OULUN AMMATTIKORKEAKOULU

Jaakko Lohilahti

RÄÄTÄLÖIDYT RATKAISUT TOIMINNANOHJAUSJÄRJESTELMISSÄ

Laajennuksen ohjelmointi moderniin toiminnanohjausjärjestelmään

RÄÄTÄLÖIDYT RATKAISUT TOIMINNANOHJAUSJÄRJESTELMISSÄ

Laajennuksen ohjelmointi moderniin toiminnanohjausjärjestelmään

Jaakko Lohilahti
Opinnäytetyö
Kevät 2021
Tietojenkäsittelyn tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn tutkinta-ohjelma

Tekijä: Jaakko Lohilahti
Opinnäytetyön nimi: Räätälöidyt ratkaisut toiminnanohjausjärjestelmissä
Työn ohjaaja: Teppo Räisänen
Työn valmistumislukukausi ja -vuosi: Kevät 2021 Sivumäärä: 40

Tämän opinnäytetyön tarkoituksena on tutustua yritysten käyttämiin toiminnanohjausjärjestelmiin, sekä niiden laajentamiseen ohjelmoinnin avulla. Jotta järjestelmien toiminnasta saadaan synnytettyä parempi kuva, käydään niiden historiallista kehitystä ensin läpi alkaen 1960-luvulta. Opinnäytetyössä tutustutaan myös siihen, mitä yrityksen on hyvä pitää mielessä nykyaikaista toiminnanohjausjärjestelmää hankkiessa. Teoriaosan lopuksi tutustutaan myös markkinoiden suosituimpiin järjestelmiin ja niiden tarjoamiin ominaisuuksiin.

Teoriaosan lähteinä käytettiin järjestelmien historian osalta Oulun Ammattikorkeakoulun verkkokirjastosta löytyneitä verkkojulkaisuja. Näistä käy hyvin itse toiminnanohjausjärjestelmien kehitys, kuin asiat mitä niitä haluavien yritysten on huomioitava. Nykyaikaisten järjestelmien kohdalla lähteinä käytettiin niin valmistajia, kuin niitä myyviä yrityksiä. Näin saatiin hyvin tuotua esille modernien järjestelmien ominaisuuksia.

Raportissa käydään myös läpi laajennuksen ohjelmointia Microsoft Dynamics 365 Business Centraliin. Tämä tehtiin käyttämällä ohjelmaa varten kehitettyä Application Language-ohjelmointikieltä. Tarkoituksena on antaa kuva siitä, minkälaista julkaistun toiminnanohjausjärjestelmän asiakaslähtöinen kehitys voi olla.

Käytännön ohjelmointiesimerkin raportoinnissa ollaan käytetty askel askeleelta lähestymistapaa. Tarkoituksena on, että tämä antaisi mahdollisesti aiheesta kiinnostuneille hyvän lähtökohdan oman työskentelyn aloittamiselle laajennusten kehitysten parissa. Opinnäytetyö voikin toimia suomenkielisenä materiaalina Application Language-ohjelmointikielelle.

Asiasanat: Tieto- ja viestintätekniikka, tietojenkäsittely, toiminnanohjausjärjestelmä, Business Central, AL-kieli

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems

Author: Jaakko Lohilahti
Title of thesis: Customized solutions in enterprise resource planning software
Supervisor: Teppo Räisänen
Term and year when the thesis was submitted: Spring 2021
Number of pages: 40

The purpose of this thesis is to research the current modern enterprise resource planning systems, and how to develop extensions for them. To give better image on what these systems exactly do for company, we first look at their historic predecessors. The thesis also covers what a company that wants such system should prioritize and be aware of, as buying such system can be very expensive.

In latter half the report goes over of how to develop extension for Microsoft Dynamics 365 Business Central, which is a modern enterprise resource planning system released in 2018. For extension development a programming language called Application Language is used. Currently this language is only used with Business Central. One goal of this thesis is to create Finnish resource on how to begin developing for Business Central with it. This is helped by report formatting which follows step-by-step format with goal of guiding readers through the process.

Keywords: Information technology, enterprise resource planning, Business Central, application language, al-language

SISÄLLYS

1 JOHDANTO

Toiminnanohjausjärjestelmät ja niiden sujuva hallinta ovat usen edellytys sujuvan liiketoiminnan takaamiseksi. Kun yritys kasvaa, muuttuvat hallinnolliset asiat kuten kirjanpito, varaston- ja projektinhallinta sekä huollonhallinta monimutkaisemmaksi. Tämän takia moni yritys pyrkii jo alusta asti parantamaan tehokkuutta erillaisilla toiminnanohjausjärjestelmillä. Tässä opinnäytetyössä pyritään löytämään vastaukset järjestelmistä kiinnostuneiden yrityksien mahdollisiin kysymyksiin liittyen toiminnanohjausjärjestelmien käyttöönottoon.

Jotta modernien järjestelmien hyödyllisyyden voi täysin ymmärtää, on hyvä tietää kuinka ne ovat historiallisesti kehittyneet. Tässä vaiheessa on hyvä tutustua menneiden vuosikymmenten edeltäjiin aina 60-luvulle asti, jolloin nopea teollistuminen vaati parempaa tapoja organisoida toimintaansa tehokkaammaksi.

Nykyaikaiset toiminnanohjausjärjestelmät ovat laajoja kokonaisuuksia, jotka sisältävät monia yrityksen tarpeita helpottavia toimintoja, eivät ne aina taivu erikoisimpiin tapauksiin. Yrityksen toimiala, tai heidän tapansa tehdä toimintaa saattaa vaatia lisäominaisuuksien ohjelmointia järjestelmiin. Modulaarisiksi tarkoitetut modernit toiminnanohjausjärjestelmät kuitenkin hyvin tukevat tätä, ja ohjelmistojen valmistajat ymmärtävät helpon muokattavuuden lisäarvon. Raportin neljännessä ja viidennessä luvussa tutustutaan tällaisen yksinkertaisen asiakaskohtaisen ohjelmistolaajennuksen luontiin Microsoft Dynamics 365 Business Centraliin.

Business Centralin laajennuksen kehitystä varten käytetään Microsoftin luomaa AL-kieltä (eng.Application Language), joka on ohjelmiston hallintaa varta vasten kehitetty ohjelmointikieli. AL-kielen suhteellinen tuoreus, kuin myös suomenkielisen dokumentaation puute on myös yksi avaintekijä opinnäytetyön aiheen valinnassa.

2 TOIMINNANOHJAUSJÄRJESTELMÄT

Kun yrityksen koko kasvaa, muuttuu myös sen organisointikyky vaikeammaksi. Yritysten valmius hallita suuria datamääriä, jotka voi sisältää henkilö-, tuotanto- ja inventaariodataa voi muuttua mahdottomaksi, ellei niitä pysty keskittämään helposti hallittavaan muotoon. Tässä kohtaa tulevat esille toiminnanohjausjärjestelmät (eng.enterprise resource planning). Nykyaikaisten toiminnanohjausjärjestelmien mahdollistamien integraatioiden avulla yritykset pystyvät niin pitämään kirjaa, kuin koordinoimaan liiketoimintaansa tehokkaasti. Ohjelmistosta riippuen on myös mahdollista järjestää dataa erillaisiksi taulukoiksi, mitä voi käyttää mahdollisten trendien seuraamisen ja muodostamiseen. Tämä auttaa yrityksen johtoa tehokkaammassa päätöksenteossa.

Tekniikan kehittyessä 80- ja 90-luvulla monet isot yritykset palkkasivat tuhansia järjestelmä-analyytikkoja, jotka kehittivät lukemattomia uusia ohjelmistoja tehdäkseen asioita tehokkaammin. Yhdistettynä henkilökohtaisten tietokoneiden yleistymiseen tämä johti siihen, että melkein jokaisella yrityksen sisäisellä osastolla oli omat järjestelmänsä. (Nestell & Olson 2018, 1)

Kun ensimmäiset ohjelmistot tulivat markkinoille, olivat niitä tarjoavat yritykset harvassa. Vaihtoehtoisten menetelmien puute merkitsi myös korkeita hintoja, mikä tarkoitti että monella toiminnanohjausjärjestelmää haluavalla yrityksellä ei välttämättä ollut varaa niihin. (Bendoly & Jacobs 2003, 238) Tästä syystä ensimmäiset ohjelmistot olivat lähes poikkeuksetta isojen yritysten käyttöön suunnattuja, eikä pienille- ja keskisuurille yrityksille ollut vielä tarjontaa.

Yleinen syy yritykselle hankkia toiminnanohjausjärjestelmä on saada poljettua kustannuksia pitkässä juoksussa. Ideana on saada yksi yksinkertaistettu tapa hallita asioita yrityksen sisäisesti, sekä vähennettyä tarvittavaa teknisen tuen määrää. Käytännössä kustannussäästöt ei aina tapahdu. Tämä voi johtua odottamattomista yksityiskohdista koskien järjestelmiä, mitkä äkkiä luo piilokustannuksia. Yritys ei esimerkiksi panosta tarpeeksi henkilöstön kouluttamiseen toiminnanohjausjärjestelmän käytössä, mikä järjestelmän tuotantoonotossa voi laskea henkilöstön tuottavuutta. (Nestell 2018, 4)

2.1 Historia

Itse toiminnanohjausjärjestelmien historian voidaan katsoa alkaneen materiaalivaatimusten suunnittelusta (eng.material requirements planning) 1960-luvulla. Tuolloin rakennuskoneita valmistanut J.I. Case teki yhdessä IBM:n kanssa ensimmäisen MRP-järjestelmän. Myöhemmin useat isot valmistajat kehittivät järjestelmiä itselleen. Vaikka järjestelmät olivat kalliita tehdä ja veivät sen aikaisella teknologialla paljon tilaa, auttoivat ne tuottajia paremmin organisoimaan materiaalien hankkimista ja sen kiertokulkua tehtaassa. Näin pystyttiin tehostamaan tehtaiden tuotantolinjastoja. (McCue 2020, Viitattu 20.5.2021)

Kirjassaan Enterprise Resource Planning and Management Information Systems, Murthy kertoo tärkeimmät hyödyt onnistuneessa MRP-järjestelmän käyttöönotossa ovat seuraavanlaiset. (Murthy 2008, 16):

- Huomattava, jopa 20-40% lasku inventaariomäärissä ja sen mukana tuomana kustannuksissa.
- Vähemmän materiaalipulaa mikä aiheutti tuotannon keskeytyksiä ja vaati resursseja linjastojen aikataulutuksissa.
- Parempi asiakaspalvelu, sekä paremmat mahdollisuudet pysyä aikataulussa ja luvata aikaisempia toimitusaikoja.

MRP-ohjelmistojen seuraajana voidaan pitää tuotantovaatimusten suunnitteluun (eng. manufacturing resource planning) suunnattuja ohjelmisto. Nämä järjestelmät, joita nimitettiin MRP II-lyhenteellä, olivat iso harppaus kohti nykyisiä järjestelmiä ja sisälsivät monia ominaisuuksia mitä moderneissakin toiminnanohjausjärjestelmissä on. Siinä missä MRP keskittyi vain raakamateriaalin ja linjastojen hallintaan, oli MRP II-järjestelmissä mukana tuotteiden jakelun organisointi, projektin- ja henkilöstönhallinta sekä talouden kirjanpito. (Hossain & Patrick & Rashid 2002, 4)

Vaikka näissä edeltävissä järjestelmissä oli paljon tämänhetkisten toiminnanohjausjärjestelmien ominaisuuksia, rajoittivat niiden käyttöönottokustannukset käytön isoimmille yrityksille. Kuten MRP-järjestelmien ensimmäisistä ominaisuuksista voi päätellä, olivat järjestelmien ensimmäiset kohderyhmät enimmäkseen tuotteita valmistavat tehtaat. Toimialasta ja yrityksen koosta riippuen tämä merkitsi sitä, että osalla yrityksistä ei ollut mahdollisuutta käyttöönottaa MRP-järjestelmiä. 90-

luvun alussa tulivat markkinoille ensimmäiset varsinaiset ohjelmistot, joita nimitettiin toiminnanohjausjärjestelmiksi. Tämä tulisi myös parantamaan mahdollisuuksia ohjelmistojen hankintaan muillekkin yrityksille.

2.2 Modernit järjestelmät

Suuri harppaus nykyaikaisissa toiminnanohjausjärjestelmissä on niiden toimiminen pilvipalvelimilla. Vanhat järjestelmät saattoivat vaatia yrityksen koosta ja toimialasta riippuen monia eri sisäisiä palvelimia, mitkä nostivat hankinta- ja ylläpitökustannuksia. Pilvessä toimiminen on myös tietoturvan kannalta parempi järjestelmille ja niiden varastoimalle datalle. Kertahankintaisen järjestelmän sijaan ja pilvipalveluita hyödyntäen, moderneja ERP-järjestelmiä myydääkin Software-as-a-Service-periaatteella. Myös toiminnan skaalaaminen on yksi pilven etu, yrityksen voi tarpeen mukaan hankkia lisää palvelinkaistaa, sekä luopua ylimääräisesti mitä se ei näe enää tarpeelliseksi. (Needleman 2019, Viitattu 20.5.2021)

Modernien toiminnanohjausjärjestelmien modulaarinen lähestymistapa on antanut yrityksille enemmän vaihtoehtoja. Kaikki yritykset eivät tarvitse ohjelmiston kaikkia ominaisuuksia, mutta melkein jokaisella on tarve talouden ja kirjanpidon moduulille (Nestell & Olson 2018.) Tämän modulaarisuuden ansiosta suosituimmissa järjestelmissä on mahdollisuus saada myös asiakkuudenhallinta, mitkä ennen oli erillisinä järjestelminä, integroitua saman ohjelmiston alle.

