

Antti Haverinen

**MICROSOFT POWER PLATFORMIN HYÖDYNTÄMINEN PROVENTIA OY:N
LÄMPÖERISTETUOTANNOSSA**

**MICROSOFT POWER PLATFORMIN HYÖDYNTÄMINEN PROVENTIA OY:N
LÄMPÖERISTETUOTANNOSSA**

Antti Haverinen
Opinnäytetyö
Kevät 2024
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehitys

Tekijä: Antti Haverinen

Opinnäytetyön nimi: Microsoft Power Platformin hyödyntäminen Proventia Oy:n lämpöeristetuo-
nossa

Työn ohjaaja(t): Jouni Juntunen

Työn valmistumislukukausi ja -vuosi: kevät 2024

Sivumäärä: 38

Työ käsittelee raaka-aineen käytön seurantasovelluksen sekä siihen liittyvän raportin suunnittelua ja toteutusta toimeksiantajayritykselle Microsoftin Power Platform -kehitystyökaluja hyödyntämällä. Toimeksiantajayritys Proventia Oy on teknologiayhtiö, jonka päätuotteita ovat raskaiden työkoneiden päästöjen puhdistusjärjestelmät.

Työn tarkoituksena oli korvata käytössä ollut Excel-taulukko ja vähentää ylimääräisiä työvaiheita luomalla sovellus, joka olisi tuotantohenkilöstön käytettävissä mobiililaitteella. Aiemman ratkaisun ongelmaksi yrityksessä koettiin samojen asioiden kirjaaminen useaan kertaan: Ensin tuotantohenkilöstö kirjasi tiedot paperille, minkä jälkeen tieto siirrettiin toimistossa paperilta Excelliin.

Käytettäviksi työkaluiksi valikoituivat sovelluksen kehittämiseen Power Apps sekä raportin luomiseen Power BI. Toimeksiantajayrityksellä oli jo valmiiksi kokemuksia sekä Power Appsilla toteutettujen sovellusten että Power BI:llä toteutettujen raporttien käyttämisestä, joten niiden pilotointi sisäisessä kehitystyössä nähtiin järkeväksi. Lisäksi yrityksessä on laajasti käytössä muita Microsoftin tuotteita muun muassa toiminnanohjausjärjestelmän muodossa, joten Microsoftin tuotteet nähtiin järkevänä ratkaisuna myös kehitystyössä.

Tuloksena oli sekä toimiva sovellus että toimiva raportti. Sovellus on ollut huhtikuuhun 2024 mennessä testikäytössä noin kahden kuukauden ajan ja siitä on hyviä kokemuksia. Se on tarkoitus julkaista yleiseen käyttöön kesän 2024 aikana. Toimeksiantajayrityksessä on tarkoitus jatkossa kasvattaa yrityksen sisäistä osaamista sekä sovelluskehityksen että raportoinnin osalta ja työ osoitti sen olevan mahdollista Microsoftin työkaluja hyödyntämällä.

Asiasanat: Microsoft Power Platform, Microsoft Power Apps, Microsoft Power BI, vähäkoodinen kehitys

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Software Engineering

Author: Antti Haverinen

Title of thesis: Utilizing Microsoft Power Platform in Proventia Oy's thermal component production

Supervisor: Jouni Juntunen

Term and year when the thesis was submitted: Spring 2024

Number of pages: 38

The goal of this bachelor's thesis was to develop an app and a visual report using Microsoft Power Platform tools. The client company, Proventia Oy, is a technology company located in Oulu, Finland and it's specialized in emission control systems for off-road machinery.

The goal for the app was to replace an old Excel file and to remove excessive work steps. Microsoft Power Platform tools were chosen because the client company already had some outsourced solutions in use which were made using Power Platform. In the future, the client company aims to use the same tools used in this thesis to develop solutions in-house.

The result for both the app and the visual report met all expectations. The app has been in pilot use during the Spring 2024 and the aim is that it's released for general use in the company during the Summer 2024.

