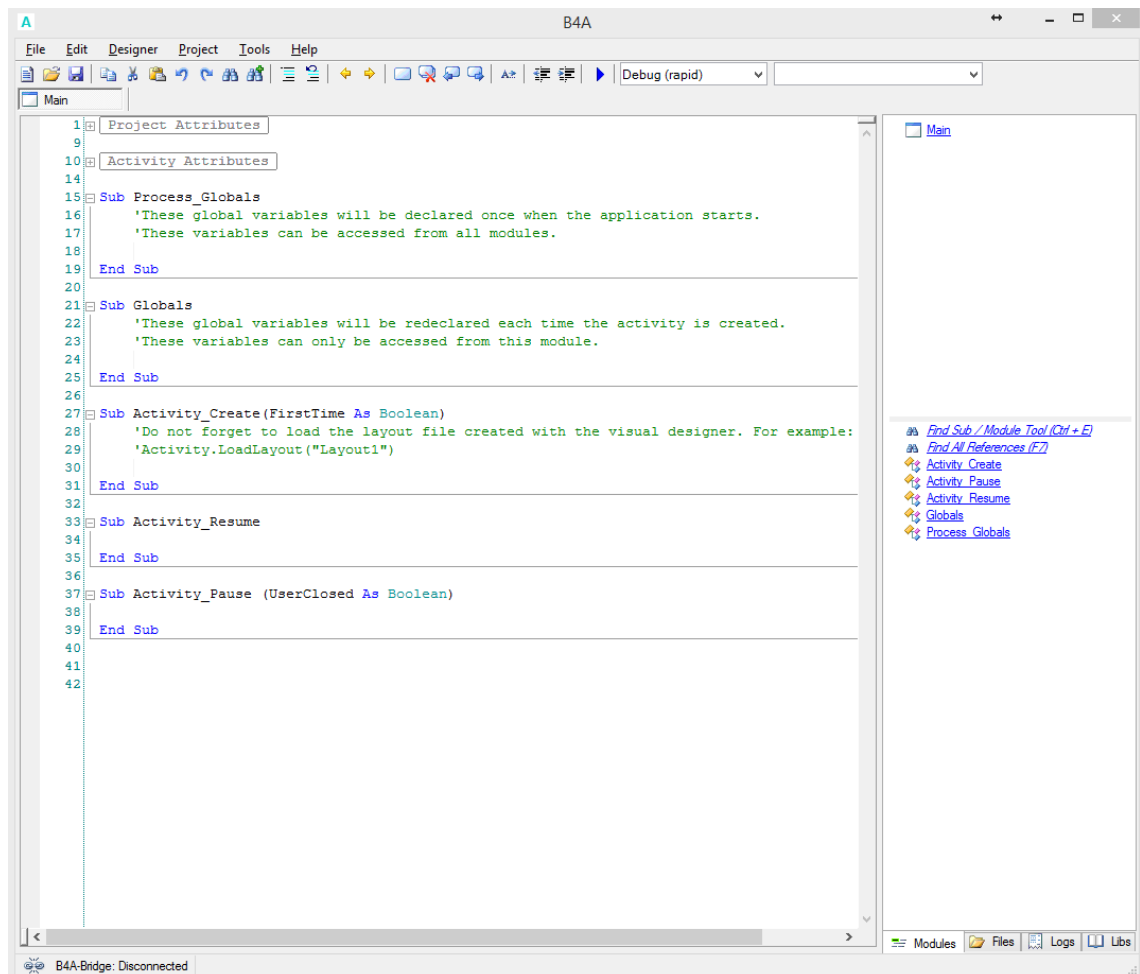
Erel Uziel on perustaja ja toimitusjohtaja yrityksessä Anywhere Software. Yrityksen päätoimipaikkana toimii Israel ja se on erikoistunut erityisesti mobiiliohjelmien kehitykseen vuonna 2005. Yritys kehittää markkinoille nopeasti ohjelmitavia työkaluja erilaisille alustoille, kuten (B4A) Basic4Android, (B4i) Basic4ios ja (B4J) Basic4Java. Esimerkiksi B4A:ta käyttää kymmenet tuhannet kehittäjät ympäri maailmaa sisältäen yrityksiä kuten NASA, HP ja IBM. Ohjelmointikieli on helppo omaksua ja samalla koodilla voidaan tehdä niin Android kuin ios ohjelmatkin hyvin pienin muutoksia. [19]

Ohjelma toimii siis eräänlaisena kääntäjänä B4A->Java. B4A on hyvin saman tyylistä kuin Visual Basic-ohjelmointikieli. Ohjelmistorakenne on seuraavanlainen:

- Sub Process\_Globals <koodia> End Sub  
Tänne määritellään yleiset muuttujat kun ohjelma käynnistyy. Nämä muuttujat ovat kaikkien moduleiden käytettävissä.
- Sub Globals <koodia> End Sub  
Tänne asetetaan muuttujat, jotka määritetään aina kun aktiviteetia käytetään.  
Nämä muuttujat ovat käytettävissä vain tässä modulissa.
- Sub Activity\_Create (FirstTime As Boolean) <koodia> End Sub  
Tässä suoritetaan ohjelmakoodit, jotka suoritetaan ensimmäisen kerran kun moduli ladataan. Esimerkiksi ladataan designerilla tehty ”pohja”.  
Activity.LoadLayout("Layout1")

- Sub Activity\_Resume <koodia> End Sub  
Koodi suoritetaan kun ohjelmaan palataan.
- Sub Activity\_Pause (UserClosed As Boolean)<koodia>End Sub  
Koodi suoritetaan kun ohjelma laitetaan taustalle.

B4A:ssa käytetään kirjastoja, jotka on tehty tiettyjä toimintoja varten, kuten esimerkiksi SQL, GPS, HTTP2Tools. Jokaiselle kirjastolle on omia komentoja, joita ohjelma tarjoaa automaattisesti kun kirjasto on käytössä. Lisäksi voidaan käyttää ohjelman sisäisiä moduleja, jotka helpottavat tiettyjä toimintoja ohjelman sisällä. B4A:ssa on myös graafinen suunnittelu ohjelma, jolla suunnitellaan tekstien, nappien ja kuvien paikat. Nämä voidaan ottaa käyttöön myös ilman, mutta graafisella suunnitteluohjelmalla se onnistuu paljon helpommin. Lisäksi se tekee tarpeelliset alustamiset sen jälkeen automaattisesti. B4A:n käyttöliittymä kuvassa 16.



Kuva 16: B4A:n käyttöliittymä

Kun ohjelma on saatu valmiiksi voi sen kääntää ja asentaa suoraan matkapuhelimeen esimerkiksi WLAN:in ja erityisen B4A-Bridgen avulla tai emulaattoriin. Emulaattori on suoraan Androidin kirjastoon kuuluva matkapuhelin emulaattori.