Toiminnanohjausjärjestelmien kehitys on kuitenkin alati jatkuvaa. Netsuite-järjestelmää kehittävä Oracle on listannut seuraavat asiat ERP-järjestelmien tulevaisuuden trendeinä. (Luther 2020, Viitattu 20.5.2021):

- Pilvipalvelut. Historiallisesti yritykset ovat pitäneet ERP-järjestelmän yrityksen omissa järjestelmissä, mutta jo tämänhetkiset ohjelmistot pystyvät toimimaan pilvessä, ja näin sen käyttö on vain yleistymään päin.
- Kaksitasoinen toiminnanohjausjärjestelmä. Vaikka ERP-järjestelmän tavoite saada integroitua kaikki yritysdata yhden ohjelmiston alle, voi siitä olla myös haittaa varsinkin suurissa monikansallisissa yhtiöissä, jotka voivat vaatia paikallistasolla enemmän ketteryyttä.
- Mobiilijärjestelmien käyttö. Historiallisesti toiminnanohjausjärjestelmien käyttöliittymät eivät ole olleet käyttäystävällisemmästä päästä. Modernien järjestelmien kohdalla tähän

on kuitenkin panostettu, mikä myös auttaa käyttöliittymien skaalautuvuudessa ja käytettävyydessä kosketusnäytön omaavissa mobiilijärjestelmissä.

- Koneoppiminen. Oracle näkee enenevissä määrin tekoälyn käytön prosessien automatisoinnissa, mikä ennaltaan tehostaisi yrityksen toimintaa.
- Järjestelmien räätälöitävyys. Historiallisesti monet järjestelmä olivat varsin monimutkaisia ja vaikeita muokata vastaamaan yrityksien erikoistarpeita. Modernien ERP-järjestelmien kohdalla tämä on helpottunut ja toimittajien interesseissä onkin saada mahdollisimman helposti lähestyttäviä ohjelmistoja asiakkaille.

3 OHJELMISTON VALINTA

Toiminnanohjausjärjestelmän, joko käyttöön ottaminen tai vanhasta siirtyminen on yritykselle iso harppaus eteenpäin. Kaikki käyttöönottoon liittyvät yksityiskohdat eivät kuitenkaan ole alussa yrityksen toimeenpaneville henkilöstölle selviä, mikä voi johtaa epäröintiin järjestelmiä kohtaan. Ohjelmistojen määrän tarjoama valinnanvara voi itsessään yksi tekijä, yrityksen johto tietenkin haluaa valita juuri oikean ohjelmiston, ja niiden määrä voi aiheuttaa epävarmuutta. Microsoft, joka tarjoaa Dynamics 365 Business Central-toiminnanohjausjärjestelmää mainitsee verkkosivuillaan kolme yleisintä epävarmuustekijää ohjelmiston käyttöönotosta. (Microsoft 2021, Viitattu 20.5.2021):

- Epävarmuus oikean ohjelman valinnasta.
- Epävarmuus toiminnanohjausjärjestelmän käyttöönoton vaatimista kustannuksista yritykselle.
- Yritys ei halua menettää olemassa olevia ja toimiviksi havaittuja järjestelmiä.

Microsoftin mukaan yritys ei tarvitse täydellistä ratkaisua ohjelmiston valinnassa. Tähän vaikuttaa modernien toiminnanohjausjärjestelmien kyky hallita dataa monista eri lähteistä ja tuoda ne yhden järjestelmän alle. Tämä voi sisältää niin jo yrityksen sisäisessä käytössä olevat ohjelmistot, kuin ulkopuoliset järjestelmät. Näihin ERP-järjestelmä voi olla yhteydessä vaikka API-rajapintoja käyttäen.

Ohjelmistopaketin valintaan kuuluu olennaisesti myös sen hinta. Yksi tekijä tähän on pakettia ostavan yrityksen koko, milloin hintaan voi vaikuttaa yrityksen sisällä sitä käyttävien henkilöiden määrä. Toiminnanohjausjärjestelmiä kauppaavat yritykset myyvät tuotetta yrityslisenssillä, mihin vaikuttavat käyttäjien määrä. Tämän lisäksi koska yleisesti ottaen modernit ERP-järjestelmät toimivat pilvessä, ovat nämä lisenssimaksut juoksevia, ja hyvä ottaa huomioon järjestelmää valittaessa.

Käyttöönoton kuluihin ei pelkästään vaikuta itse ohjelmiston lisenssit. Yritysohjelmistoja, ja niitä koskevia informaatioita tilastoivan betterbuysin (Better Buys 2021, Viitattu 20.5.2021) mukaan seuraavan listan asiat ovat avaintekijöinä hinnan muodostumisessa. Näitä ovat:

- Myytävät lisämoduulit jotka eivät kuulu peruspakettiin, mutta ovat olennaisia yrityksen toiminnan kannalta.

- Harva toiminnanohjausjärjestelmä sopii suoraan yrityksen tarpeisiin. Tämä voi aiheuttaa lisäkustannuksia, kun ohjelmistoon tehdään lisää ominaisuuksia.
- Varsinkin isoimpien järjestelmien kohdalla käytetään yleensä kolmannen osapuolen yrityksiä, jotka kauppaavat ja käyttävät omia konsultteja toiminnanohjausjärjestelmän käyttöönotossa.
- Mikäli asiakas haluaa pilviversion sijaan paikallisessa verkossa toimivan järjestelmän, tuovat sen vaatima laitteisto lisäkustannuksia.

3.1 Yrityksen tarpeiden kartoitus

Kun toiminnanohjausjärjestelmän käyttöönotto ei ole yksinkertainen asennus, vaan jopa kuukausia kestävä prosessi, on yritysten mietittävä tarkkaan mitä lisäarvoa he haluaisivat ohjelmiston antavan heille. Tällaisen tilanteen kartoittamisessa on hyvä katsoa yrityksen toiminnan kipukohtia, ja miettiä mitkä ovat prosesseja jotka vievät liikaa aikaa.

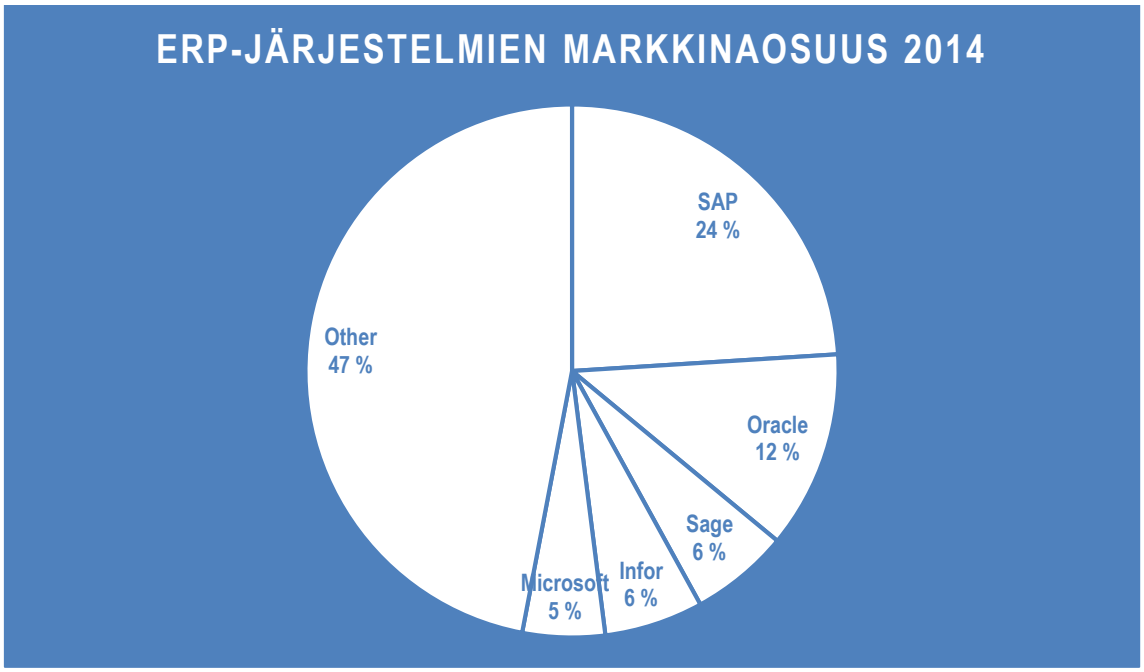
Helposti yritys voi nähdä mikäli eri osastoilla on vaikea tehdä yhteistyötä keskenään. Esimerkiksi kasvun myötä nousevan informaation määrää voi olla liki mahdoton pitää kirjaa osastojen kesken, ellei käytössä ole järjestelmää, joka keskittäisi kaiken datan yhden tietokannan alle.

Pelkästään nykyaikaisten toiminnanohjausjärjestelmien toimiminen pilvipalveluissa voi olla iso vaikuttava tekijä. Tämän avulla avainhenkilöt pystyvät mistä tahansa pääsemään käsiksi yrityksen tietokantoihin. Tämä on mahdollista vaikka organisaatio olisi päättänyt hankkia paikallisesti yrityksen omalle laitteistolle asennetun ohjelmiston. (ERP News 2016, Viitattu 20.5.2021)

Vaihtoehtoisesti yrityksellä voi olla varteenotettavia syitä olla hankkimatta toiminnanohjausjärjestelmää. Järjestelmä voi esimerkiksi olla liian monimutkainen sen tuomiin hyötyihin nähden, mikä itsessään nostaa kuluja koulutuksen tarpeeseen. Mikäli yrityksellä ei toimialan takia ole tarvetta kuin parille ERP-järjestelmän tarjoamalle moduulille, kuten henkilöstön ja kirjanpidon hallinnalle, on järkevämpää suunnata katseet kevyempiin ratkaisuihin. (Simon, Viitattu 20.5.2021)

3.2 Markkinoilla olevat järjestelmät

Kun yritys on lopulta kartoittanut kipupisteensä ja päättänyt hankkia toiminnanohjausjärjestelmän, tulee seuraavaksi eteen itse sopivan järjestelmän valinta. Tähänkin yrityksen koko ja toimiala vaikuttaa merkittävästi. Vaikka isoimpien järjestelmien valmistajat pystyvät tarjoamaan kokonaisvaltaisempia kokonaisuuksia, ovat nämä usein myös tarkoitettu suuren liikevaihdon ja datamäärän omaaville asiakkaille. Tämä ei silti ole iso este, sillä markkinoilla olevien järjestelmien määrä on kasvanut merkittävästi, ja myös pienemmät yritykset pystyvät tarjoamaan toimivia ratkaisuja. Tämä ohjelmistojen tarjonnan pirstaloituminen käy ilmi Gartnerin vuosittaisesta ERP-järjestelmien markkinatutkimuksesta, jossa vuonna 2018 viisi isointa järjestelmää piti hallussaan 51% markkinoilla olevista järjestelmistä. (Wilson 2019, Viitattu 20.5.2021) Alempana Kuviossa 1 näkyy myös vuoden 2014 tilanne eri järjestelmien kesken.



KUVIO 1. Havainnekuva suosituimpien järjestelmien markkinaosuuksista vuonna 2014 (PaperFree 2020, Viitattu 20.5.2021)

On myös hyvä ottaa huomioon , että varsinkin isoimpien järjestelmien kohdalla yritys ei välttämättä voi edes suoraan hankkia järjestelmää valmistajalta, vaan pitää ostaa se jälleenmyyvän yrityksen kautta. Tämä voi aiheuttaa tilanteen, missä järjestelmän lisäksi yrityksen pitää etsiä sopiva yhteistyöyritys järjestelmän käyttöönottoon. Koska ERP-järjestelmiä jälleenmyyvät yritykset tekevät myös käyttöönoton, ja useinmiten hoitavat tukipalvelut järjestelmän ollessa pystyssä, on

jälleenmyyjillä myös omat intressit asiakassuhteissa. Osa jälleenmyyjistä voi keskittyä asiakassuhteissaan isoihin yrityksiin, kun taas osa panostaa pieniin- ja keskisuuriin organisaatioihin.

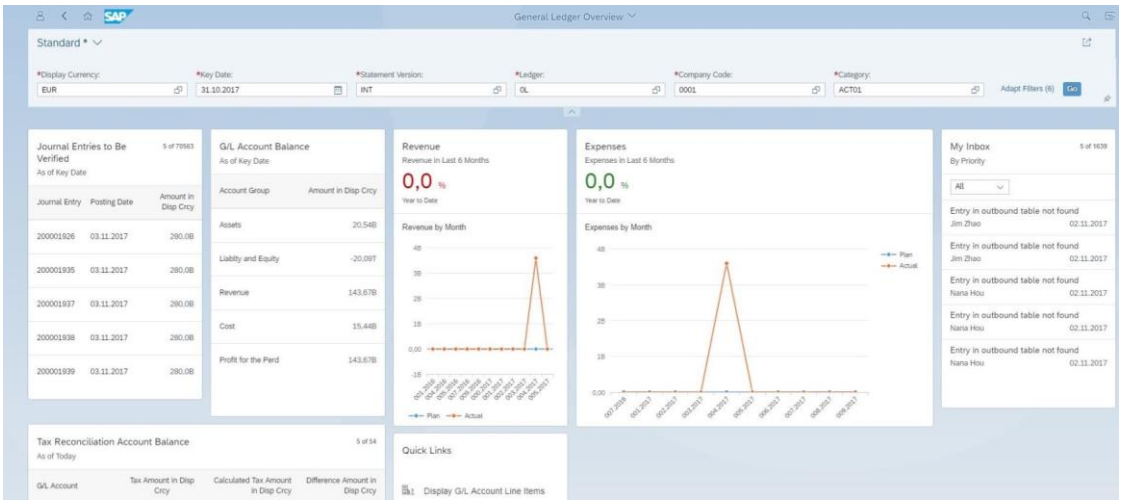
3.2.1 SAP

Vuonna 1972 perustettu Saksalainen SAP on yksi isoimmista toiminnanohjausjärjestelmiä toimittavista yrityksistä, ja isoin Eurooppalainen ohjelmistotalo. Sen perustivat viisi entistä IBM:n työntekijää. SAP oli alusta asti perustettu palvelemaan yrityksiä ja prosessoimaan niiden tuottamaa dataa, ja tämän myötä syntyi heidän ensimmäinen kirjanpitojärjestelmä RF. RF toimi kivijalkana joka moduuleiden määrän kasvaessa tuli tunnetuksi nimellä SAP R/1. Tämä ohjelmistojen nimeämiskäytäntö on käytössä myös heidän nykyisessä lippulaivajärjestelmässä. (SAP 2021, Viitattu 20.5.2021)

SAP:n tuorein toiminnanohjausjärjestelmä on nimeltään SAP S/4HANA. Se on lyhenne nimestä **SAP** Business **Suite 4**/SAP **HANA**. SAP S/4HANA on yksi suosituimmista järjestelmistä, ja on käytössä yrityksillä kuten Walmart, Apple ja Ford Motor (Markovski & Micik & Pang 2020, Viitattu 20.5.2021). SAP kertoo kotisivuillaan ohjelmiston keskeisiksi ominaisuuksiksi seuraavat: (SAP 2021, Viitattu 20.5.2021)

- Sisäänrakennettu tekoälyanalytiikka ja prosessien automatisointi.
- RAM-muistissa toimiva tietokanta ja yksinkertaistettu tietomallinnus.
- Kuluttajaläheinen käyttäjäkokemus.
- Toiminnallisuuksia monille eri aloille.