Keywords: Microsoft Power Platform, Microsoft Power Apps, Microsoft Power BI, low-code development

SISÄLLYS

1	JOHDANTO	6
2	VÄHÄKOODINEN KEHITYS	9
3	MICROSOFT POWER PLATFORM	12
3.1	Microsoft Power Apps	12
3.2	Microsoft Power BI	17
4	SOVELLUKSEN JA RAPORTIN KEHITTÄMINEN	20
4.1	Vaatimusmäärittely	21
4.2	SharePoint-luettelon luominen	22
4.3	Power Apps -sovelluksen kehittäminen	23
4.4	Luettelonäkymä sovelluksessa	25
4.5	Päänäkymä sovelluksessa	27
4.6	Power BI -raportin kehittäminen	30
5	POHDINTA	34
	LÄHTEET	36

1 JOHDANTO

Tämän opinnäytetyön tavoitteena on toteuttaa raaka-aineen käytön seurantasovellus Microsoft Power Platformilla Proventia Oy:n lämpöeristeiden sarjatuotannossa. Työ toimii samalla pilottiprojektina yrityksen sisäisessä sovelluskehityksessä. Proventialla on tunnistettu tarve modernisoida työtehtäviä, joihin nykyisillä toimintatavoilla liittyy paljon Excelin käyttöä ja paperisten dokumenttien siirtoa henkilöltä toiselle. Power Platformin työkalut valittiin käyttöön, koska yrityksessä on jo ennestään muutamia ulkopuolisten palveluntarjoajien Power Appsilla toteuttamia ratkaisuja sekä Power BI:llä toteutettuja raportteja. Tulevaisuudessa sekä sovelluskehitystä että raportointia on tarkoitus siirtää entistä enemmän yrityksen sisällä toteutettavaksi ja tähän sopivimmaksi koetaan vähäkoodiset (englanniksi low-code) kehitysohjelmat, koska niiden käyttöön ei vaadita aiempaa ohjelmointitaitoa. Proventialla on myös valmiiksi laajasti käytössä Microsoftin palvelut, muun muassa toiminnanohjausjärjestelmä Business Central, joten Power Platformin käytön uskotaan tuovan synergiaetuja sen Microsoftin muihin järjestelmiin yhteensopivien liitännöiden kautta.

Proventia Oy on vuonna 1994 perustettu oululainen teknologiayhtiö, jonka pääkonttori sijaitsee Oulunsalossa. Proventian tuotteilla ja palveluilla pyritään torjumaan ilmastonmuutosta sekä ratkaisemaan nykyisiä ilmansaasteongelmia. Toimipisteitä yrityksellä on Suomen lisäksi Iso-Britanniassa ja Tšekissä, jossa sijaitsee päästöhallintajärjestelmien sekä lämpöeristeiden sarjatuotantolaitos. Vuoden 2024 alussa Proventialla oli noin 200 työntekijää ja se on organisoitu kahteen liiketoimintalueeseen: työkoneiden järjestelmiin ja komponentteihin, joka jaetaan päästöjen puhdistusjärjestelmiin, lämpöeristeisiin ja akustoihin sekä testausjärjestelmiin. (Proventia Oy 2024a.)

Proventian lämpöeristeiden tuotanto tapahtuu Oulussa Limingantullissa sekä Tšekissä Kuřimissa. Proventian lämpöeristeistä puhutaan PTC-tuotteina, mikä tulee sanoista Proventia Thermal Components. Lämpöeristeiden tarkoitus on auttaa pakokaasun puhdistusjärjestelmien lämmönhallinnassa, jotta ne toimisivat mahdollisimman optimaalisesti. Eristeet vähentävät lämmön säteilyä ja parantavat paloturvallisuutta ahtaissa moottoritiloissa. Raaka-aineina lämpöeristeissä toimivat 0,15 millimetriä paksu ruostumaton teräs sekä eristevilla. Ruostumattomaan teräkseen embossataan pintakuviointi ennen varsinaisen tuotteen valmistamista prässi- ja leikkauslinjoilla. Eristevillan paksaus on pääsääntöisesti 3–25 millimetriä. (Proventia Oy 2024b.) Erilaisia lämpöeristeitä on esitetty kuvassa 1.