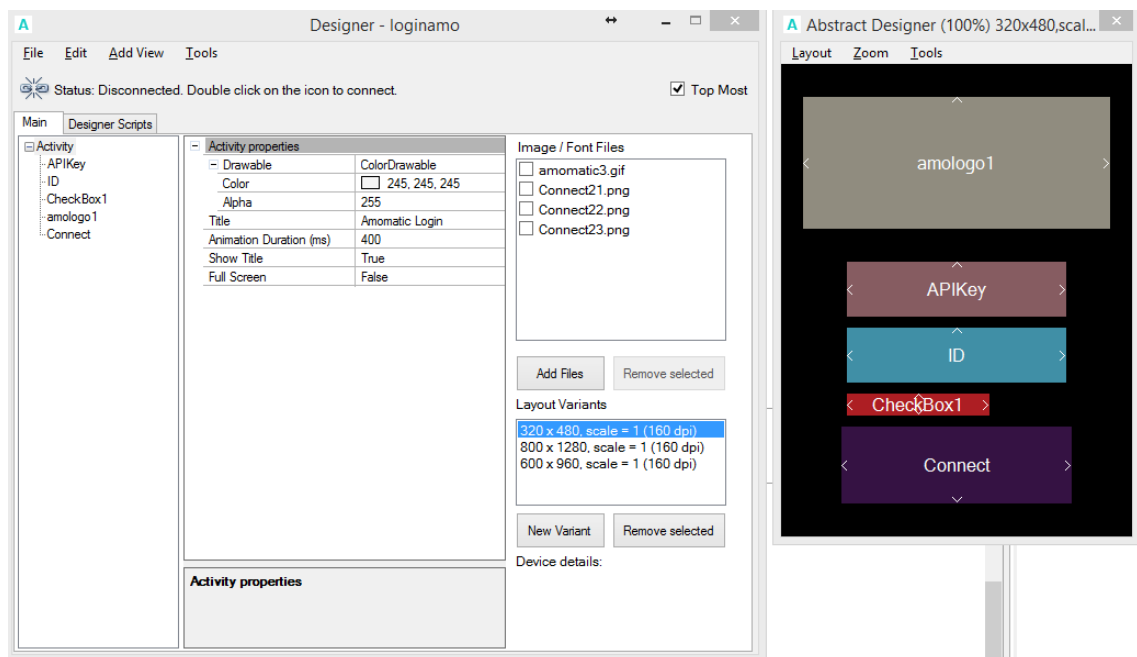
### 6.3 Android ohjelman tekeminen

Ohjelma tehtiin Basic For Android(B4A) ohjelmalla, joka helpottaa Javaa osaamattomien työtä androidin sovelluskehityksessä. Ohjelma itsessään perustuu samaan androidin peruskirjastoon ja emulaattoriin.

Kun API endpoint vastaa sopivaan kyselyyn JSON-datalla, voidaan ohjelmaa alkaa tekemään. Aluksi tehtiin etusivu, johon tulee seuraavat asiat:

- logo kuvatiedosto
- tekstikentät avainta ja moduulin id:tä varten
- checkbox avainten tallennusta varten
- (connect) yhteyden avaus nappi.

Nappi ja tekstikentät voidaan tehdä B4A:n visuaalisella suunnittelijalla (Kuva 17):



Kuva 17: B4A Abstract Designer

Ohjelmaan voidaan tehdä monia erilaisia määrittelyjä, mutta perusrunko Amomatic-ohjelmaan muodostuu seuraavasti: Ensimmäiseen Activity-moduuliin tehdään yhteyden alustus ja JSON koodin haku, käyttäen B4A:n httpools2-kirjastoa. Yhteysosoite rakennetaan käyttäen käyttäjän määrittelemää avainta(APIKEY), ja id:tä(ID). Yhteysosoitteen perään siis lisätään modulin(method), nimi ja apikey.

```
Dim srv As String =
"http://osoite.net/DesktopModules/DnnSharp/DnnApiEndpoint/Api.ashx?"

Dim srvnew As String = srv & "method=" & ID & "&apikey=" & APIKey
Job1.Initialize("Job1", Me)
Job1.Download(srvnew)
```

Jos yhteys onnistuu voidaan JSON tallentaa ja avata seuraava aktiviteetti (DataGraph\_1). Koko ohjelma voidaan periaatteessa tehdä yhdessä modulissa (aktiviteetti), mutta koodin selkeyden takia tehdään sisäänkirjautumissivu ja grafiikkapuoli eri aktiviteetteihin.

```
If Job.Success = True Then
json = Job.GetString
StartActivity(DataGraph_1)
```

Tämän jälkeen voidaan periaatteessa grafiikka piirtää käyttäen Googlen grafiikkaa:

```
Sub Graph1
Dim template As String
template = File.ReadString(File.DirAssets, "template1.html")

WebView1.ZoomEnabled = False
Dim options As Map
options.Initialize
CreateColumnChartOptions1(options)
Dim jg As JSONGenerator
Dim html As String = template
html = html.Replace("$TYPE$", "Bar" & "Chart")
jg.Initialize(options)
html = html.Replace("$OPTIONS$", jg.ToString)
html = html.Replace("$COLUMN1$", "'string', 'Date'")
html = html.Replace("$COLUMN2$", "'number', 'TotalSumPerDay'")
html = html.Replace("$ROWDATA$", Main.json)
WebView1.LoadHtml(html)
End Sub
```



Koodissa ladataan "template1.html"-tiedosto, jossa on Googlen perusdiagrammin koodi:

```
<html>
<head>
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">
google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
    var data = new google.visualization.DataTable();
    data.addColumn($COLUMN1$);
    data.addColumn($COLUMN2$);
    data.addRows($ROWDATA$);

    var options = $OPTIONS$
var chart = new google.visualization.$TYPE$(document.getElementById('$TYPE$'));
    chart.draw(data, options);
    }
</script>
</head>
<body>
<div id="$TYPE$" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

Tiedoston tiettyihin kohtiin, kuten esim. "\$OPTIONS\$" tehdään muualla ohjelmassa sopivat määrittymiset oikeassa muodossa ja tallennetaan siihen kohtaan. JSON voidaan syöttää suoraan "\$ROWDATA\$" -kohtaan, jos se on suoraan sopivassa muodossa kuten: [ ["2014.11.20",15],["2014.11.25",12],["2014.11.28",3],["2015.01.23",369],  
["2015.01.26",190],["2015.01.27",115],["2015.01.28",125]]

Tässä JSON muodossa tulee siis ensiksi päivämäärä (DATE) ja tonnimäärä (TotalSumPerDay).

Jos JSON:ia ei saada suoraan sopivaan muotoon lähetyksessä, pitää se purkaa ja poimia tarpeellinen tieto siitä. Tähän voidaan käyttää JSONParser-toimintoa. Googlen

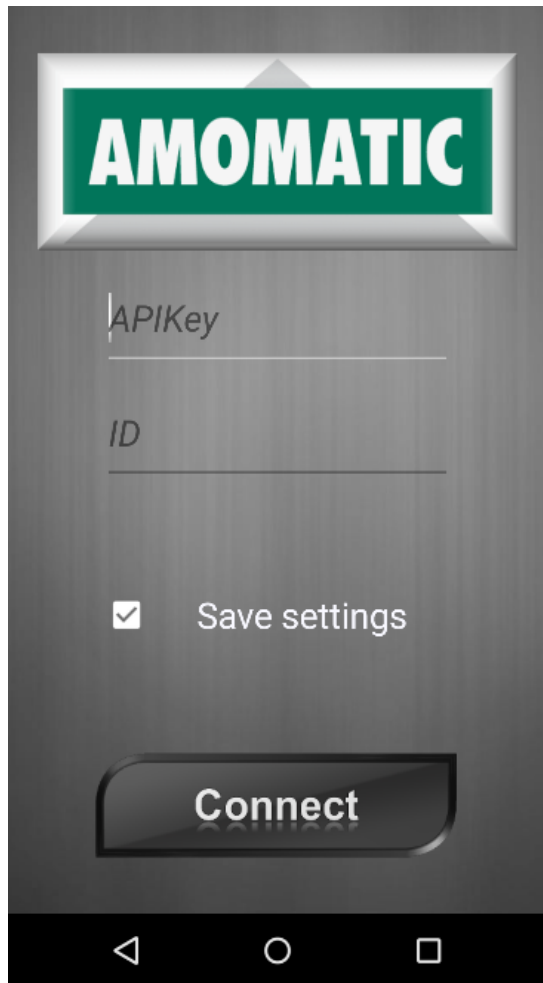
grafiikan data voidaan syöttää myös Googlen omassa muodossa, jolloin olisi helpompaa syöttää sarakearvot ja dataarvot dynaamisesti suoraan.