Ohjelmiston voi kuvitella koostuvan kahdesta erillisestä osasta, joista jälkimmäinen, SAP HANA osa toimii tietokoneen muistiin tallennettuna tietokantana massamuistin sijaan. Koska yrityksen tietokantakyselyiden ei tarvitse tapahtua kiintolevyiltä, nopeuttaa tämä huomattavasti suurien datamäärien käsittelyä. (Anikin 2016, Viitattu 20.5.2021) Omana osanaan SAP HANA voi toimia erillisenä tietokantana monille muillekkin ohjelmistoille, kuin SAP:n omille ERP-järjestelmille. SAP S voidaan ajatella olevan itse front-puoli HANA:n tietokannalle, tämän osion voi nähdä kuviossa 2.



KUVIO 2. Ruutukaappaus SAP S/4HANA:n pääkirjanpidon koontisivusta.

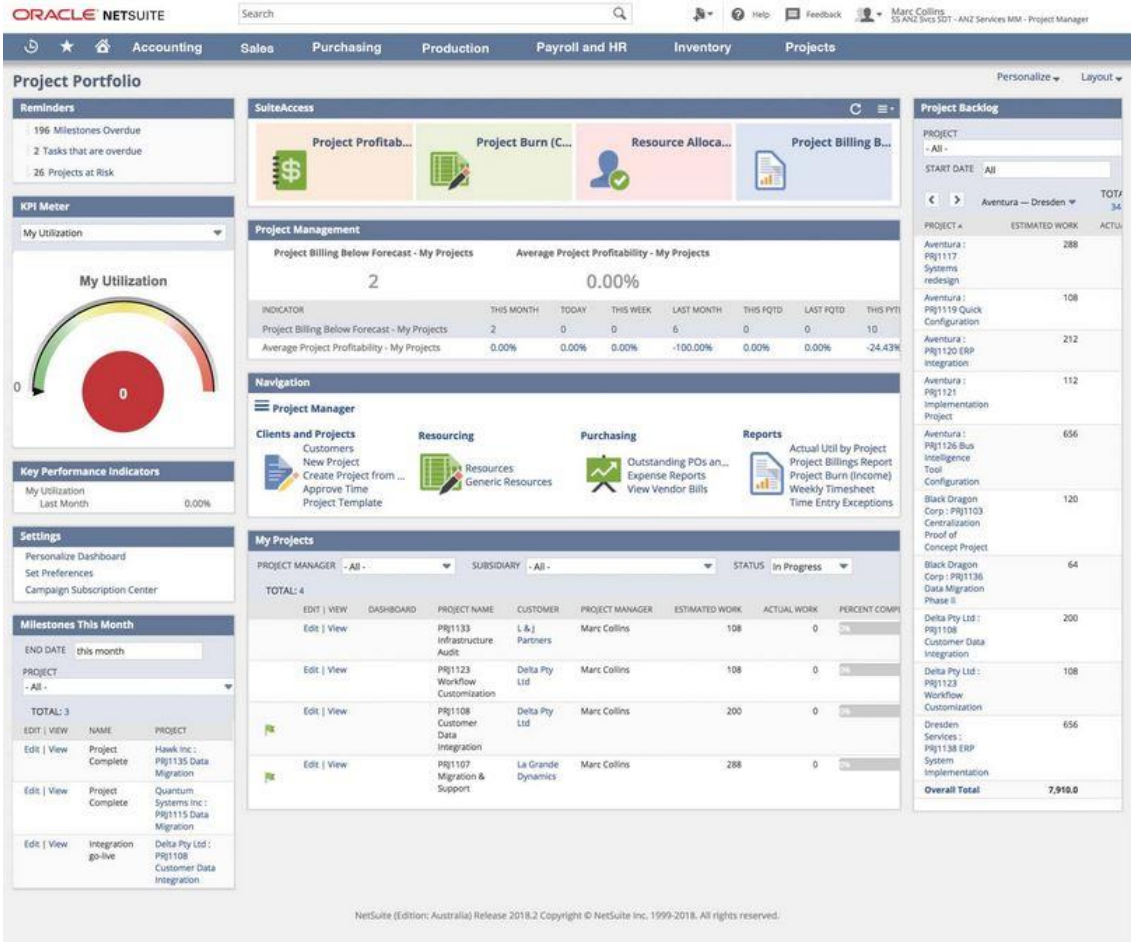
Riippuen yrityksen tilanteesta, seurataan SAP:n käyttöönotossa eri menetelmiä. Karkeasti sanottuna nämä jakautuvat sen perusteella onko yrityksellä jo aikaisempi SAP versio, vai ovatko he uusia käyttäviä. (Ferrari 2019, Viitattu 20.5.2021)

- Greenfield – Yritys on uusia SAP käyttäjä, tai he haluavat vain aloittaa puhtaalta pöydältä toiminnanohjausjärjestelmän kanssa. Tässä menetelmässä ei tuoda historiallisia yritystietoja uuteen järjestelmään.
- Brownfield – Yritys haluaa konvertoida aikaisemman SAP järjestelmän uusimpaan versioon. Tämä pitää sisällään myös historiallisen yritysdatan tuomisen uuteen järjestelmään.
- Bluefield – Tässä metodissa yhdistyy Green- ja Brownfieldin kantavat teema. Yritys ottaa järjestelmän käyttöön historiallisen datan kanssa, mutta samalla ottaa uuden suunnan itse järjestelmän asetuksissa. Tämä hyödyttää eritoten silloin jos asiakas on aikaisemmasta versiosta päivittävä, mutta haluaa myös uudistaa toimintatapojansa järjestelmäpäivityksen lisäksi.

3.2.2 Oracle Netsuite

Netsuite (kuvio 3) on Oraclen kehittämä toiminnanohjausjärjestelmä pienille- ja keskisuurille yrityksille. Alunperin omana yrityksenä toiminut Netsuite alkoi kehittämään järjestelmää vuonna 1998. Netsuite oli ensimmäinen verkossa toimiva ERP-järjestelmä, ja sitä myös ensimmäisenä varsinaisena pilvipalveluna. Vuonna 2016 Oracle, jolla oli jo omastakin takaa toiminnanohjausjärjestelmä, yhdistyi Netsuiten kanssa. (Kodella 2020, Viitattu 20.5.2021)

Netsuiten hinnoittelu on hyvin asiakaskohtaista. Hinta koostuu käyttäjämääristä, yrityksen haluamista moduuleista, käyttöönotosta sekä lisenssistä. Itse ohjelmiston peruslisenssillä maksaa 999 dollaria kuukaudessa, ja käyttäjäkohtainen lisenssi 99 dollaria. (Betterbuys 2021, Viitattu 20.5.2021)



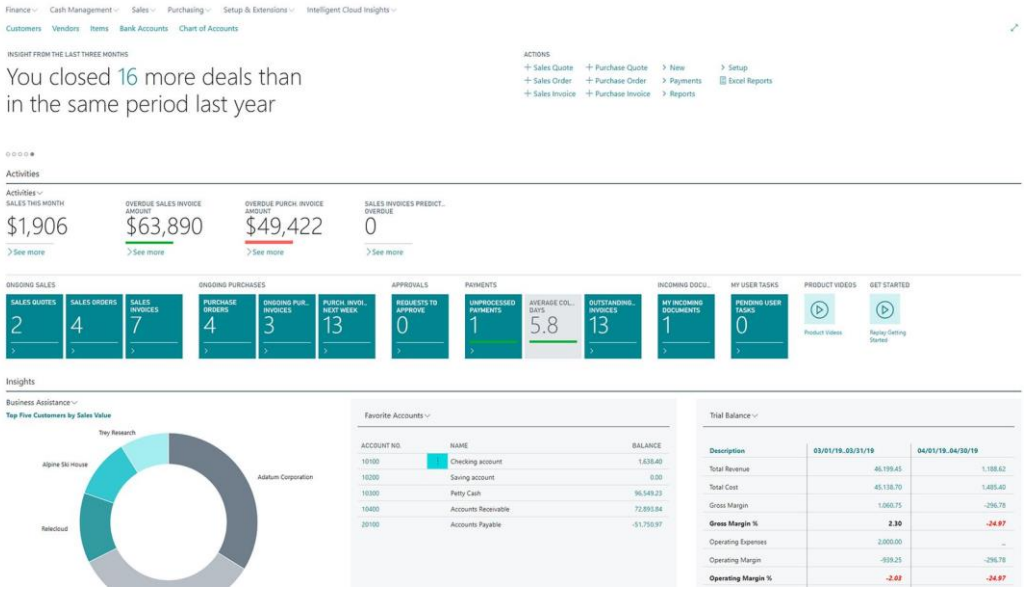
KUVIO 3. Ruutukaappaus Netsuiten projektinhallinnan koontisivusta.

3.2.3 Microsoft Dynamics 365 Business Central

Microsoft Dynamics 365 Business Central on uusin versio Microsoftin pienielle- ja keskisuurille yrityksille suunnatusta toiminnanohjausjärjestelmästä, joka ennen 2018 tunnettiin nimellä Dynamics Nav. Business Centralin keskeisiin ominaisuuksiin kuuluu sen mahdollisuudet integroida Microsoftin muita paljon yritysmaailmassa käytettyjä ohjelmistoja, kuten nimestä pääteltävä Office 365. Myös datan analysointiin käytettävää Power Bi:ä on mahdollista linkittää suoraan Business Centralin yritysdataa sisältäviin tietokantoihin, mikä virtaviivaistaa yritystoiminnan raportointia.

(Duilio & Demiliani 2018, Viitattu 20.5.2021) Business Central tarjoaa käyttäjälle selkeästi luettavan käyttöliittymän (kuvio 4), josta nopealla silmäyksellä on luettavissa tärkeimmät yrityksen toimintaa koskevat tiedot.

Kuten muutkin modernit järjestelmät, myös Business Central on muokattavissa asiakasyrityksen tarpeisiin. Tähän käytetään Microsoftin kehittämää **AL-Language** kieltä, jota tällä hetkellä käytetään vain Business Centralin laajennusten kehityksessä. Tätä hyödyntäen voidaan ohjelmistoa laajentaa esimerkiksi uusia kenttiä tai sivuja lisäämällä.



KUVIO 4. Ruutukaappaus D365 Business Centralin roolikeskuksena kutsutusta etusivusta.

Business Centralin hinnoittelu määräytyy monesta asiasta, joista tärkeimpinä on käyttäjien määrä ja heidän käyttäjätaso. Vaikka yrityksellä olisi monta järjestelmää käyttävää työntekijää, eivät kaikki välttämättä tarvitse kalleinta lisenssiä. Alla olevaan taulukkoon 1 on koottu kolme eri käyttäjälisenssiä heidän kuukausittaisella hinnalla. On myös hyvä muistaa, että alla mainitut hinnat sisältävät vain ohjelmiston lisenssoinin. Ne eivät sisällä varsinaisen käyttöönottoprosessin kuluja, joka toteutetaan Microsoftin kanssa yhteistyötä tekevän yrityksen kanssa.

TAULUKKO 1. Dynamics 365 Business Central-lisenssien hinnat (Microsoft 2021, Viitattu 20.5.2021)

Dynamics 365 Business Central Essentials	Dynamics 365 Business Central Premium	Dynamics 365 Business Central Team Members
59e/käyttäjä/kk	84,30e/käyttäjä/kk	6,70e/käyttäjä/kuukausi
Sisältää kaikki moduulit paitsi tuotannon- ja huollonhallinta.	Sisältää kaikki moduulit.	Lisenssi käyttäjälle, joka työskentelee vain yhden osa-alueen parissa. Esim. laskutus

4 KEHITYSYMPÄRISTÖN PYSTYTYS

Usein toiminnanohjausjärjestelmää käyttöönottaessa tulee tilanne, jossa asiakkaalla on tarve toiminnallisuudelle mitä ohjelmistossa ei ennalta ole. Tämä voi johtua yrityksen toimialasta, tai tavasta pitää kirjaa tiedoista tavalla, johon toiminnanohjausjärjestelmässä ei ole valmiiksi toiminnallisuutta. Tässä luvussa käydään läpi miten Microsoft Dynamics 365 Business Central järjestelmään kehitetään laajennus käyttäen AL-Language ohjelmointikieltä, sekä kuinka alussa pystytetään toimivat kehitysympäristöt.