KUVA 1. Proventia Oy:n lämpöeristeitä (Proventia 2024c)

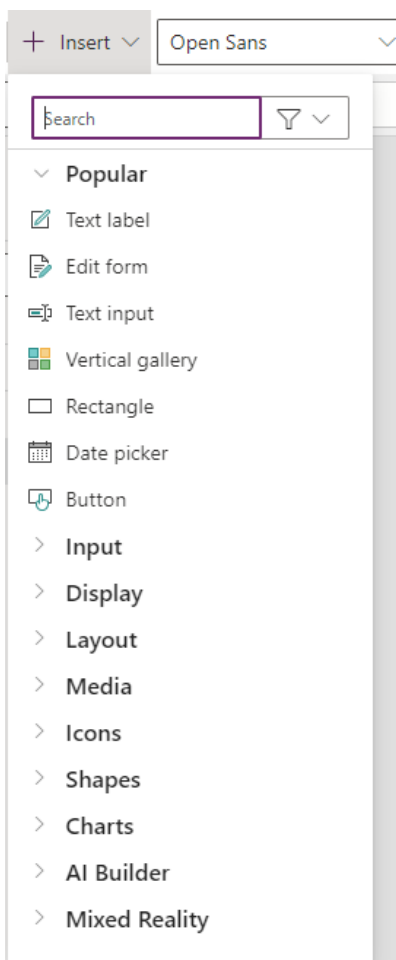
Proventian lämpöeristeiden tuotannossa on jo käytössä toimivaksi todettu Power BI -raportti, jonka avulla muun muassa tarkastellaan tuotannon toteutuneita valmistusmääriä. Toiminnanohjausjärjestelmän kautta saatavat tiedot ovat usein hankalasti luettavassa muodossa ja ne joudutaan hakemaan järjestelmästä useasta sijainnista, minkä jälkeen tietojen yhdistelyn joutuu tekemään käsin esimerkiksi Exceliä hyödyntäen. Business Centralin valmiit raporttipohjat ovat geneerisiä ja eivät yleensä suoraan palvele eri liiketoiminta-alueiden tarpeita suoraan. Power BI -raporttien on koettu nopeuttavan ja selkeyttävän tietojen hakua järjestelmästä.

Lämpöeristeiden tuotannossa käytetään vielä paljon perinteistä tapaa merkitä tiedot kynällä paperille. Tiedot viedään tämän jälkeen vielä yleensä esimerkiksi toiminnanohjausjärjestelmään tai Exceliin, jolloin sama työ tulee tehtyä kahteen kertaan ja virheen mahdollisuus esimerkiksi erilaisten käsialojen vuoksi kasvaa. Tämän työn puitteissa tarkoituksena on kehittää lämpöeristeiden tuotannossa syntyvien niin sanottujen hukkapala-aihioiden käytön seurantaan liittyvää ohjelmaa ja ohjelman kautta tehtyjen kirjauksien pohjalta raportointia. Hukkapala-aihiolla tarkoitetaan tuotannossa ruostumatonta terästä leikatessa syntyviä ylimääräisiä paloja, joita voidaan hyödyntää myöhemmin tuotannossa. Tavoitteena on saada tehtyä tuotannon työntekijöiden käyttöön sovellus, jonka avulla

operaattorit voivat itse tehdä hukkapala-aihioiden käyttöön liittyviä kirjauksia. Nykyisessä mallissa operaattorit merkitsevät tuotantotilauksen työkortille maininnan hukkapala-aihioiden käyttämisestä ja tämä tieto siirretään toimiston puolella erilliseen Exceeliin tuotantotyön loppukirjauksen yhteydessä.

2 VÄHÄKODINEN KEHITYS

Vähäkoodinen kehitys (englanniksi low-code) on visuaalinen lähestymistapa ohjelmointiin, jonka on nimensä mukaisesti tarkoitus olla vähemmän käsin kirjoitettua koodia vaativaa kehitystyötä. Perinteisessä ohjelmointiympäristössä ohjelmointi tapahtuu kirjoittamalla koodia rivi riviltä. Vähäkoodisen kehityksen perustana on yleensä graafinen käyttöliittymä, jossa toteutettavaan ohjelmaan lisätään valmiita komponentteja vedä ja pudota -menetelmää hyödyntäen. (IBM 2024.) Kuvassa 2 on esimerkkinä kuvakaappaus Microsoftin Power Apps -ympäristön komponenttien lisäysvalikosta. Valikon kautta saa lisättyä sovellukseen esimerkiksi napin, jonka sijaintia ruudulla pystyy vapaasti muokkaamaan vetämällä. Napin lisääminen tai sen liikuttaminen ei vaadi käyttäjältä perinteistä koodin kirjoittamista.