```
var data = new google.visualization.DataTable(
  {
    cols: [{id: 'task', label: 'Employee Name', type: 'string'},
           {id: 'startDate', label: 'Start Date', type: 'date'}],
    rows: [{c:[{v: 'Mike'}, {v: new Date(2008, 1, 28)}, f:'February 28, 2008'}]},
           {c:[{v: 'Bob'}, {v: new Date(2007, 5, 1)}]},
           {c:[{v: 'Alice'}, {v: new Date(2006, 7, 16)}]},
           {c:[{v: 'Frank'}, {v: new Date(2007, 11, 28)}]},
           {c:[{v: 'Floyd'}, {v: new Date(2005, 3, 13)}]},
           {c:[{v: 'Fritz'}, {v: new Date(2011, 6, 1)}]}
    ]
  }
)
[20]
```

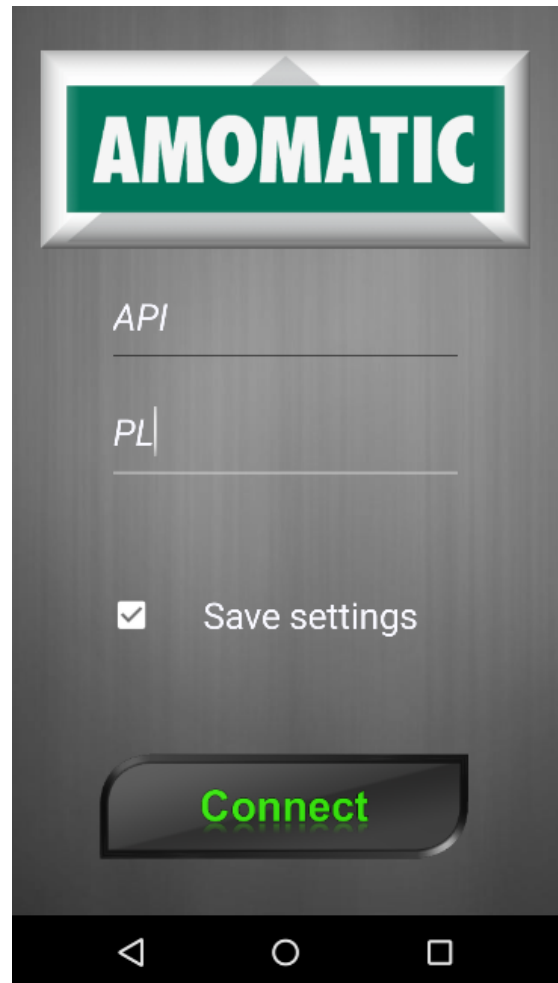
Tässä muodossa ongelmana on vian hankala haku ja hankala luettavuus. Tähän kuitenkin pyritään Amomaticin ohjelmassa v.2.0, koska silloin data voi sisältää enemmän dynaamista tietoa kuin tällä hetkellä.

### 6.3.1 Ohjelma nyt ja suunnitelmat

Amomaticin ensimmäiseen Android-ohjelmaversioon lisättiin taustakuva, tehtiin kuvankäsittelyohjelmalla connect-nappi, ja sisäänkirjaustietojen tallennuskohta. Tietojen tallennukseen käytettiin KeyValueStore-modulia, joka tallentaa tärkeät tiedot puhelimen muistiin tiedostona. "Connect" eli yhdistämisnapista tehtiin 3 eri versiota. Valkoinen kun nappi ei ole käytössä, vihreä kun nappi on valmiina yhdistämään ja sininen kun siitä painetaan. Kuvassa 18 ja 19 nähdään napin vaihdokset. Ohjelmaan saadaan viimeisten seitsemän päivän tuotantomäärät pylväsdiagrammina sekä käytetyt raaka-ainemäärät piiras-diagrammina kuten kuvissa 20 ja 21. Interaktiivisuuden takia käytettiin siis Googlen grafiikoita ja saatiin näin sormella näpäyttämällä lisätietoa diagrammeista.

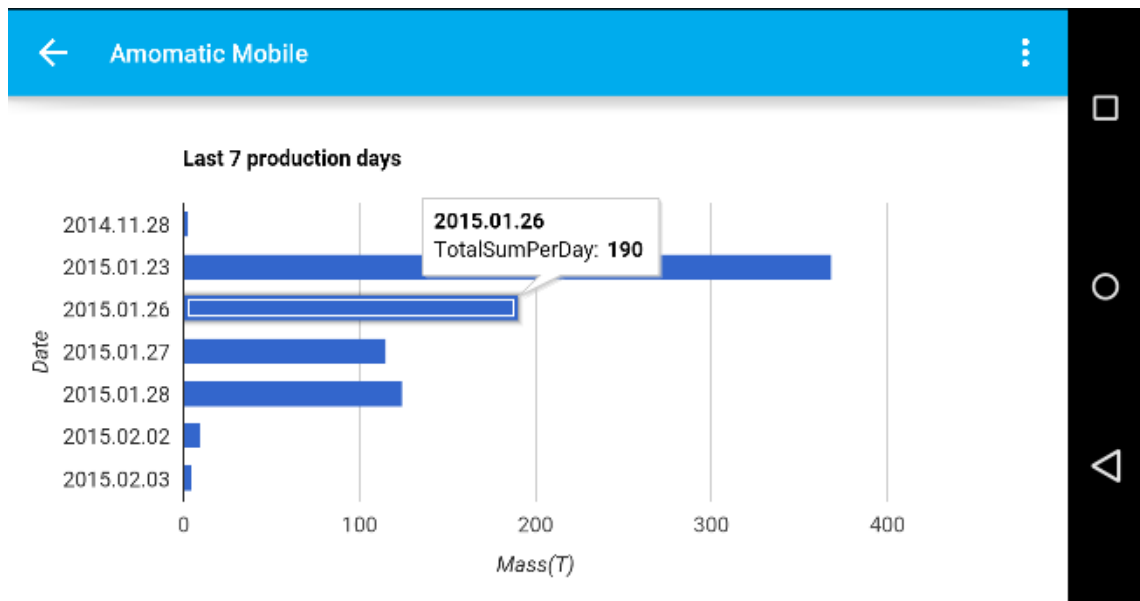


Kuva 18: Amomatic Android-ohjelman alkuruutu.

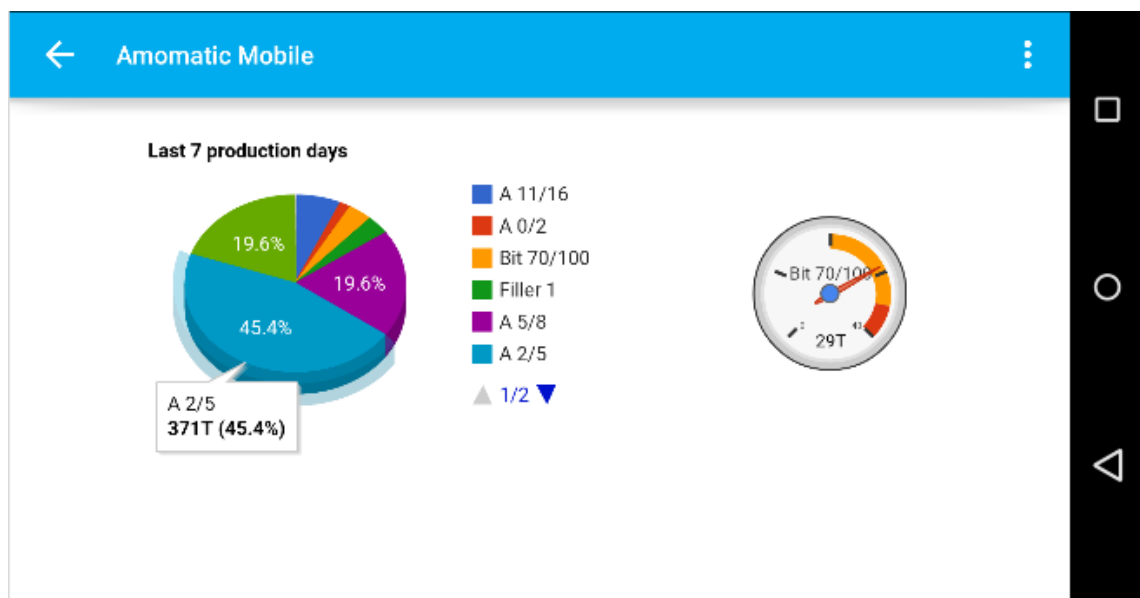


Kuva 19: Ohjelman "connect"-nappi valmiina yhdistämään.

Lisäksi tehtiin vielä erillinen "mittari" bitumille tai polttoaineelle. Tätä voitiin pitää versiona 1.0.0, koska ohjelma toimi täysin ongelmitta aluksi WLAN:lla sisäverkossa ja sittemmin Azuren pilvipalvelun kautta. Ensimmäisessä versiossa nämä tiedot olivat eri sivuilla ja sivuja voitiin vaihtaa ylhäällä olevista lehdistä, kuvissa kuitenkin jo versio 1.1.0

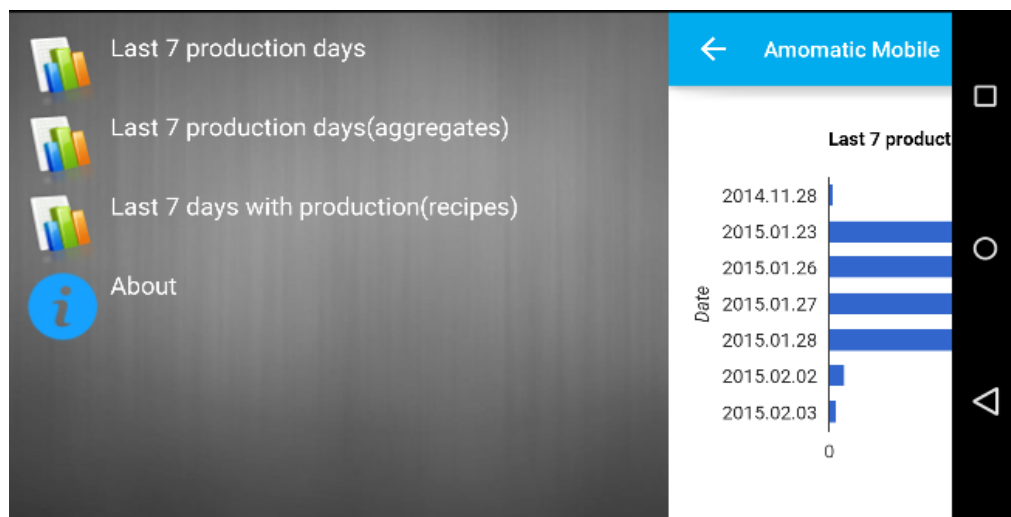


Kuva 20: Viimeisen 7 tuotantopäivän tuotantomäärät.

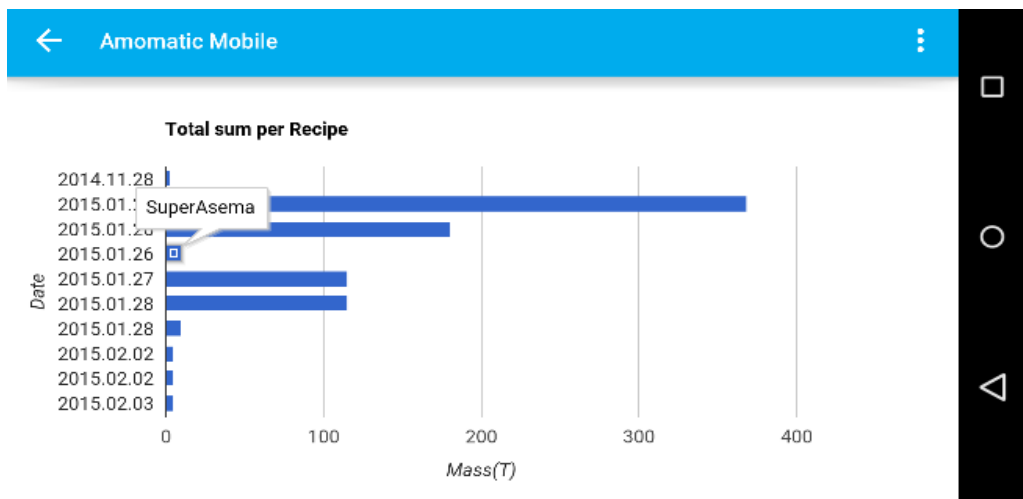


Kuva 21: Viimeisen 7 päivän käytetyt raaka-aineet.

Versioon 1.1.0 haluttiin tehdä erillinen valikon käyttöliitymä ja päädyttiin kokeilujen kautta vasemmalta avautuvaan menuun, johon voitaisiin helpommin lisätä lisää graafeja (Kuva 22). Kokonaisuudessa ja ylämenussa käytettiin uuden 5.0 Androidin "Material Design" tyylistä ulkoasua. Tähän liittyy pieniä hienouksia, kuten varjoja ja värien käyttöä kun käytössä on uusin 5.0 android-käyttöjärjestelmä. Lisäksi haluttiin vielä näyttää tuotantodatat resepteittäin ja tämä tehtiin kolmanteen diagrammiin kuva 23. Lisänä tähän on tarkoitus vielä lisätä liukupalkki, josta voitaisiin haarukoida 1 - 7 päivää diagrammiin.



Kuva 22: Uusi vasemmalta esiin tuleva menu.



Kuva 23: Viimeiset 7 tuotantopäivää resepteittäin.

Ohjelmaa ei haluttu vielä toistaiseksi ladata Google Play-sovelluskauppaan, ja siksi tehtiin API-kyselyllä version tarkistus palvelimelta. Ohjelman käynnistyessä ohjelma katsoo onko uudempaa sovellusversiota tarjolla ja ilmoittaa siitä näytöllä: "Update Available". Kun ohjelmaan on annettu oikeat avaimet ja on päästy katsomaan graafeja, niin "About"-valikon alta voidaan ladata uusien ohjelmaversio ja asentaa se. Näin asiakkaan ei tarvitse ladata ohjelmaa sivuiltamme kuin kerran.