Laajennus tulee sisältämään uuden sivun luonnin Business Centraliin, joka sisältää sen omia varta vasten luotuja kenttiä. Näitä kenttiä varten tarvitaan myös tietokantataulut pitämään dataa, jotka luodaan myös. Laajennuksessa käydään myös läpi miten siinä luodut tietueet saadaan keskustelemaan yhdessä Business Centralin omien tietokantojen kanssa. Lopuksi käydään vielä läpi miten sivu, ja sen sisältämät tiedot voidaan koota tulostettavaksi raportiksi.

4.1 Kehitysympäristöt

Jotta laajennusta pystyy kehittämään turvallisesti, on hyvä pitää kehitystyö erillään tuotannossa olevasta järjestelmästä. Tällä estetään mahdollisten bugien vaikutukset järjestelmän käyttöön sekä estetään tilanteita, jossa pahimmillaan ei voi toteuttaa liiketoimintaa järjestelmän avulla. Käyttöönottoprojektin aikana, jolloin suurin osa laajennuksista tehdään tällä ei ole niin suurta merkitystä, koska tuotantoympäristö ei ole vielä varsinaisessa käytössä. On kuitenkin hyvien toimintatapojen mukaista käyttää järjestelmän tarjoajan vartavasten tekemiä ympäristöjä kehitykseen.

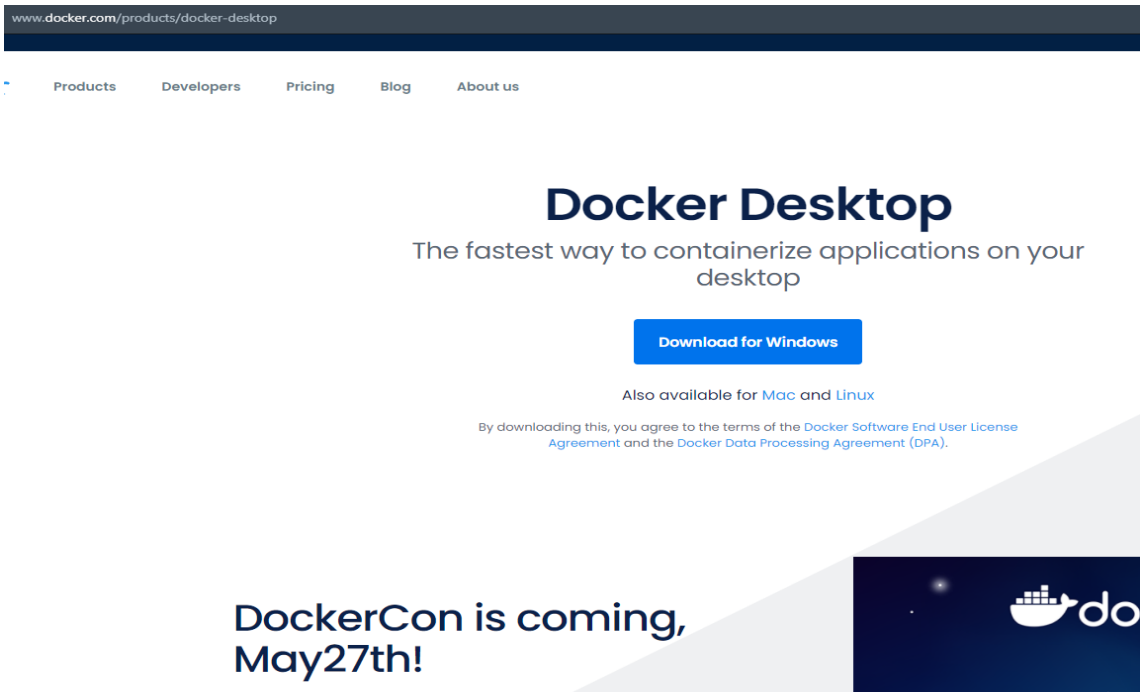
Business Centralin kohdalla Microsoft tarjoaa varsinaisen tuotantoympäristön lisäksi kaksi erillistä sandbox-ympäristöä. Nämä ovat erillisiä tuotantoympäristöstä kopioituja ympäristöjä, jossa tehdyt toimet eivät vaikuta muihin. Sandbox-ympäristössä on myös rahaliikenteeseen liittyvät toiminnallisuudet pois käytöstä. Tämän takia ne soveltuvatkin hyvin järjestelmän esittelyyn, sen koulutuskäyttöön sekä ohjelmoidun laajennuksen testaukseen. Sandbox-ympäristöjä voi käyttäjän mielen mukaan poistaa käytöstä, ja ottaa uudelleen käyttöön. Yleensä näin tehdään jotta saadaan

tuore kopio tuotantoympäristöstä, tai palauttaakseen ympäristö alkuasetuksiin. Haittapuolena näissä pilvessä toimivissa järjestelmissä on niiden rajallinnen määrä, sekä niiden vaatima ohjelmistolisenssi.

Toinen tapa saada kehitysympäristö käyttöön on paikallinen virtualisointi. Tässä tavassa käytetään Docker-nimistä ohjelmaa ylläpitämään paikallista asennusta Business Centralista Microsoftin tarjoaman levykuvan avulla. Tämä vaihtoehto on ilmainen ja antaa kehittäjälle myös mahdollisuuden työskennellä ilman internet-yhteyttä. Dockerin kanssa työskennellessä on kuitenkin hyvä muistaa sen vaatima keskusmuistin määrä virtualisoinnin pyörittämiseksi. Business Centralin kohdalla on suositeltavaa varata vähintään kuusi gigatavua sujuvaan käyttöön. Kokonaisuudessaan kehitykseen käytettävästä tietokoneesta olisi hyvä löytyä 16 gigatavua keskusmuistia.

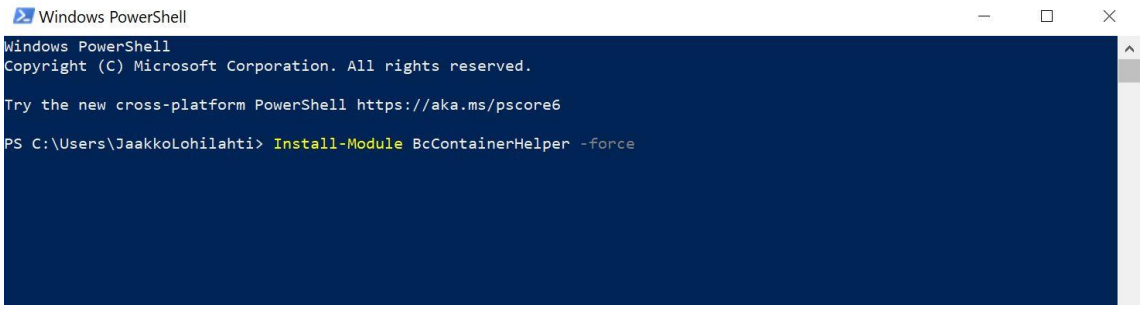
4.1.1 Docker

Ensimmäisenä asiana ennen itse laajennuksen ohjelmointia on paikallisen Business Central-ympäristön pystytys. Jotta Dockeriin saadaan Business Centralin levykuva, tulee se konfiguroida PowerShell komennolla. Tätä ennen tulee kuitenkin itse Docker olla asennettuna tietokoneella. Dockerin voi ladata heidän verkkosivuiltaan osoitteesta www.docker.com (kuvio 5).



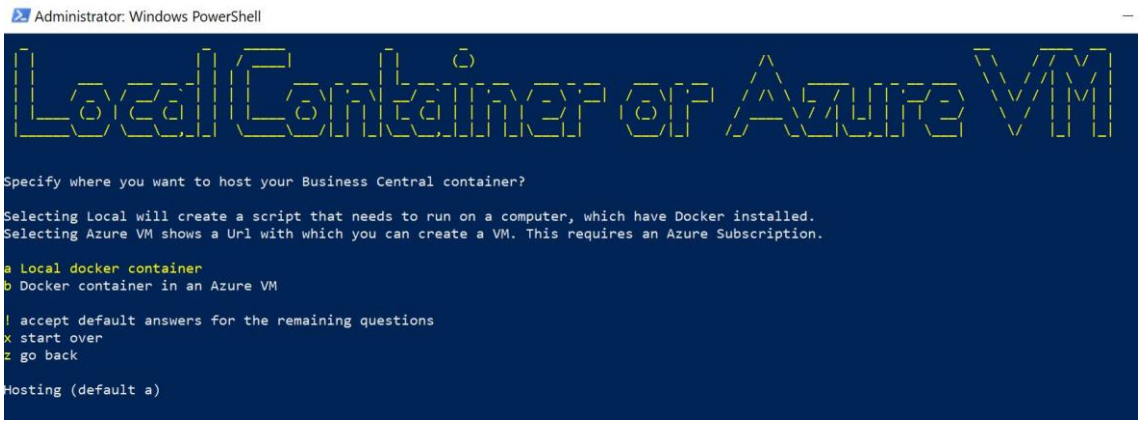
KUVIO 5. Ruutukaappaus Dockerin lataussivulta.

Kun Docker on asennettu on aika asentaa BcContainerHelper niminen laajennus PowerShelliin. Tämä mahdollistaa muokatun Business Central-levykuvakkeen lataamisen Microsoftilta, ja konfiguroinnin suoraan Dockeriin. Kuten kuviossa 6 on esimerkkinä, tulee ensin avata Windows PowerShell järjestelmänvalvojana, ja antaa komento ”**Install-Module BcContainerHelper -force**”. Tämä asentaa uusimman version BcContainerHelperistä, missä voi mennä muutama minuutti. Mikäli PowerShell palauttaa virheen, jossa kerrotaan komennon olevan tuntematon parametsi, tulee PowerShell ensin päivittää ”**Install-Module -Name PowerShellGet -Repository PSGallery -Force**”-komennolla.



KUVIO 6. Ruutukaappaus Windows PowerShellistä.

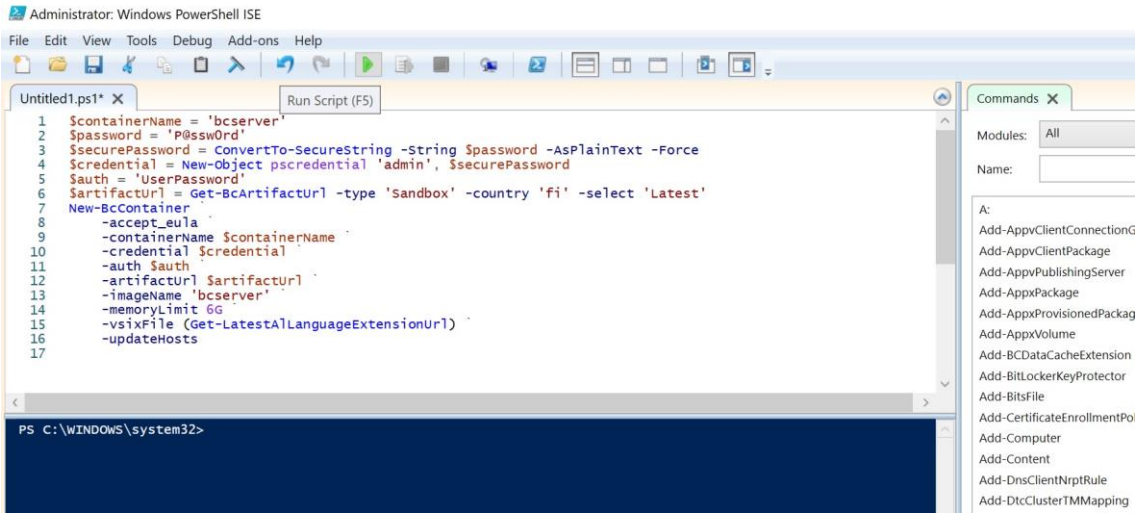
Kun BcContainerHelper on asennettu, voidaan viimeiseksi antaa komento ”**new-BcContainerWizard**”. Tämä aukaisee uuden ikkunan, missä määritellään Business Central levykuvakkeen asetukset, kuten ohjelman kieli ja versio.



KUVIO 7. Ruutukaappaus Windows PowerShellistä.

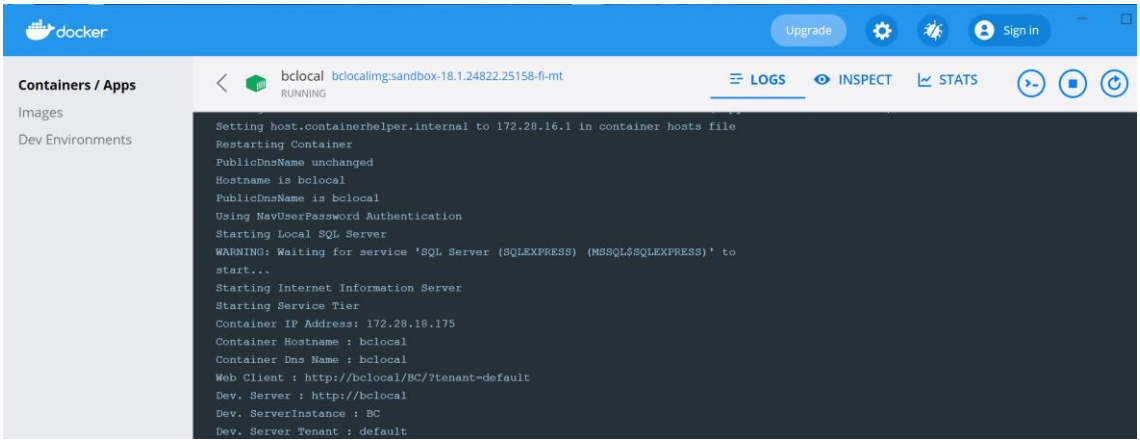
Kuten ylläolevasta kuviosta 7 huomaa, ohjelma korostaa vakiovalintaa, tässä tapauksessa valintaa ”**Local docker container**”. Mikäli tarkoituksena on vain kehittää laajennuksia Business Centraliin, voi kaikkiin kysymyksiin vastata vakiovalinnalla. Kun kaikkiin kysymyksiin on vastattu, tekee

ohjelma muistiotiedoston. Tämä tiedosto pitää sisällään generoidut luontilausekkeet levykuvan latausta varten. Nämä lausekkeet pitää laittaa Windows PowerShell ISE:n (kuvio 8), joka on laajennettu PowerShell graafisella käyttöliittymällä.