KUVA 2. Microsoft Power Apps -kehitysympäristön komponenttien lisäysvalikko

Vähäkoodisen kehityksen on tarkoitus olla helposti omaksuttava ohjelmointitapa, joka vaatii vain vähän tietoa perinteisestä ohjelmoinnista. Tarkoituksena on saada kokonaan ilman aiempaa ohjelmointikokemusta tai vähäisen ohjelmointikokemuksen omaavat henkilöt luomaan uusia sovelluksia pienellä vaivalla. Vain hieman ohjelmointiosaamista vaativien kehitysympäristöjen avulla yritykset voivat toteuttaa nopeasti yksinkertaisia ratkaisuja, joihin on aiemmin vaadittu osaavaa ohjelmoijaa. Myös kustannussäästöjä syntyy, kun kehitystyötä tekee olemassa oleva henkilöstö, eikä kaikkia ratkaisuja varten ole välttämätöntä palkata ulkopuolista osaamista. Sovellusten kehitys onnistuu jopa ilman minkäänlaista ohjelmointikokemusta, jolloin yrityksen IT-tiimi voi keskittyä monimutkaisempien sovellusten ja ratkaisujen luomiseen. Vähäkoodisen sovelluskehityksen ei siis ole tarkoitus korvata perinteisempää ohjelmointityötä, vaan tarjota sen rinnalle helpommin opeteltava vaihtoehto yksinkertaisempaan kehitystyöhön. (Microsoft 2024a.) Attidon mukaan (2024) vähäkoodinen kehitys sopii yrityksille, jotka haluavat muun muassa automatisoida rutiineja, helpottaa mobiilikäyttöä, parantaa olemassa olevien järjestelmien käyttöä, suoraviivaistaa liiketoimintaprosesseja sekä palvella asiakkaitaan paremmin.

Terminä vähäkoodinen kehitys on yleistynyt vasta 2010-luvulla. Markkinatutkimusyhtiö Forrester käytti termiä low-code ensimmäistä kertaa vuonna 2014 luokitellessaan tämän termin alle kehitysympäristöt, jotka korostavat sekä kehitysympäristön käytön että niillä tehtävän ohjelmoinnin helpoutta. (Kissflow 2024.) Vähäkoodisen kehityksen juuret ulottuvat kuitenkin huomattavasti pidemmälle. Jo 1970-luvulla pinnalle nousi RAD-kehitys (rapid application development), jonka perusperiaatteisiin kuuluvat nopea kehitystyö pitkän suunnittelun sijasta sekä käyttäjäpalautteen kuunteleminen ja uusien versioiden nopea julkaisu palautteeseen pohjautuen. Verrattuna perinteisempään vesiputousmalliin, jossa ohjelmistotuotanto etenee vaihe vaiheelta alaspäin vesiputouksen tapaan ja käyttäjät saavat ensimmäisen version testattavaksi verraten myöhäisessä vaiheessa, RAD-kehityksessä käyttäjät saavat lopputuotteen prototyypin nopeasti nähtäväksi ja sen pohjalta saatua palautetta hyödynnetään jatkokehityksessä. (Microsoft 2024l.) On huomioitava, että RAD-kehitys ja vähäkoodinen kehitys eivät ole toistensa synonyymeja, vaan voisi sanoa, että RAD on toiminut alkusysäyksenä vähäkoodisen kehitystyölle. Sovelluskehitys vähäkoodisia työkaluja hyödyntäen voi olla RAD-periaatteen mukaista, mutta RAD-kehitystyö voi olla myös perinteistä, paljon käsin kirjoitettavaa koodia sisältävää koodaamista. (Binunsky 2019.)