Suunnitelmia ohjelman suhteen on monia, mutta alkuun ohjelma pitäisi saada testaukseen ja kuunnella asiakkaita, mitä ohjelmaan todellisuudessa tarvitaan. Jos tarvitaan paljon erillaista dataa näkyville yhteen näyttöön, niin listamuotoinen taulukko puolustaa paikkaansa. Listamuotoista taulukkoa aiotaankin testata kolmantena

esityksenä, jotta haluttua dataa on helpompi saada esille. Lisäksi kehityslistalla on JSON-koodin purkaminen ja uudelleen muokkaaminen Googlen muotoon, jotta saadaan lisättyä enemmän dynaamista sisältöä. Näin olisi mahdollista sisällyttää JSON-koodiin lisäksi diagrammin tyyppi, asetustiedot, sarakkeet ja data. Näin voitaisiin asiakkaasta riippuen, syöttää erillainen diagrammi tarpeen mukaan.

## 7 Yhteenveto

Projekti tuntui alussa selvältä, vaikka lähes kaikki projektissa mukana olleet ohjelmat olivat entuudestaan tuntemattomia. Pian kävi kuitenkin ilmi, että projekti oli haastava. Tähän vaikutti asemalle asennetut ohjelmat ja lisenssit, jotka kieltävät lisäosien asentamisen. Microsoft SQL Serverin monimuotoisuus asetti myös haasteita, koska saman asian voi tehdä niin monella eri tapaa. Julkaisualustan valinta oli pitkä prosessi, koska nykyään vaihtoehtoja on valtavasti, esimerkiksi Wordpress, Drupal, DNN, Ruby On Rails. Ruby On Rails jäi erikoisuutensa ja hienojen tutorialiensä takia parhaiten mieleen. Se olisi kuitenkin vaatinut yhden uuden ohjelmointikielen periaatteellista hallintaa ja jäi siksi toistaiseksi kokeilematta.

Tiedonsiirtoon mietittiin monia eri vaihtoehtoja. Microsoftin työkalut jäivät rajoitteiden ja hintojen ulkopuolelle, joten kolmannen osapuolen synkronisointiohjelma tuntui kaikkein suoraviivaisimmalta tavalta hyödyntäen samalla Tosiboxin VPN-yhteyttä. Devartin DB-Forge DataSync suoriutui testeistä parhaiten ja se jäi käyttöön. 3G-verkossakaan VPN-yhteyden muodostus ja yhteyden nopeus eivät ole kovin nopeaa, joten synkronisointiohjelman nopeus on oleellinen asia. Synkronisointiohjelmalla voidaan määrittää, kuinka monta päivää taaksepäin tietojensiirto ulottuu ja näin säästetään aikaa päivittäisessä tiedonsiirrossa. VPN:n yhteyden avaus ja synkronisoinnin automatisointi onnistui hyvin ja se on toiminut luotettavasti. Luotettavuus tulee todelliseen testiin vasta silloin, kun asemien määrää lisätään. Tulevaisuudessa voidaan tehdä manuaalisesti taulusta kopiointi ja tyhjennys, jolloin tapahtuma nopeutuu entisestään.

DNN:n valinta pohjaksi vaikuttaa oikealta ratkaisulta, kun periaatteellisia yhteensopivuusongelmia ei ole tullut. Ostetut DNN-modulit ovat toimineet hyvin paitsi Onyaktechin modulit, jotka eivät ole toimineet luvatussa tavalla. Virhetilanteet on raportoitu Onyaktechille ja korjauksia on jo tehty. Asfalttiaseman raporttijärjestelmä on

tehty Visual Studiolla, joten tulevaisuudessa sen hyödyntäminen DNN-sivustolla pitää selvittää. Lisäksi lisättiin myös asfalttiaseman käytettävyyden mittaaminen kyselyllä kolmelle eri asemalle. Tämä tehtiin Forms modulilla ja tiedot tallennetaan suoraan Azuren tietokantaan. Sivusto antaa nyt mahdollisuuden lisätä kaikenlaisia tietoa kerääviä ja tuottavia sivuja.

Android-sovelluksen rakentaminen B4A työkaluin oli haastavaa, mutta myös palkitsevaa. B4A:n mainoslauseessa siitä puhutaan Rapid Application Development (RAD) -työkaluna, jolla saa paljon aikaa pienessä ajassa. DNN API End Point ja My Token modulit antoivat tavan, miten Googlen graafeja voisi mobiilisovelluksessa näyttää, kun datan muuttaa JSON-muotoon. Mobiilisovelluksen tekemisessä suurimmat haasteet olivat liukumenuvalikon graafien asetusten saaminen oikeaan Googlen ymmärtämään muotoon.