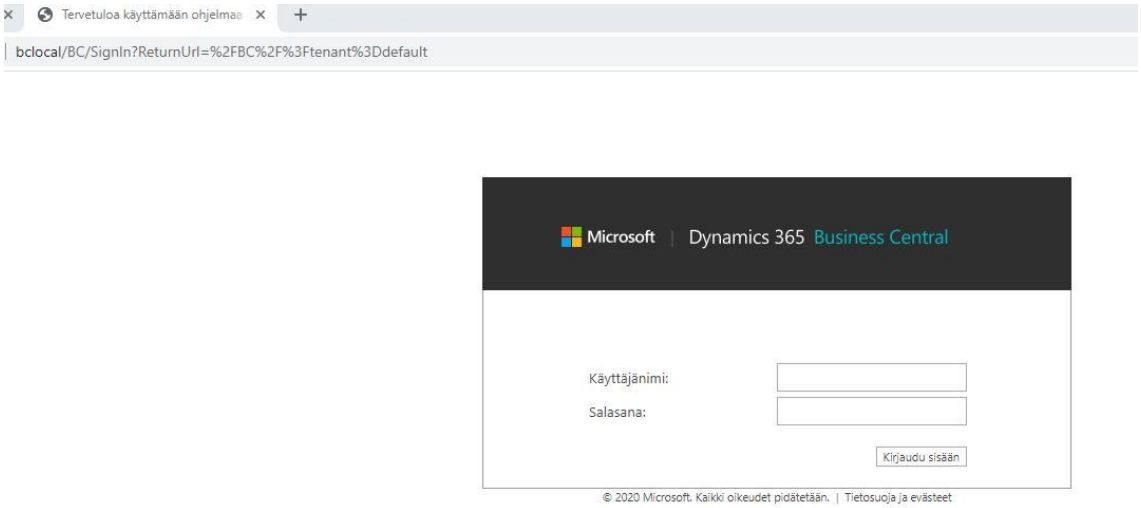
KUVIO 8. Ruutukaappaus Windows PowerShellistä ISE:stä, joka sisältää generoidut luontilausekkeet.

Kun scripti ajetaan, ladataan Business Centralin levykuva koneelle. Tähän voi mennä latauksen ja asennuksen yhteenlaskettuna aikana jopa yli tunti riippuen tietokoneen tehoista. Kun asennus on valmis, ilmestyy se Dockeriin, mistä sen voi laittaa päälle. Docker säiliön aktivoinnissa voi mennä muutama minuutti. Kuviossa 9 on kuvakaappaus aktivoinnin aikana muodostuvista palvelimen tiedoista, joita voidaan myöhemmissä vaiheissa hyödyntää.



KUVIO 9. Ruutukaappaus Dockerista, jossa paikallinen Business Central asennus on laitettu päälle.

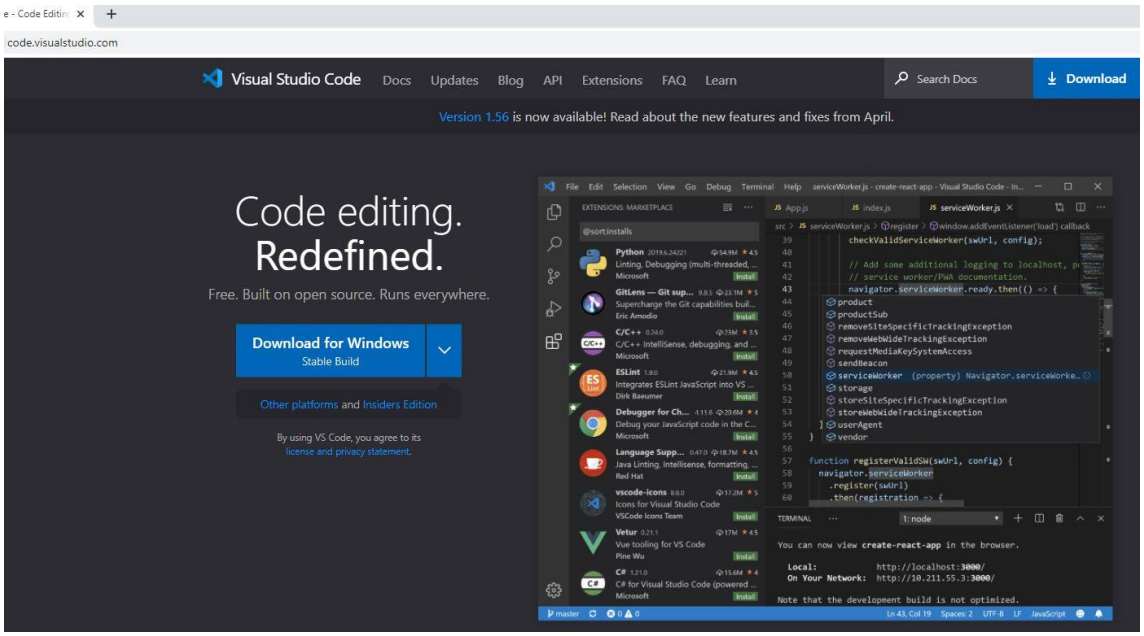
Ylemmän kuvan alareunasta voi huomata kohdan ”**Web Client**”. Tässä kohtaa olevan verkko-osoitteen avulla kirjaudutaan itse Business Centraliin (kuvio 10). Tunnukset pystyi asennusvaiheessa määrittämään itse, mutta jos käytettiin vakiovalintoja on käyttäjänimi ”**admin**”, ja salasana ”**P@ssw0rd**”.



KUVIO 10. Ruutukaappaus Business Centralin kirjautumisruudusta.

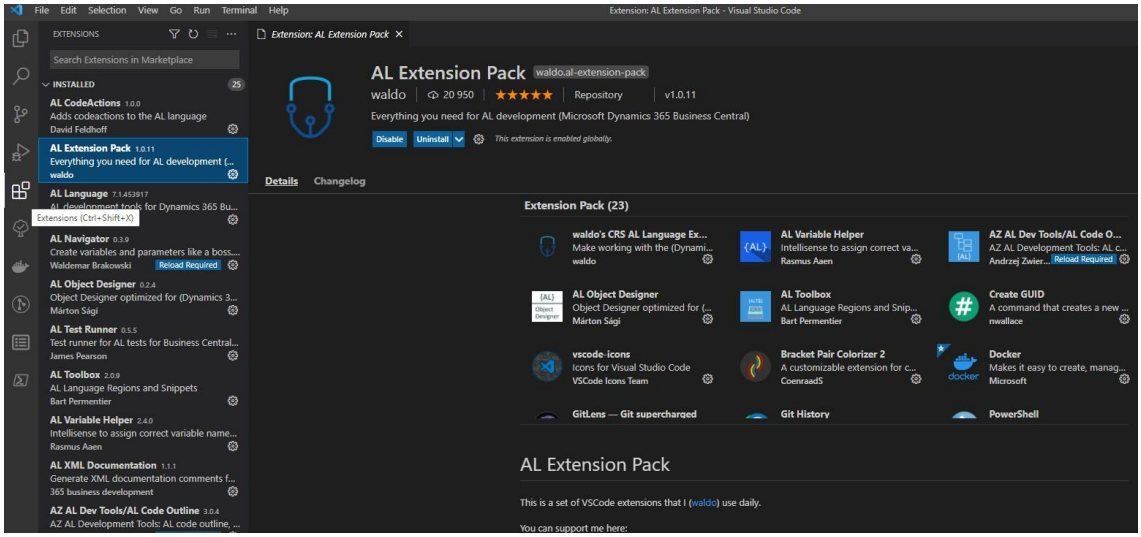
4.1.2 Visual Studio Code

Itse laajennuksen ohjelmointiin käytetään Visual Studio Code editoria. Tässä hyödyttävät erittäin paljon Coden ja Dockerin välinen linkitys, joka tekee ohjelmoinnin tuloksen testauksesta helppoa. Myös Visual Studio Coden sisältämä laajennuskauppa, josta on saatavilla AL-kielen ohjelmoinnissa hyödyttävä laajennus auttaa paljon virheiden etsinnässä, sekä automaattisessa täydennyksessä. Visual Studio Coden voi ladata ilmaiseksi heidän verkkosivuiltaan osoitteesta ”https://code.visualstudio.com/” (kuvio 11).



KUVIO 11. Ruutukaappaus Visual Studio Coden etusivulta, jossa on myös latauslinkki.

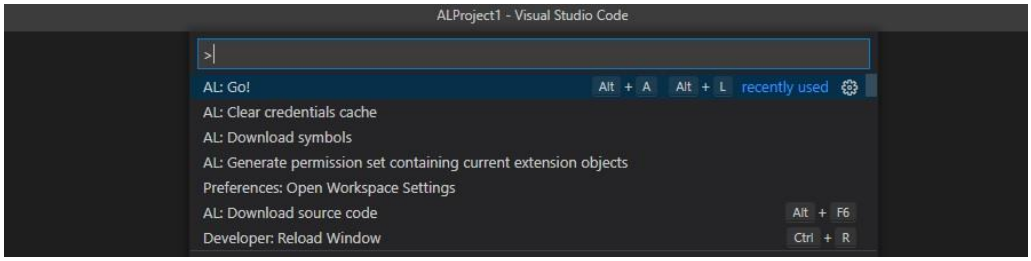
Kun Visual Studio Code on asennettu, kannattaa ennen varsinaista projektin aloittamista ladata ”**AL Extension Pack**” laajennusten kauppapaikalta. Laajennus auttaa oikeaoppisen syntaksin tarkastuksessa AL-kielessä, sekä sisältää myös muita ohjemointia helpottavia paketteja, kuten Docker integraation sekä Git historian.



KUVIO 12. Ruutukaappaus Visual Studio Coden laajennuskaupasta AL Extension Pack valittuna.

Kun kuvion 12 AL-kielen laajennus on asennettu, saadaan käyttöön ohjelmointikieleen liittyvät komennot Visual Studio Coden komentopalettiin. Tämä antaa mahdollisuuden aloittaa uuden projektin AL-kielellä, sekä antaa käyttäjälle myös muita hyödyllisiä komentoja AL-koodiin liittyen,

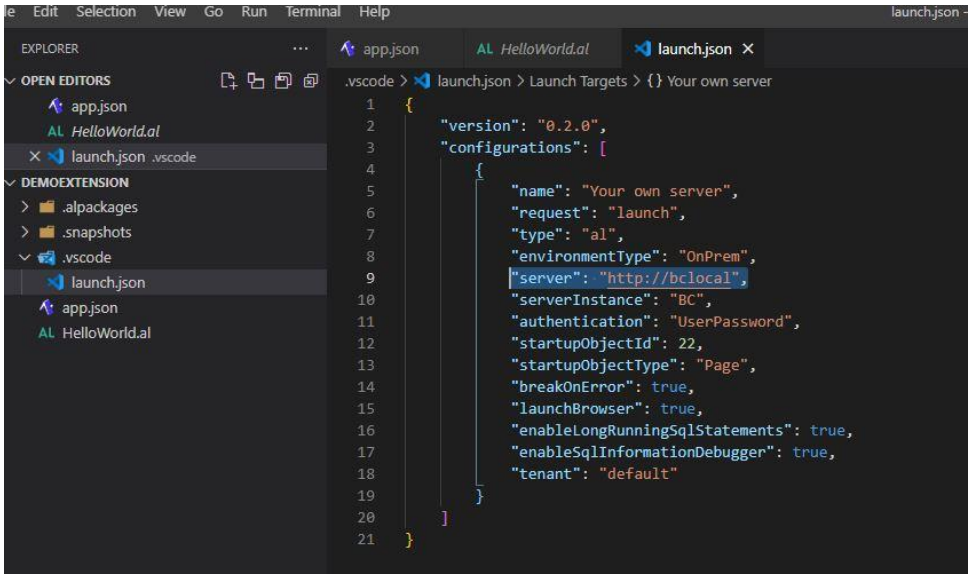
kuten koodin testaamiseen liittyviä toimintoja. Komentopaletti aukeaa näppäinyhdistelmällä **Ctrl+Shift+P** (kuvio 13). Jotta uuden projektin voi aloittaa, pitää avautuvasta komentopaletista avata valinta "***AL:Go!***". Tämän jälkeen komentopaletti kysyy kansiota ja nimeä projektille, sekä Business Centralin versiota. Microsoft julkaisee kaksi kertaa vuodessa ison päivityksen Business Centraliin, jotka myös lisäävät, ja poistavat ominaisuuksia käytöstä. Laajennusten yhteensopivuuden takia kannattaa aina työskennellä viimeisimmän version kanssa. Tämän jälkeen kysytään vielä kohdeympäristöä. Vaihtoehtoina on Microsoftin tarjoama sandbox-ympäristö, tai paikallinen Dockeriin sidottu ympäristö. Valinnalla ei ole väliä, koska asetuksia voi muuttaa vapaasti **launch.json** tiedostossa projektin luonnin jälkeen.



KUVIO 13. Ruutukaappaus avoimesta komentopaletista AL:Go! valittuna.