2000-luvulla vähäkoodisen kehityksen suosio on kasvanut jatkuvasti ja sitä myötä erilaisten kehitysalustojen määrä on kasvanut. Kansainvälinen tutkimusyhtiö Gartner arvioi vuonna 2022, että

vuonna 2023 yli puolet keskisuurista ja suurista yrityksistä hyödyntävät vähäkoodista kehitystä päivittäisessä liiketoiminnassaan. (IBM 2024.) Nykyisin vähäkoodisia ohjelmointialustoja on olemassa lukemattomia eri käyttötarkoituksiin ja erilaisille kohderyhmille. Pääsääntöisesti alustat ovat kokonaan tai lähes kokonaan yrityskäyttöön suunnattuja. Näihin kuuluvat muun muassa Microsoftin Power Apps, Appianin Low-Code Platform, OutSystemsin sekä Salesforcein Einstein 1 Platform. Kaikille kehitysympäristölle yhteistä on yritysdatan käyttö ohjelmien luomisessa ja sen tuonti helposti kaikkien saataville. Jokaisella ympäristöllä on kuitenkin omat erityispiirteensä, eivätkä ne välttämättä sovi kaikille yrityksille. (Gartner 2024.) Esimerkiksi Salesforce on tuotteissaan keskittynyt yritysten asiakkuuksien hallintaan ja sen Einstein 1 Platformin avulla luodut sovellukset on tarkoitettu Salesforcein muiden tuotteiden tueksi (Salesforce 2024), kun vastaavasti Microsoftin Power Platform on vahvasti kytköksissä Microsoftin muihin tuotteisiin (Microsoft 2024b).

Microsoftin (2024c) mukaan vähäkoodiseen kehitykseen liittyvä keskeinen haaste on kehitystyön hajaantuminen laajemmalle joukolle työntekijöitä, jolloin IT-osaston voi olla hankala tietää tarkalleen, mitä kehitystöitä on meneillään ja mihin ne vaikuttavat erityisesti tietoturvan kannalta. Vähäkoodisuus tuo mukanaan myös omat rajoitteensa kehitystyöhön. Kontrollit sisältävät tietyt ominaisuudet, joita käyttäjä voi muokata, mutta käyttäjä ei voi luoda kontrolleille uusia ominaisuuksia Microsoftin tekemien lisäksi. Ulkoasun osalta esimerkiksi painikkeiden ja laatikoiden kokoa ja sijaintia voi muokata, mutta täysin halutunlaisen ja yksilöllisen ulkoasun luonti on haastavaa. (Quixy 2023.)

Myös Kyanon Digital nostaa blogikirjoituksessaan (Kyanon Digital 2021) esiin useita vähäkoodiseen kehitykseen liittyviä ongelmia. Yksi näistä ongelmista on tietyn palveluntarjoajan tuotteisiin lukittautuminen. Vähäkoodiset sovellukset eivät välttämättä ole avointa koodia ja ne toimivan ainoastaan tietyssä ympäristössä, eikä koodi näin ollen ole hyödynnettävissä muualla. Tähän liittyvät myös mahdolliset integraatio-ongelmat. Sovellusten yhdistämisessä muihin palveluihin saatetaan selvästi suosia ohjelmointiympäristön tarjoajan muita palveluita tai integraatio muihin palveluihin on kokonaan estetty. Sovellukset voivat myös sallia yrityksen sisällä pääsyn tietoon sellaisille henkilöille, jolle se ei muuten ole sallittua, jos sovelluksen käyttöoikeuksista ei pidetä huolta. Sovellukset saattavat myös läpäistä muuten tarkan tietoturvaseulan, jos niiden luominen ja jakaminen ei ole yhtä tiukkaan rajattua kuin perinteisillä tavoilla luodut ja jaetut sovellukset.

3 MICROSOFT POWER PLATFORM

Microsoft Power Platform on vuonna 2018 alkunsa saanut yrityksille suunnattu ohjelmistokokonaisuus, joka koostuu keskenään yhteensopivista kehitystyökaluista ja -ohjelmistoista (Microsoft 2023a). Ohjelmointikielenä Power Platformissa toimii Microsoftin kehittämä avoimen lähdekoodin Power Fx -kieli, joka pohjautuu esimerkiksi Microsoft Excelissä käytettyyn Visual Basic for Applications eli VBA-kieleen. Se on vahvasti tyytety kieli, joka muistuttaa laskentataulukkoa. Power Fx:ssä kontrollien ominaisuudet yhdistyvät toisiinsa kaavojen avulla ja jonkin ominaisuuden arvon muuttuessa siihen liitetyt muut kontrollit päivittyvät automaattisesti. Esimerkiksi Power Appsilla luodussa sovelluksessa voi olla liukukytin, jolla säädetään kontrollin väriä. Liukukytintä liikuttaessa sen arvo muuttuu ja tämän myötä siihen yhdistetyn kontrollin väri muuttuu automaattisesti. Tätä voi verrata Excelissä kaavan sisältävään soluun, jossa kaavaan tuodaan tietoa muista soluista. Kun kaavaan tietoa tuovien solujen arvoja muokataan, päivittyy kaavan sisältämän solun arvo automaattisesti. (Microsoft 2023a.)