Projektin periaatteen selvittyä oli otettava huomioon myös tietoturva asiat. VPN-yhteyden käyttö asemalta Amomaticin palvelimelle on suojattu vahvasti, joten siinä ei ongelmia ole. Pilvipalvelun eli DNN:n tietoturva on myös vuosia koeteltu ja hyväksi havaittu. Siksi sitä käyttävät mm. NASA, Bank of America ja U.S. Dept. of Defense. Sivuille ja moduleille määritellään oikeat käyttäjät erikseen, joten tietoturva siinä suhteessa on kaksinkertainen. Jokainen asema, jossa käytetään mobiiliohjelmia DNN Api-Endpoint-modulin kanssa, on suojattu avaimella, joka voi olla 1 – 15 merkkiä pitkä ja lisäksi pitää tietää ID ennenkuin dataa saadaan. Avain lähetetään http-otsikossa, joten periaatteessa se olisi kaapattavissa yleisen WLANin kautta, jos ohjelmaa siellä käytetään. Avaimen ja ID:n pelkkä arvaaminen on siis erittäin vaikeaa. Jos ne saisi tietoonsa palvelin lähettää ainoastaan JSON-muodossa tuotantotietoja ilman tunnistetietoja. Ilman tunnisteita, tiedoilla ei ole mitään arvoa, koska tietoja ei pystytä varmuudella sijoittamaan mihinkään asemaan. Jos API-kysely halutaan tulevaisuudessa täysin salattavaksi, on siihen olemassa API-Endpoint-modulissa AES ja RSA suojaukset tai koko sivuston muuttaminen https-suojaukselle. Tämä vaatii muutoksia tietysti myös mobiiliohjelmaan.

## 8 Lähdeluettelo

- 1 Amomaticin sisäinen materiaali.
- 2 Features Supported by the Editions of SQL Server 2014. Verkkodokumentti  
<<http://msdn.microsoft.com/en-us/library/cc645993.aspx>>Luettu 15.2.2015
- 3 Timo Rahko. 2007. NDIR-tekniikka kaasujen monikomponenttianalysissä.
- 4 Tosibox Oy. Verkkosivusto.  
<<http://www.tosibox.com/>>Luettu 15.2.2015
- 5 SQL Server 2014-Types of Replication. Verkkodokumentti.  
<<https://msdn.microsoft.com/en-us/library/ms152531.aspx>>Luettu 15.2.2015
- 6 Transaction Statements (Transact-SQL) Verkkodokumentti.  
<<https://msdn.microsoft.com/en-us/library/ms174377.aspx>>Luettu 15.2.2015
- 7 Paul Atkinson. Robert Vieira. 2012. Microsoft SQL Server 2012 Programming
- 8 Editions and Components of SQL Server 2014. Verkkodokumentti.  
<<https://msdn.microsoft.com/en-us/library/ms144275.aspx>>Luettu 15.2.2015
- 9 Dušan Petrović. 2008. Microsoft SQL Server 2008 A Beginner's Guide.
- 10 Wappalyzer uncovers the technologies used on websites. Verkkosivusto.  
<<https://wappalyzer.com/categories/cms>> Luettu 20.2.2015
- 11 Water&stone. 2011. 2011 Open Source CMS Market Share Report.  
Verkkodokumentti. <<http://www.waterandstone.com/downloads/2011OSCMSMarketShareReport.pdf>> Luettu 20.2.2015
- 12 Drupal.org. Luettavissa: <http://www.drupal.org>. Luettu 1.2.2015
- 13 DNN Corporation. Verkkosivusto. <<http://www.dnnsoftware.com/>> Luettu 20.2.2015
- 14 Dnn CMS Platform-sivusto. Verkkosivusto.  
<<https://dotnetnuke.codeplex.com/>>Luettu 20.2.2015
- 15 Onyaktech. Verkkosivusto. <<https://www.onyaktech.com/>> luettu 10.3.2015
- 16 Microsoft Azure. Verkkosivusto <<http://azure.microsoft.com/>> Luettu 1.3.2015
- 17 DNN Sharp. Verkkosivusto.  
<<http://www.dnnsharp.com/>>Luettu 15.2.2015
- 18 Ecma Standard 404. 2013. The JSON Data Interchange Format
- 19 B4A. Verkkosivusto.  
<<http://www.b4x.com/>>Luettu 15.2.2015



20 Google Charts. Verkkodokumentti.

<[https://developers.google.com/chart/interactive/docs/datatables\\_dataviews](https://developers.google.com/chart/interactive/docs/datatables_dataviews)>

Luettu 27.2.2015

Amomatic Android-ohjelman graafisen osuuden koodi, joka sisältää version tarkistuksen ja päivitys-ohjelman.