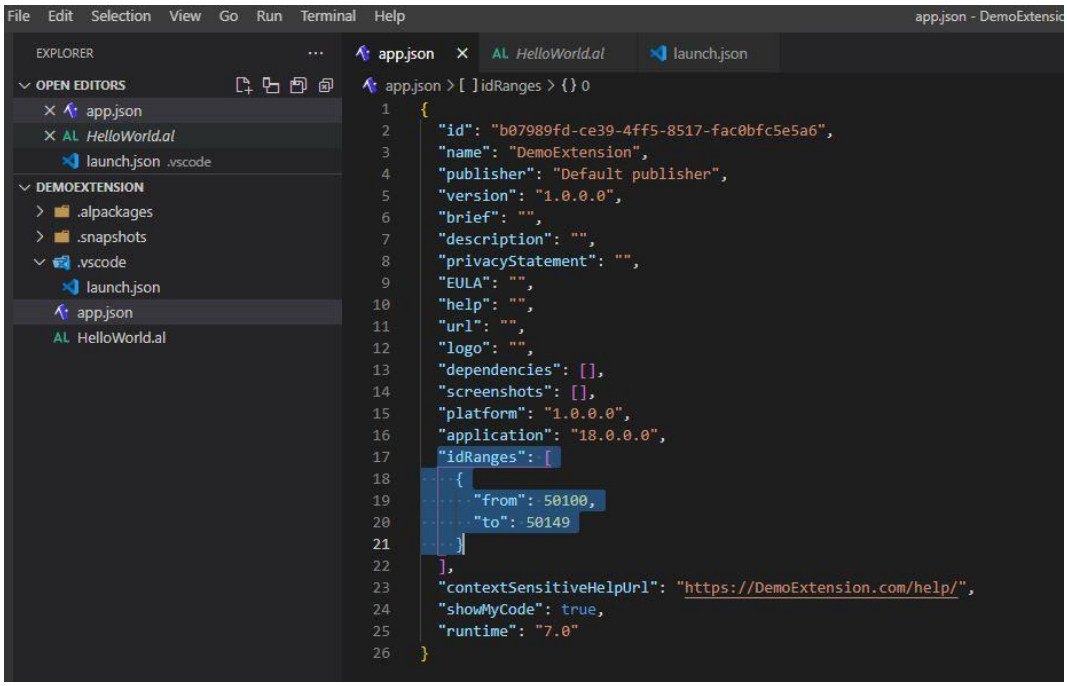
4.2 Projektin alkuasetukset

Mikäli levykuvakkeen luonnin aikana valittiin paikalliselle levykuvakkeelle eri nimi, pitää se käydä vaihtamassa kuviossa 14 näkyvässä **launch.json** tiedostossa. Koska levykuvakkeen nimi on myös palvelimen osoite, estäisi se Visual Studio Codea yhdistymästä paikalliseen Business Central ympäristöön ja sen tietokantaan. Tämä myös estäisi lataamasta symbooleja, joka lopputuloksena tarkoittaisi että jokainen rivi koodia näkyisi virheenä. Levykuvakkeen nimen voi tarkistaa Docker säiliön käynnistämisen jälkeen, kuten kuviosta 9 näkee.



KUVIO 14. Ruutukaappaus launch.json tiedostosta palvelinnimi ylimaalattuna.

Toinen tärkeä asia on app.json tiedosto. Se sisältää informaatiota laajennuksen valmistajasta, jotka näkyvät Business Centralissa. Tämä voi olla hyödyllistä asiakkaalle, mikäli laajennus aiheuttaa ongelmia ja he tarvivat tukea sen kanssa. Toinen hyvin tärkeä asia on kuviossa 15 väritetty tunnuslukualue, jonka sisältämälle alueelle **kaikki** laajennuksessa luotavat objektit kuuluvat. Kahdella eri laajennuksella **ei voi** *olla* saman numeron tunnuslukua yhdessä ja samassa Business Central ympäristössä, elleivät ne kuulu eri kategoriaan. Esimerkiksi tietokantataululla ja sivulla voi olla tunnusluku 90,000, mutta kahdella eri tietokantataululla ei voi olla sama luku. Tämän takia on tärkeää, että yrityksen sisällä pidetään kirjaa käytettävistä luvuista eri laajennuksissa. Microsoft on varannut numerot 0-49,999 Business Centralin perusohjelman käyttöön, ja antanut luvut 50,000-99,999 laajennusten tekijöiden käyttöön.



KUVIO 15. Ruutukaappaus app.json tiedostosta tunnuslukualue ylimaalattuna.

Kun kaikki asetukset ovat valmiita, voi Visual Studio Codesta käynnistää suoraan paikallisella palvelimella olevan Business Central-ympäristön. Tämä on myös hyvä siitä, että kaikki ohjelmoidut laajennuksen toiminnallisuudet ladataan samallakertaa ympäristöön. Tämä onnistuu painamalla **F5**. Mikäli käyttäjä haluaa myös virheenjäljittää, mikä on varsinkin monimutkaisimmissa laajennuksissa käytännöllistä, on näppäinyhdistelmä **Ctrl+F5**. Visual Studio kysyy ensimmäisellä kerralla käyttäjää ja salasanaa, mitkä ovat samat kuin levykuvakkeen luonnissa annetut. Mikäli tuolloin tehtiin levykuvake vakioasetuksilla, ovat ne ”**admin**” ja ”**P@ssw0rd**”.

5 LAAJENNUKSEN OHJELMOINTI

Business Centralia käyttävä toimistokalusteita myyvä yritys haluaa pitää kirjaa heillä huolletuista kalusteista. Koska tällaiste toiminnallisuutta ei perusohjelmassa ole, käydään tässä läpi miten sellaisen voi tehdä. Ensiksi on kuitenkin hyvä perehtyä, mitä toiminnallisuuksia sellaisen tekeminen vaatisi:

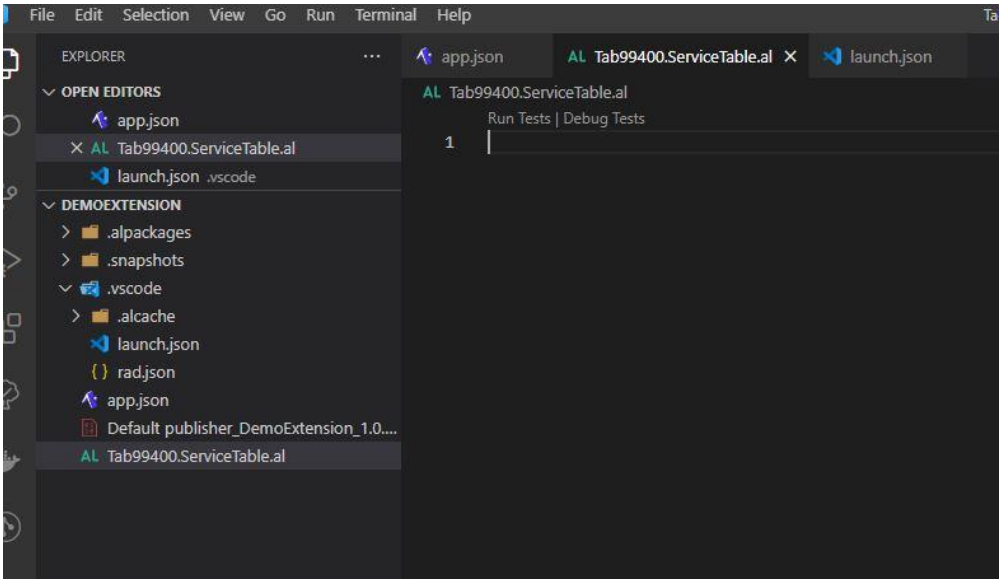
1. Jotta huolletut tuotteet löytyisi helposti, tulisi olla jonkinlainen koontisivu. Tällä sivulla näkyisi kaikki tuotteet, ja jonkin niistä valitsemalla pääsisi näkemään tarkemmin tietoja tapahtuneesta huoltokerrasta.
2. Yksittäinen sivu, joka pitää sisällään yhden tuotteen huoltokerran sisältämät tiedot. Tässä näkyisi tarkemmin mikä tuote on kyseessä, mahdollisesti asiakkaan tiedot, ja vapaa kommentti huollon suorittajalta.

Yhteenvetona laajennus tulee tarvitsemaan kaksi yrityyppistä sivua, yhden uuden raportin sekä yhden tietokannan. Tietokannan tietoja voidaan hyödyntää molemmilla sivulla, sillä listatyyppisellä koontisivulla näytetään huoltosivun tietoja myös. Muita toiminnallisuuksia ovat myös Business Centralissa jo olemassa olevien tietokantojen datan hakeminen, jottaa saadaan hyödynnettyä asiakas- ja tuotetietoja huoltosivulla.

5.1 Huoltotaulukko

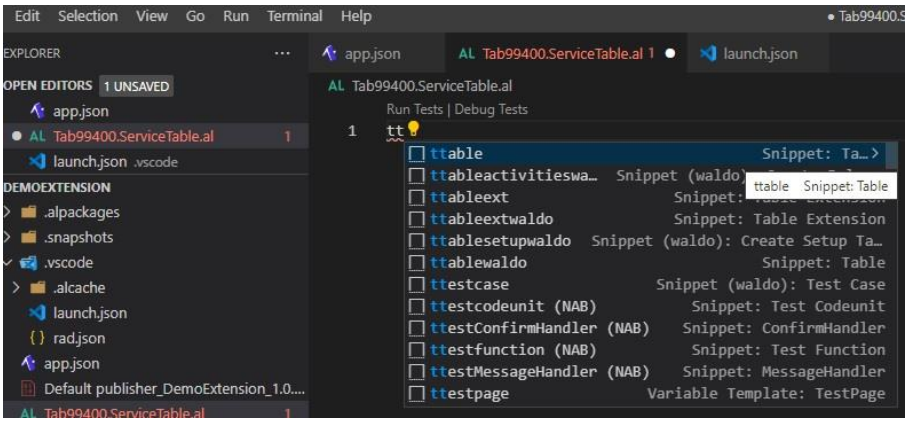
Koska laajennusten lisäämät kentät ja sivut vaativat tietokannan taustalle ennen kuin ne voidaan luoda, on hyvä aloittaa sen tekemällä se ensin. Tämä tehdään luomalla tiedosto projektikansion alle, joka näkyy Visual Studiossa vasemmalla puolella. On hyvän ohjelmistokehitystavan mukaista tuoda jo tiedoston nimessä esille sen tarkoitus, ja varsinkin Business Centralin kanssa myös tunnusluku. Tätä laajennusta varten ollaan *app.json* tiedostossa varattu tunnusluvut 99400-99405. Koska laajennuksessa ei ole tarvetta kovin monelle eri tiedostolle, ei myöskään ole tarvetta varata kovin laajaa lukualuetta. Tässä esimerkissä annamme tiedostolle nimeksi **Tab99400.ServiceTable.al**. Nimen ensimmäinen osa kertoo kyseessä olevan tietokantataulu, jonka jälkeen tulee tunnusluku. Näiden lisäksi nimetään taulu asianmukaisesti, tässä tapauksessa

taulu pitää sisällään tavaroiden huoltoon liittyviä tietueita. Lopuksi annamme tiedostopäätteeksi ”**al**”, joka on AL-kielen tiedostotunnus. Kuviossa 16 näkyy uusi, mutta vielä tyhjä .al tiedosto.

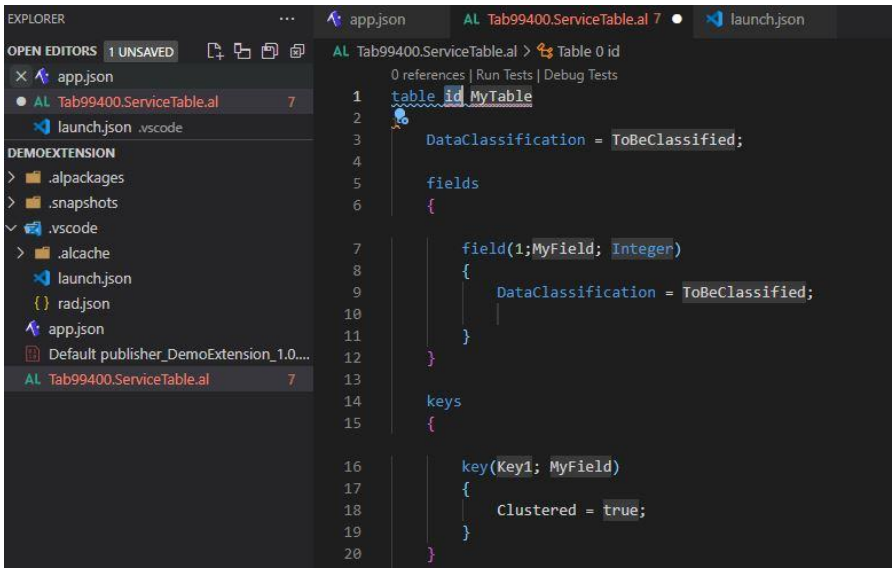


KUVIO 16. Ruutukaappaus tyhjästä tietokantatiedostosta.

Nopeuttaakseen ohjelmointia, voidaan käyttää **AL Extension Packin** mukana tulleita koodipätkiä (kuvio 17). Nämä antavat heti alusta asti rakenteen tietokannoille, ja myöhemmissä vaiheissa myös sivuille. Esimerkiksi tietokannan pätkä on ”ttable”. Kaikissa laajennuspaketin mukana tulleissa pätkissä ensimäinen kirjain on ”t”.



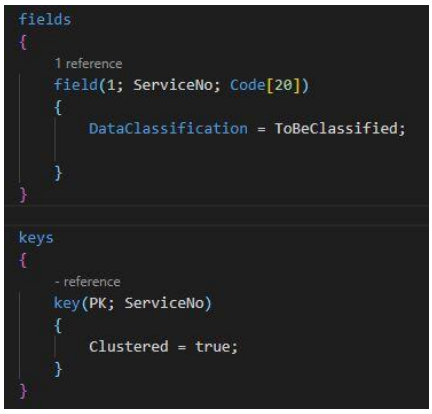
KUVIO 17. Ruutukaappaus AL-kielen laajennuspaketin mukana tulleista koodipätkistä.



KUVIO 18. Ruutukaappaus luodusta tietokantataulun pohjasta.