Power Platform -tuoteperheeseen kuuluvat Power Apps-, Power Bi-, Power Pages-, Power Automate- sekä Copilot Studio -työkalut. Power Apps on tarkoitettu mobiili- ja verkkosovellusten kehittämiseen, Power BI datan visualisointiin ja raportointiin ja Power Pages sivustojen toteuttamiseen. Power Automatella onnistuu prosessien automatisointi ja sitä hyödynnetään usein muissa Power Platformilla toteutetuissa ratkaisuissa. Power Automatella on mahdollista esimerkiksi automatisoida sähköpostimuistutus, kun Power Appsilla toteutetussa sovelluksessa tehdään uusi kirjaus. Copilot Studio on chattibottien luomiseen tarkoitettu tekoälyä hyödyntävä työkalu, joka tunnettiin aiemmin nimellä Power Virtual Agents. (Microsoft 2024b.) Tässä työssä keskitytään Power Platformin kahteen työkaluun: sovelluskehitykseen tarkoitettuun Power Appsiin sekä raportointiin ja datan visualisointiin tarkoitettuun Power BI -ohjelmaan.

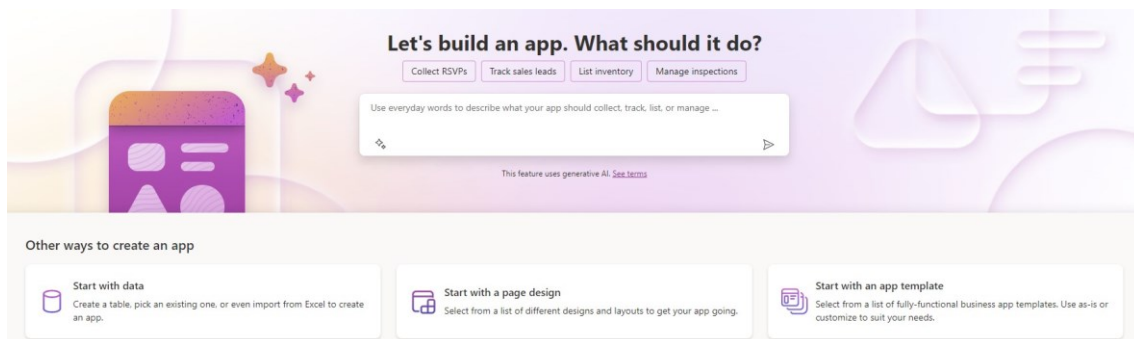
3.1 Microsoft Power Apps

Power Apps on Microsoftin loppuvuodesta 2015 esittelemä (Staples 2015) ja syksyllä 2016 julkiin käyttöön tuotu kehitystyökalu, jolla voidaan luoda selaimella, mobiililaitteilla tai Microsoft Teamsilla käytettäviä sovelluksia (Phillips 2016). Kehitystyötä on mahdollista tehdä selaimella, Teamsin kautta tai Microsoft Storen kautta ladattavissa olevalla Power Apps -sovelluksella. Power

Appsilla luodut sovellukset on pääsääntöisesti tarkoitettu yrityskäyttöön, ja ne ovat helposti yhdistettävissä muihin saman alustan työkaluihin. (Microsoft 2024c.)