```
#Region Activity Attributes
    #FullScreen: False
    #IncludeTitle: true
#End Region
#Extends: android.support.v7.app.ActionBarActivity

Sub Process_Globals
    'These global variables will be declared once when the
    application starts.
    'These variables can be accessed from all modules.
    'Private currentPage As Int

End Sub

Sub Globals
    'These global variables will be redeclared each time the
    activity is created.
    'These variables can only be accessed from this module.

    Dim template,template2,template3 As String
    Dim html As String = template
    Dim html2 As String = template2
    Dim html3 As String = template3

    Public json2 As String
    Public json3 As String

    Private WebView1 As WebView
    Private WebView2 As WebView
    Private WebView3 As WebView
    Dim toast As ToastMessageShow
    Dim AC As AppCompatActivity
    Dim ABHelper As ActionBar
    Private ActionBar As ActionBarDark
    Private pContent As Panel
    Private template As String

    Dim sm As SlidingMenu
    Dim clv1,clv2,clv3,clv4 As CustomListView
    Dim Panell1 As Panel
    Dim lbl1 As Label
    Dim OK_DOWNLOAD As Int = 0

    Dim cd As CustomDialog
    Dim CheckBox2 As CheckBox
    Dim ret As Int
    Dim update As Int
End Sub

Sub Activity_Create(FirstTime As Boolean)

    Activity.LoadLayout("Pagemain") ' Loads "Main" layout file
    WebView1.Initialize("Page1")
    WebView2.Initialize("Page1")
    WebView3.Initialize("Page1")

    ActionBar.SetAsActionBar
```

```

AC.SetElevation(ActionBar, 8dip)
'Set Title and Subtitle for the Toolbar
ActionBar.Title = "Amomatic Mobile"
pContent.LoadLayout("page1")

ABHelper.Initialize
ABHelper.ShowUpIndicator = True
Dim xml As XmlLayoutBuilder
Dim bd As BitmapDrawable
'bd = xml.GetDrawable("ic_plus_one_black_24dp")

bd = xml.GetDrawable("ic_refresh_black_24dp")
'Activity.AddMenuItem3("Refresh", "Menu", bd.Bitmap, True)
Activity.AddMenuItem3("Refresh", "Menu", bd.Bitmap, False)

ActionBar.SetLayoutAnimated(200, 0, 0, 100%x,
AC.GetMaterialActionBarHeight)
ActionBar.SetVisibleAnimated(200, True)
pContent.SetLayoutAnimated(200, 0, AC.GetMaterialActionBarHeight,
100%x, 100%y - AC.GetMaterialActionBarHeight)

sm.Initialize("sm")
Dim offset As Int = 40%x
sm.BehindOffset = offset
sm.Mode = sm.LEFT

' EXTRA
Dim jo As JavaObject = sm
jo.RunMethod("setTouchmodeMarginThreshold", Array(50%x))
' EXTRA
Dim jo As JavaObject = sm
Dim degree As Float = 0.5
jo.RunMethod("setFadeDegree", Array(degree))
jo.RunMethod("setFadeEnabled", Array(True))

clv1.Initialize(Me, "clv1")
clv1.DefaultTextColor= Colors.Transparent
sm.Menu.AddView(clv1.AsView, 10dip, 10dip, 100%x - offset - 20dip,
100%y)
clv1.Add(CreateListItem("Last 7 prod. days", clv1.AsView.Width,
50dip), 50dip, "Graph1")
clv1.Add(CreateListItem("Last 7 prod. days (agg.)", clv1.AsView.Width,
50dip), 50dip, "Graph2")
clv1.Add(CreateListItem("Last 14 prod. days (recipes)",
clv1.AsView.Width, 50dip), 50dip, "Graph3")
clv1.Add(CreateListItem2("About", clv1.AsView.Width, 50dip), 50dip,
"About")