Ensimmäisenä asiana uudessa taulussa täytyy määrittää sen tunnusluku ja nimi. Tunnusluvun pitää olla jokin aikasemmin varatun alueen sisällä olevista luvuista. Nimen kannattaa olla yhtenäinen tiedostonimessä olevan nimen kanssa. Tässä tapauksessa kutsumme taulua nimellä *ServiceTable.* Tämän jälkeen voimma alkaa määrittämään itse tietueita. Yksittäiset kentät tulevat **fields**-alueen aaltosulkeiden sisälle. Jokainen yksittäinen kenttä pitää olla uniikilla tunnusluvulla ja nimellä varustettu, mutta tämän luvun ei tarvi seurata aikaisemmin määritetyn alueen lukuja.

Tunnusluvun ja nimen lisäksi pitää määrittää myös datan tyyppi. Nämä ovat hyvin pitkälti samoja kuin muissakin kielissä. Kuten ylempänä olevasta kuviosta 18 näkyy, määritetään **keys** osiossa myös tietokannan avain. Yleensä tämä on ensimmäisen kenttä, joten sen olisi tässä tapauksessa hyvä yksilöidä koko huoltotapahtuma. Tämän takia myös datan tyypin on hyvä olla numeraalinen (kuvio 19).



KUVIO 19. Ruutukaappaus tietokantataulun yksilöivästä kentästä.

Tämän lisäksi tarvitaan myös muita tietueita, näitä ovat:

- Työn tilaan viittaava tietue. Tämä voi pitää sisällään muutaman vaihtoehdon, jotka Business Centralissa näkyy pudotusvalikkona.
- Tietue huollossa olevalle nimikkeelle. Tämä voi olla yksinkertaisimmillaan sen nimi. Datatyyppi kannattaa laittaa tekstiksi.
- Asiakkaan tietue, joka voidaan tehdä samallalailla kuin nimikkeen tietue.
- Tekstikenttä huoltoa suorittavan henkilön kommenteille.

Alhaalla olevassa kuviossa 20 näytetään miten nämä on hoidettu. Tietue **"Service Status"** sisältää datatyypin **option**. Mikä Business Centralissa tarkoittaa monivalintaista kenttää. Sen sisälle on myös annettu vaihtoehdot **OptionMembers** kohtaan. **InitValue** puolestaan tarkoittaa arvoa, joka kentällä on vakiona, ja sen on hyvä olla jokin tyhjään arvoon viittaava.

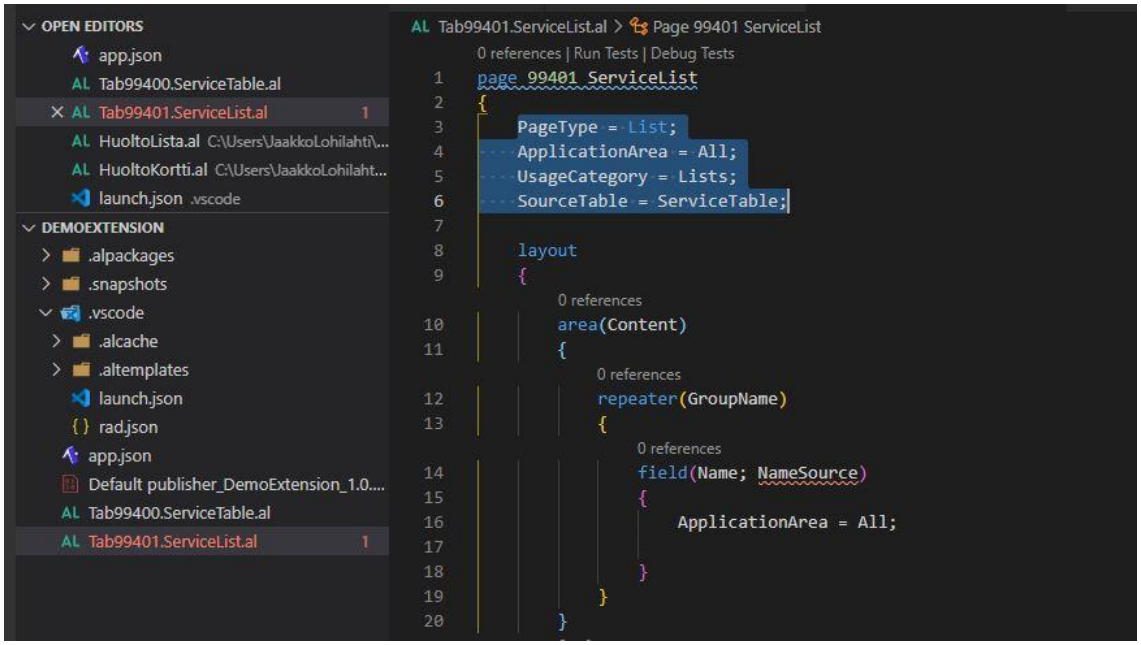
Tietueet **CustomerName** ja **ItemDescription** ovat asiakkaan ja nimikkeen nimet. **"Comment Field"** kenttä on muuten samanlainen, mutta sisältää 300 merkin rajan 100 sijaan.

```
5   fields
6   {
        2 references
7       field(1; ServiceNo; Code[20])
8       {
9           DataClassification = ToBeClassified;
10
11      }
        1 reference
12      field(2; "Service Status"; Option)
13      {
14          Caption = 'Service Status';
15          InitValue = "--";
16          OptionMembers = "--","Jonossa","Työn alla","Valmis";
17      }
        0 references
18      field(3; CustomerName; Text[100])
19      {
20          Caption = 'Customer Name';
21
22
23      }
        1 reference
24      field(4; ItemDescription; Text[100])
25      {
26          Caption = 'Description';
27
28
29      }
        0 references
30      field(5; "Comment Field"; Text[300])
31      {
32          Caption = 'Description';
33
34      }
35  }
```

KUVIO 20. Ruutukaappaus huoltolaajennuksen valmiista tietokannasta.

5.2 Huoltolista-sivu

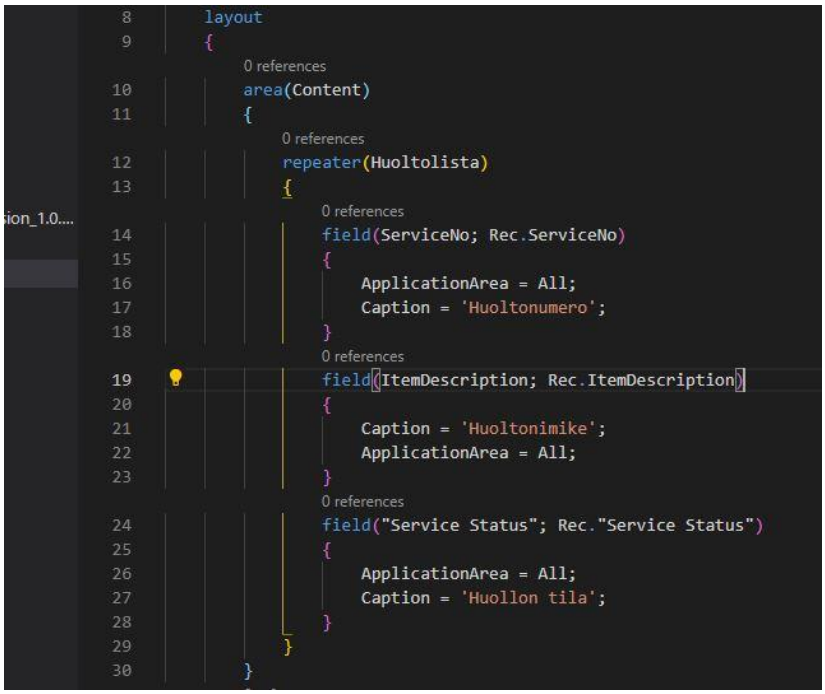
Kun tietokantaosio on valmis, voidaan alkaa ohjelmoimaan Business Centralissa näkyviä osia. Ennen kuin voimme tehdä yksittäisen nimikkeen huoltosivun, on hyvä olla koontisivu valmiina. Tällä sivulla näkyy kaikki nimikkeet sekä niiden huoltotilanne nopealla katsauksella. Seuraten luodun tietokannan esimerkkiä, voidaan käyttää samanlaista tiedoston nimeämisperiaatetta. Sen jälkeen kun tiedosto on luotu, tulee pätkällä tpage monta eri vaihtoehtoa esille. Yhdellä näistä saadaan suoraan pohja listatyyppiselle koontisivulle.



KUVIO 21. Ruutukaappaus luodusta listasivun pohjasta.

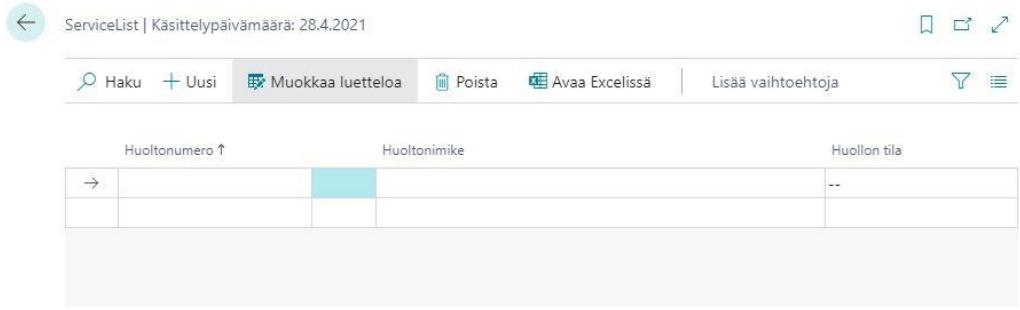
Sivutunnuksen ja nimen alapuolella voidaan nähdä, kuinka sivu on tyypitetty listamaiseksi **PageType** kohdassa (kuvio 21). Myös **SourceTable** kohdassa voidaan nähdä, miten aikaisemmin tehty tietokanta on sivun käytössä. Kenttien lisääminen on hyvin samanlaista kuin tietokantojen tietueiden teko, sillä erotuksella ettei niitä tarvitse numeroida. Ne koostuvat kahdesta osasta, omasta nimestä ja tietueeseen linkittävästä nimestä. Hyvin usein nämä ovatkin samoja, mutta ainoastaan jälkimmäisen pitää olla täysin sama kuin tietokannassa. Jos nimi on laitettu lainausmerkkien sisälle, voi se myös sisältää välilyöntejä.

Koska listasivu ei välttämättä tarvitse näyttää kaikkia tietoja tietokannasta, voidaan tällä kertaa käyttää vain Huoltonumeroa, nimikkeen nimeä, ja huollontilaa.



KUVIO 22. Ruutukaappaus listasivun koodista kentillä.

Koska kenttien tietueet sisältävät suurimman osan toiminnallisuudesta, on itse kenttien tekeminen hyvin suoraviivaista. Tässä esimerkissä lisätty **Caption** on käyttäjälle näkyvä kentän nimi Business Centralissa (kuvio 22), ja se onkin riippumaton koodissa olevasta nimestä. Kun kaikki on valmista, voidaan käyttää näppäinkomentoa **Ctrl+F5** puskeakseen laajennuksen testiympäristöön.

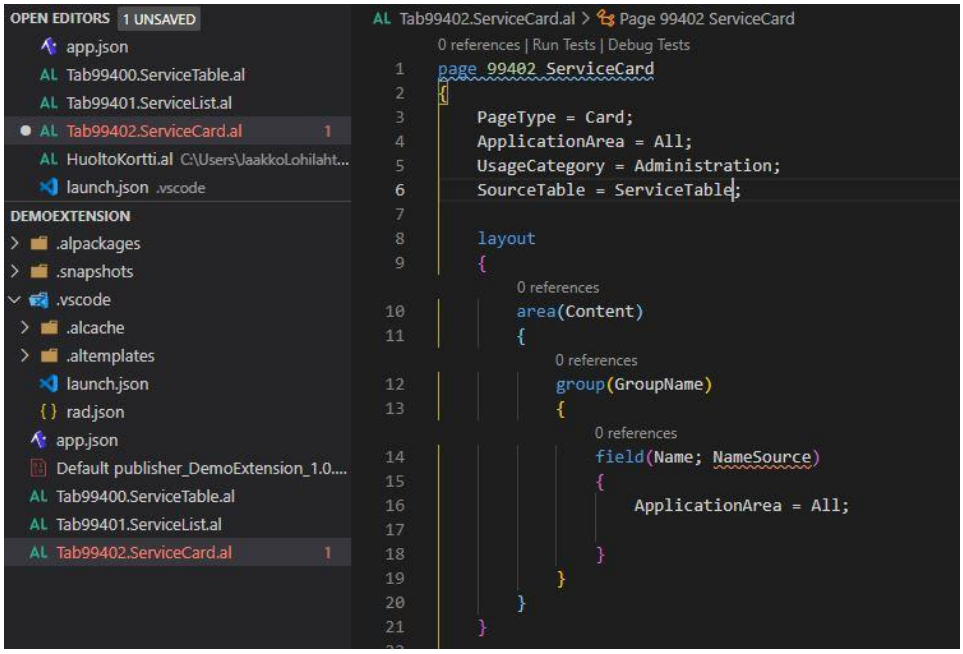


KUVIO 23. Ruutukaappaus listasivusta Business Centralissa.

Huoltoja kokoava listasivu on valmis. Kuviosta 23 voi nähdä, kuinka sivu sisältää toimintoja mitä laajennuksessa ei oltu edes määritelty. Melkein jokaista tietokantoja sisältäviä sivuja Business Centralissa voi muokata Excelissä. Tämä on hyödyllistä silloin, kun tarvitsee muokata suuria datamääriä. Listasivun jälkeen voidaan alkaa työskentelemään korttityyppisen sivun kanssa.