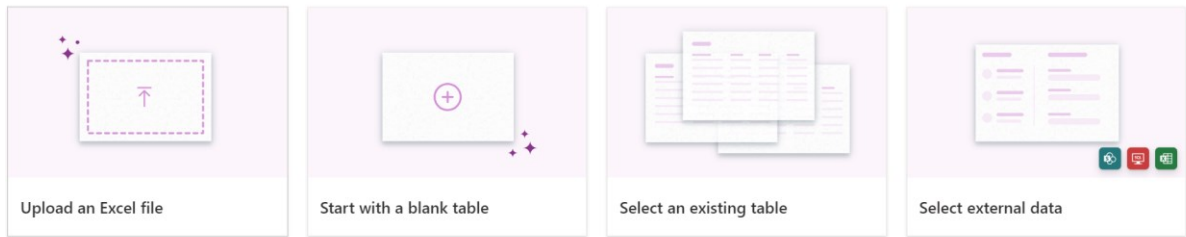
Kuvassa 3 on esitetty Power Appsin selainversion aloitusnäkyminen. Aloitusnäkymissä korostuu selvästi Microsoftin panostus Copilot-tekoälyyn, sillä sivun yläosasta selkeästi suurimman tilan vie tekstikenttä, jota voi käyttää sovelluksen luomiseen Copilotin avulla. Copilot-komentokehoteen alapuolella on kolme lisävaihtoehtoa sovelluksen luomiseksi. Vasemmalta puolelta sivua sijaitsee sovelluksen luominen datan perusteella, keskellä ulkoasuun perustuva suunnittelu sekä oikealla valmiin sovelluspohjan käyttäminen. (Microsoft 2024d.)

Ohjelman luominen Copilotin avulla aloitetaan kirjoittamalla tekstikenttään kuvaus halutun ohjelman ominaisuuksista ja käyttäjän tarpeista. Oletuksena Copilot luo sovelluksen pohjan Microsoft Dataverseen, joka on Microsoftin liiketoimintasovelluksissa käytettävien tietojen säilytykseen tarkoitettu tallennuspaikka, jota Microsoft itse kutsuu taulukkojoukoksi (Microsoft 2023c). Luodun taulukon muokkaamista voi jatkaa kirjoittamalla lisäohjeita Copilotille. Valmiin taulukon pohjalle tehdyin sovelluksen kehittämistä voi jatkaa graafisen käyttöliittymän kautta. (Microsoft 2024d.)



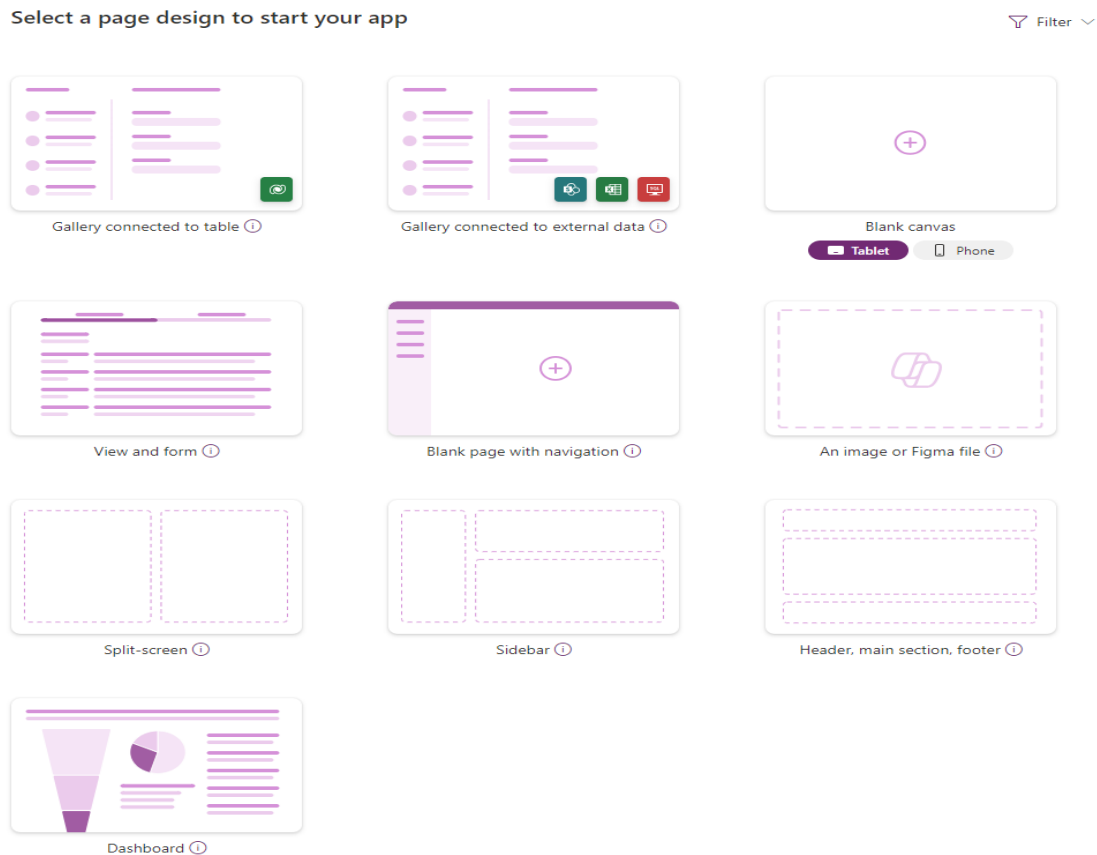
KUVA 3. Power Appsin selainversion kehitysnäkyminen