Dim b2 As Bitmap
b2.Initialize(File.DirAssets, "bg2.png")
sm.Menu.SetBackgroundImage(b2)
clv1.AsView.Color = Colors.Transparent

update=0
conn2
    Graph1
End Sub

```

```

Sub CreateListItem(Text As String, Width As Int, Height As Int) As
Panel
Dim p As Panel
p.Initialize("")
p.Color = Colors.Transparent
Dim b As Bitmap
b.Initialize(File.DirAssets,"success.png")
Dim i As ImageView
i.Initialize("")
i.Bitmap = b
Dim lbl As Label
lbl.Initialize("lbl")
lbl.Color= Colors.Transparent
lbl.Text = Text
lbl.TextSize = 16
lbl.TextColor = Colors.White
p.AddView(i, 0,0,Height,Height)
p.AddView(lbl, Height+5dip,0,80%x,Height-5dip)
Return p
End Sub

```

```

Sub CreateListItem2(Text As String, Width As Int, Height As Int) As
Panel
Dim p As Panel
p.Initialize("")
p.Color = Colors.Transparent
Dim b As Bitmap
b.Initialize(File.DirAssets,"info.png")
Dim i As ImageView
i.Initialize("")
i.Bitmap = b
Dim lbl As Label
lbl.Initialize("lbl")
lbl.Color= Colors.Transparent
lbl.Text = Text
lbl.TextSize = 16
lbl.TextColor = Colors.White
p.AddView(i, 0,0,Height,Height)
p.AddView(lbl, Height+5dip,0,80%x,Height-5dip)
Return p
End Sub

```

```

Sub clv1_ItemClick(Index As Int, Value As Object)
    'Log(Index & " = " & Value)

```

```

Select Index
Case "0"
pContent.RemoveAllViews
pContent.LoadLayout ("page1")
WebView1.LoadHtml (html)

```

```

'Graph1
sm.HideMenus
Case "1"
pContent.RemoveAllViews
pContent.LoadLayout ("page1")
WebView1.LoadHtml (html2)
sm.HideMenus

```

```

Case "2"
pContent.RemoveAllViews
pContent.LoadLayout ("page1")
WebView1.LoadHtml (html3)
sm.HideMenus

```

```

Case "3"
pContent.RemoveAllViews
pContent.LoadLayout("page2")
clv2.Initialize(Me, "clv2")
Panell1.AddView(clv2.AsView,0,3dip,100%x,100%y)
'clv2.DefaultTextBackgroundColor=Colors.ARGB(255,0,172,237)
clv2.AsView.Color = Colors.Black
clv2.Add(CreateListItem3(Main.version ,clv2.AsView.Width, 50dip),
50dip, "Version")
clv2.Add(CreateListItem3("About", clv2.AsView.Width, 50dip), 50dip,
"About")
clv2.Add(CreateListItem3("Program Update", clv2.AsView.Width, 50dip),
50dip, "Program Update")

sm.HideMenus
End Select
End Sub
Sub clv2_ItemClick(Index As Int, Value As Object)
'Log(Index & " = " & Value)

Select Index
Case "0"

pContent.RemoveAllViews
pContent.LoadLayout("page4")

clv4.AsView.Color = Colors.black
clv4.DefaultTextBackgroundColor = Colors.Black
clv4.AddTextItem("Version:1.1.9" & CRLF & "Table chart and minor
modifications.",1)
clv4.AddTextItem("Version:1.1.8" & CRLF & "Fixes and program update
function",1)
clv4.AddTextItem("Version:1.1.5"& CRLF &"Totally new sliding menu.",1)
clv4.AddTextItem("Version:1.1.0"& CRLF &"Recipes Chart",1)

Case "1"
pContent.RemoveAllViews
pContent.LoadLayout("page3")
clv3.AsView.Color = Colors.black
clv3.DefaultTextBackgroundColor = Colors.Black
clv3.AddTextItem("©2015 Code By Tomi Saloranta, Amomatic Oy",1)
clv3.AddTextItem("Sliding Menu" & CRLF & "©2012 Jeremy Feinstein" &
CRLF & "SlidingMenu is licenced under Apache License Version 2.0",1)
clv3.AddTextItem("Licensed under the Apache License, Version 2.0 (the
License);you may Not use this File except In compliance with the
License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0" & CRLF &"Unless required
by applicable law OR agreed To In writing, software distributed under
the License Is distributed on an 'As Is' BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express OR implied.See the License For
the specific language governing permissions AND limitations under the
License.",1)

Case "2"
vercheck2
If update=1 Then
    Dim Bmp As Bitmap
    Dim pnl As Panel
    pnl.Initialize("pnl")
    Dim bgnd As ColorDrawable
    bgnd.Initialize(Colors.Transparent, 5dip)
    pnl.Background = bgnd

```

```

CheckBox2.Initialize("chkbox")
CheckBox2.TextSize=16
CheckBox2.Text = "Check to Update"
pnl.AddView(CheckBox2, 50dip, 20dip, 100dip, 60dip)

lbl1.Initialize("")
pnl.AddView(lbl1, 5dip, 120dip, 100dip, 60dip)
cd.AddView(pnl, 5%x, 0%y, 30%x, 30%y) ' sizing relative to the
screen size is probably best
ret = cd.Show("Amomatic Mobile Update","OK", "Cancel","",Null)
If ret = DialogResult.POSITIVE AND CheckBox2.Checked= True
Then
ToastMessageShow("Updating...", True)
upgrade
Else
ToastMessageShow("Cancelled!", True)
End If
Else
ToastMessageShow("No Update Available!" ,True)
End If
End Select
End Sub

Sub vercheck2
Dim vercheck As HttpJob
    Dim srvver As String = "http://XXX/Api.ashx?
method=VERSION&apikey=XXX"
    vercheck.Initialize("vercheck",Me)
    vercheck.Download(srvver)
End Sub

Sub CreateListItem3(Text As String, Width As Int, Height As Int) As
Panel
Dim p As Panel
p.Initialize("")
p.Color = Colors.Transparent
Dim b As Bitmap
b.Initialize(File.DirAssets,"info.png")
Dim i As ImageView
i.Initialize("")
i.Bitmap = b
Dim lbl As Label
lbl.Initialize("")
lbl.Color= Colors.Transparent
'lbl.Gravity = Bit.OR(Gravity.CENTER_VERTICAL, Gravity.LEFT)
lbl.Text = Text
lbl.TextSize = 16
lbl.TextColor = Colors.White
p.AddView(i, 0,0,Height,Height)
p.AddView(lbl, Height+5dip,0,80%x,Height-5dip)
Return p
End Sub

Sub Activity_ActionBarHomeClick
sm.ShowMenu
    'Log("bar button clicked")
End Sub
Sub Menu_Click
'Log("refresh button clicked")
conn1
End Sub

Sub CheckBox2_CheckedChange(Checked As Boolean)

```

```

ToastMessageShow("Updating...", True)
    If cd.Response=-1 Then
        End If
End Sub

Sub upgrade
Dim upgradelink As String = "http://xx/Api.ashx?
method=Application&apikey=xx"
    Dim jobapk As HttpJob
    jobapk.Initialize("JobApkDownload",Me)
    jobapk.Download(upgradelink)
End Sub