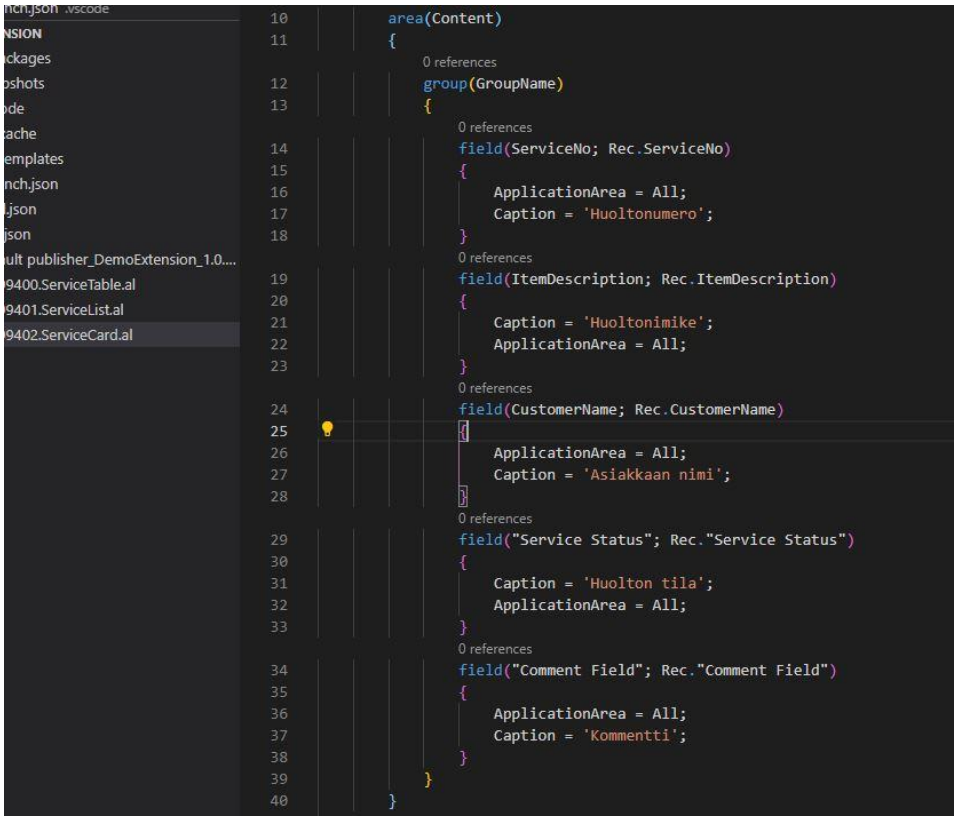
5.3 Huoltokortti

Siinä missä listatyyppinen sivu kerää monen eri sivun tietoja helposti luettavaan muotoon, on korttisivu yksittäinen sivu jollekkin tietylle asialle. Näitä voi olla yksittäinen asiakas, nimike tai tässä yhteydessä yksittäinen huoltokerta. Viime esimerkkien pohjalta korttisivun luonti ei poikkea kaavasta, tällä kertaa valitaan pätkä, joka luo korttisivun listasivun sijaan.



KUVIO 24. Ruutukaappaus korttisivun koodipohjasta.

Kuten kuviosta 24 näkee, hyvin samankaltaiselta näyttävä pohja eroaa vain **PageType** kohdassa. Tällä kertaa voidaan lisätä kaikki tietokannassa olevat tietueet kenttänä sivulle **group** kohdan aaltosulkeiden sisälle (kuvio 25).

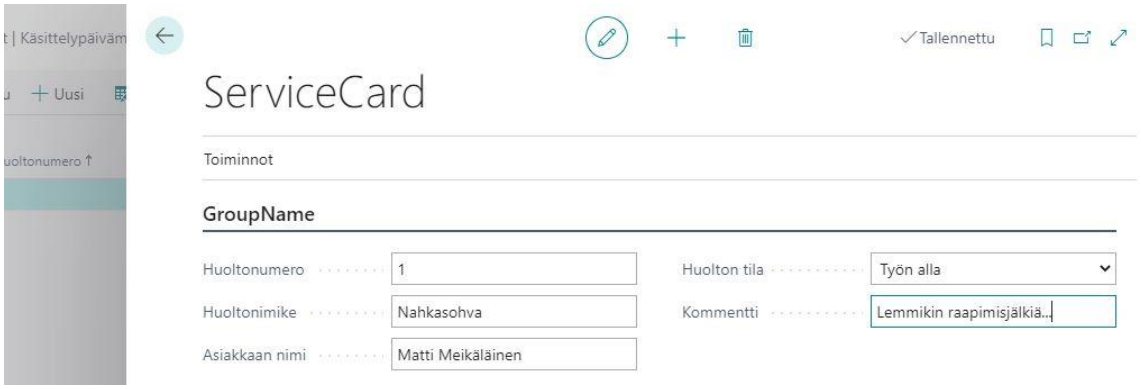


KUVIO 25. Ruutukaappaus korttisivun koodista kaikilla kentillä.

Ennen korttisivun katsomista Business Centralista, pitää se liittää listasivuun kuvion 26 tavalla. Tämä linkitys antaa listasivulle mahdollisuuden koota uuden kortin tiedot aina kun sellainen tehdään. Tämä onnistuu lisäämällä kuvanmukaisesti **CardPageID = ServiceCard;** listasivun koodiin Visual Studiossa.

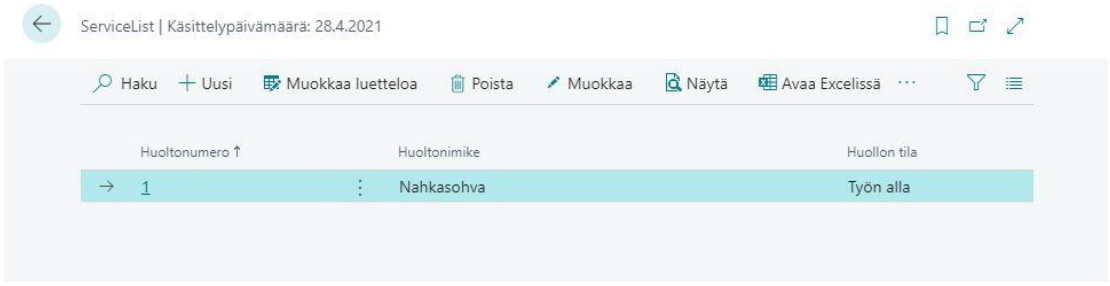
KUVIO 26. Ruutukaappaus korttisivun linkityksestä listasivun koodiin.

Kun kaikki on valmista, voidaan korttisivulle mennä huoltolistan kautta painamalla **Uusi** navigaatiopalkissa.



KUVIO 27. Ruutukaappaus korttisivusta ja sen täytetyistä kentistä.

Kuten kuviosta 27 näkyy, korttisivun tiedot tallentuvat automaattisesti tietokantaan aina kun kenttään on lisätty tietoa. Tämä on myös yksi Business Centralin ominaisuuksista mitä ei tarvitse ohjelmoida erikseen. Pilvipohjaisena ratkaisuna Business Central on aina tallentamassa tuoretta dataa tietokantoihin muutoksen tapahtuessa. Tässä tapauksessa myös huoltolistasivulle kuvion 28 mukaisesti.



KUVIO 28. Ruutukaappaus listasivusta, jossa näkyy kuvan 27 korttisivun tietoja.

Tässä esimerkissä käytiin läpi miten Business Centraliin voi lisätä sivuja ja tietokantoja. Nämä eivät kuitenkaan ole ainoat ominaisuudet, mitä käyttäjä pystyy laajentamaan. Korttisivun kentät voisi esimerkiksi integroida käyttämään Business Centralin nimike- ja asiakastietoja. Yksi mahdollinen lisäominaisuus olisi myös tulostettava raportti huoltokortti-sivulle, johon kerätään sen tiedot joko yrityksen sisäiseen arkistointiin, tai asiakkaalle annettavaan dokumenttiin.

6 YHTEENVETO

Opinnäytetyön tarkoituksena oli pyrkiä selvittämään toiminnanohjausjärjestelmien yrityksille mahdollistava arvo. Jotta olisi helpompi ymmärtää mistä on kysymys, tehtiin tämä tutustumalla ensin niiden historialliseen kehitykseen. Läpi käytiin myös mitä yrityksen on hyvä pitää mielessä järjestelmää hankkiessa, näitä eivät olleet pelkästään itse toiminnanohjausjärjestelmän valinta. Yrityksellä voi olla monta kipukohtaa, mitkä voivat vaikuttavat järjestelmän käyttöönotossa, alun suunnittelussa näitä kaikkia ei välttämättä saa kartoitettua.

Opinnäytetyössä tutustuttiin myös siihen, mitä yksinkertaisen laajennuksen tekoon kuuluu. Koska harvalle yritykselle pystyi suoraan ottamaan käyttöön toiminnanohjausjärjestelmää ilman, että sitä ei muokattaisi jollain asteella. Nämä usein johtuivat yrityksen entisistä tavoista hallita toimintaansa, joita halutaan mahdollisimman kivuttomasti pystyä jatkamaan uudessä järjestelmässä. Tämä on tietenkin ymmärrettävää yrityksen kannalta.

Itse laajennuksen ohjelmointia Microsoft Dynamics 365 Business Centraliin käytiin läpi raportin viimeisellä puoliskolla. Ohjelmiston laajennuksena tehtiin yksinkertainen sivu järjestelmään, johon käyttäjä voi laittaa tietoja, ja jotka lopulta kerättiin paremmin näkyville erilliselle koontisivulle. Ajatuksena oli tuoda esille, mitä tällaisen kokonaisuuden teko vaatii ohjelmoinnin osalta. Tarkoitus oli tuoda esille Business Centralin ohjelmointiin käytettävää AL-kieltä, ja näyttää miten suhteellisen lyhyillä pätkillä koodia pystyy toiminnanohjausjärjestelmään tekemään uusia ominaisuuksia.

Laajennuksen ohjelmoinnin raportointi toteutettiin askel askeleelta niin, että opinnäytetyötä seuraamalla olisi mahdollista päästä alkuun laajennusten ohjelmoinnissa Business Centraliin. Tätä auttaakseen sisällytettiin myös kehitysympäristöjen pystyttäminen raporttiin. Olemassaolevan järjestelmän ohjelmointi on huomattavasti erillaisempaa sen omien sääntöjen takia. Kun nämä oppii, antaa se kuitenkin antaa helposti seurattavan rakenteen laajennusten kehittämiseen.

LÄHTEET

Anikin, D. 2016. What an in-memory database is and how it persists data efficiently. Viitattu 20.5.2021, https://medium.com/@denisanikin/what-an-in-memory-database-is-and-how-it-persists-data-efficiently-f43868cff4c1

Bendoly, E. & Robert Jacobs, F. 2003. Enterprise resource planning: Developments and directions for operations management research. European Journal of Operational Research.

Better Buys. 2021. How Much Does and ERP system Cost? 2021 Pricing Guide. Viitattu 20.5.2021, https://www.betterbuys.com/erp/erp-pricing-guide/

Demiliani, S. & Tacconi, D. 2018. *Dynamics 365 business central development quick start guide: Modern development techniques for dynamics 365 business central*. 1st edition. Birmingham; Mumbai: Packt.

ERP News. 2016. 7 Signs Your Company Needs an ERP. Viitattu 20.5.2021, https://erpnews.com/7-signs-company-needs-erp/

Ferrari, S. 2019. SAP S/4HANA System Conversion: Keep Calm and Move with Brownfield. Viitattu 20.5.2021, https://www.techedgegroup.com/blog/s4-hana-brownfield-greenfield-bluefield

Hossain, L. & Patrick, J. D. & Rashid, M. A. 2002. The Evolution of ERP Systems: A Historical Perspective. Idea Group Publishing.

Kodella. 2020. A Brief History of NetSuite: The First Cloud Company. Viitattu 20.5.2021, https://kodella.com/netsuite-erp/a-brief-history-of-netsuite-and-its-solutions/

Luther, D. 2021. 8 ERP Trends for 2021. Viitattu 20.5.2021, https://www.netsuite.com/portal/resource/articles/erp/erp-trends.shtml

Markovski, M. & Micik, A. & Pang, A. 2020. Top 10 ERP Software Vendors, Market Size and Market Forecast 2019-2024. Viitattu 20.5.2021, https://www.appsruntheworld.com/top-10-erp-software-vendors-and-market-forecast/

McCue, I. 2020. The History of ERP. Viitattu 20.5.2021, https://www.netsuite.com/portal/resource/articles/erp/erp-history.shtml

Microsoft. 2021. Dynamics 365 Business Centralin hinnoittelu. Viitattu 20.5.2021, https://dynamics.microsoft.com/fi-fi/business-central/pricing/

Microsoft. What is ERP and why do you need it? Viitattu 20.5.2021, https://dynamics.microsoft.com/en-us/erp/what-is-erp/

Murthy, C. S. V. 2008. Enterprise resource planning and management information systems: (text and case studies). Mumbai [India]: Himalaya Pub. House Pvt. Ltd.

Needleman, T. 2019. The Best ERP Software. Viitattu 20.5.2021, https://www.pcmag.com/picks/the-best-erp-software

Nestell, J. G. & Olson, D. L. 2018. *Successful ERP systems:* A guide for businesses and executives. New York, New York: Business Expert Press.

PaperFree. 2020. SAP ERP vs Oracle ERP. Viitattu 20.5.2021, https://paperfree.com/en/Magazine/Sap-vs-Oracle-ERP-Comparison#lnk_more

SAP. 2021. SAP S/4HANA Cloud ERP system. Viitattu 20.5.2021, https://www.sap.com/products/s4hana-erp.html

SAP. 2021. The early years. Viitattu 20.5.2021, https://www.sap.com/about/company/history/1972-1980.html

Simon. 3 Reasons You Don't Really Need ERP. Viitattu 20.5.2021, https://www.cin7.com/blog/supply-chain/3-reasons-you-dont-really-need-erp/

Wilson, D. 2019. "The ERP Software Market: $35 billion+, 40 years in the making, but still growing nicely!" by Chris Pang. Viitattu 20.5.2021, https://blogs.gartner.com/debbie_wilson/2019/04/19/erp-software-market-35-billion-40-years-making-still-growing-nicely-chris-pang/