Sub InstallApk
If OK_DOWNLOAD = 1 Then
    'Log("---- AppUpdating.InstallApk")
    'intent to install
    Dim i As Intent
    i.Initialize(i.ACTION_VIEW, "file://" &
File.Combine(File.DirDefaultExternal, "tmp.apk"))
    i.SetType("application/vnd.android.package-archive")
    StartActivity(i)
    Else
    ToastMessageShow("Download is not accomplished correctly.", True)
End If

End Sub
Sub Activity_KeyPress (KeyCode As Int) As Boolean
    If KeyCode = KeyCodes.KEYCODE_BACK AND sm.Visible Then
        sm.HideMenus
        Return True
    End If
    Return False
End Sub

Sub conn1
Dim Job1 As HttpJob
Dim srv As String = "http://xx/Api.ashx?"
Dim srvnew As String = srv & "method=" & Main.ID2 & "&apikey=" &
Main.APIKey2
'Log(srvnew)
Job1.Initialize("Job1",Me)
Job1.Download(srvnew)
End Sub

Sub conn2
Dim Job2 As HttpJob
Dim srv As String = "http://xx/Api.ashx?"
Dim ID3 As String =Main.ID2 & "2"
''Log(APIKey2 & ID3)
Dim srvnew2 As String = srv & "method=" & ID3 & "&apikey=" &
Main.APIKey2
'Log(srvnew2)
Job2.Initialize("Job2", Me)
Job2.Download(srvnew2)
End Sub

Sub conn3
Dim Job3 As HttpJob
Dim srv As String = "xxx/Api.ashx?"
Dim ID4 As String = Main.ID2 & "3"
''Log(APIKey2 & ID4)
Dim srvnew3 As String = srv & "method=" & ID4 & "&apikey=" &
Main.APIKey2

```

```

Job3.Initialize("Job3", Me)
Job3.Download(srvnew3)
End Sub

Sub JobDone(Job As HttpJob)
Dim vernro As String
'Dim error As String

If Job.Success = True Then
    Select Job.JobName
    Case "vercheck"
        vernro = Job.GetString
        If vernro = Main.version Then
            update=0
        Else
            update=1
        End If

    Case "Job1"
        Main.json = Job.GetString
        'Log(Main.json)
        conn2

    Case "Job2"
        json2 = Job.GetString
        'Log(json2)
        conn3

    Case "Job3"
        json3 = Job.GetString
        'Log(json3)
        If Main.json.Contains("error") OR Main.json="" Then
            toast.Stop
            ToastMessageShow("Check the Apikey & ID" ,True)
        Else
            If json2.Contains("error") OR json2="" Then
                ToastMessageShow("Graph2 is not in use" ,True)
            End If
            If json3.Contains("error") OR json3="" Then
                ToastMessageShow("Graph3 is not in use" ,True)
            End If
            Graph1
        End If
    End If

    Case "JobApkDownload"
        'Log(TAB & "-- JobApkDownload")

        'copy from external to storage card
        Dim out As OutputStream
        out = File.OpenOutput(File.DirDefaultexternal,"tmp.apk",False)
        File.Copy2(Job.GetInputStream, out)
        out.Close
        'Log(out)
        OK_DOWNLOAD = 1
        ToastMessageShow("Starting installation", True)
        InstallApk
    End Select
End If
Job.Release
End Sub

Sub Graph1

    template = File.ReadString(File.DirAssets, "template1.html")

```



```

WebView1.ZoomEnabled = False
Dim options As Map
options.Initialize
CreateColumnChartOptions1(options)
html=template
Dim jg As JSONGenerator
'Dim html As String = template
html = html.Replace("$TYPE$", "Bar" & "Chart")
jg.Initialize(options)
html = html.Replace("$OPTIONS$", jg.ToString)
html = html.Replace("$COLUMN1$", "'string', 'Date'")
html = html.Replace("$COLUMN2$", "'number', 'TotalSumPerDay'")
html = html.Replace("$ROWDATA$", Main.json)
WebView1.LoadHtml(html)
toast.Stop
Graph2
End Sub
Sub Graph2
    template2 = File.ReadString(File.DirAssets, "template2.html")
    WebView2.ZoomEnabled = False
    Dim options As Map
    options.Initialize
    CreateColumnChartOptions2(options)
    html2=template2
    Dim jg As JSONGenerator
    html2 = html2.Replace("$TYPE$", "Pie" & "Chart")
    jg.Initialize(options)
    html2 = html2.Replace("$OPTIONS$", jg.ToString)
    html2 = html2.Replace("$COLUMN1$", "'string', 'ComponentName'")
    html2 = html2.Replace("$COLUMN2$", "'number', 'TotalSum'")
    html2 = html2.Replace("$ROWDATA$", json2)
    html2 = html2.Replace("$GAUGE1$", "'number', 'LABEL'")
    'Dim test As String = Main.json2.IndexOf("Filler 1")
    Dim AGG As String = "Bit 70/100"
    Dim lenght As String = AGG.Length
    If json2.Contains(AGG) Then
        Dim v As Int= json2.IndexOf(AGG)
        'Log(v)
        Dim value As Int= json2.Substring2(v+lenght+2,v+lenght+2+2)
            html2 = html2.Replace("$AGG$",AGG)
            html2 = html2.Replace("$DATA$",value)
    End If
    'Log(html2)
    Graph3
End Sub
Sub Graph3
    template3 = File.ReadString(File.DirAssets, "template3.html")
    WebView3.ZoomEnabled = False
    Dim options As Map
    options.Initialize
    CreateColumnChartOptions3(options)
    html3=template3
    Dim jg As JSONGenerator
    html3 = html3.Replace("$TYPE$", "Table")
    html3 = html3.Replace("$COLUMN1$", "'string', 'Date'")
    html3 = html3.Replace("$COLUMN2$", "'number', 'TotalSum (T)'")
    html3 = html3.Replace("$COLUMN3$", "'string', 'RecipeName'")
    html3 = html3.Replace("$ROWDATA$", json3)
End Sub

Private Sub CreateColumnChartOptions1 (options As Map)
'Dim color As String = "["green"]"
'    options.Put("colors","(yellow)")

```

```

options.Put("title", "Last 7 production days")
Dim v As Map
v.Initialize
v.Put("title", "Date")
Dim h As Map
h.Initialize
h.Put("title", "Mass(T)")
options.Put("vAxis", v)
options.Put("hAxis", h)
options.Put("legend", "none")
End Sub

Private Sub CreateColumnChartOptions2 (options As Map)
options.Put("title", "Last 7 production days")
Dim v As Map
v.Initialize
v.Put("title", "Date")
'options.Put("vAxis", "T")
' options.Put("legend", "none")
options.put("is3D", "true")
End Sub

Private Sub CreateColumnChartOptions3 (options As Map)
options.Put("title", "Total sum per Recipe")
Dim v As Map
v.Initialize
v.Put("title", "Date")
Dim h As Map
h.Initialize
h.Put("title", "Mass(T)")
options.Put("vAxis", v)
options.Put("hAxis", h)
options.Put("legend", "none")
End Sub

Sub Activity_Resume
Graph1
End Sub

Sub Activity_Pause (UserClosed As Boolean)

End Sub

```