Datan pohjalta luotavalle sovellukselle on olemassa neljä aloitusvaihtoehtoa (kuva 4). Vaihtoehtoina ovat olemassa olevan Excel-tiedoston siirto Dataverseen, kokonaan uuden taulukon luominen Dataverseen, valmiiksi Dataversessä olevan taulukon käyttö tai Dataversen ulkopuolella olevan lähteen käyttö, joita voivat olla SharePoint, Excel tai SQL-tietokanta. Power Apps luo datan perusteella sovellukselle ulkoasuun ja siinä näytettävät tiedot, jotka ovat muokattavissa. Tällaista sovellusta kutsutaan mallipohjaiseksi sovellukseksi (model-driven app).



KUVA 4. Vaihtoehdot tietolähteelle luotaessa sovellusta datan pohjalta

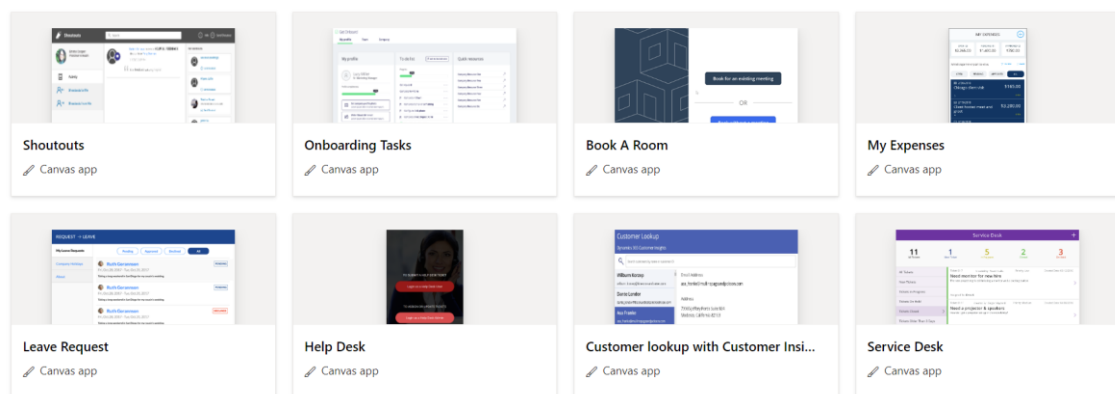
Sovelluksen pohjana voi käyttää myös ulkoasua, jolloin sovelluksen voi luoda täysin tyhjältä pohjalta tai käyttää jotain valmiista asetteluvaihtoehdoista. Tällaista sovellusta kutsutaan pohjaan perustuvaksi sovellukseksi (englanniksi canvas app). Kuvassa 5 näkyy pohjaan perustuvan sovelluksen vaihtoehdot. Oikeassa yläalaidassa näkyvä vaihtoehto Blank canvas on nimensä mukaisesti täysin tyhjä pohja, johon käyttäjä voi lisätä haluamiaan komponentteja ja halutessaan yhdistää datalähteen sovellukseen. Kaksi muuta yläarvin vaihtoehtoa mahdollistavat datalähteen yhdistämisen valmiiksi määriteltyyn ulkoasuun, kun taas muut vaihtoehdot tarjoavat erilaisia valmiita ulkoasuja ilman datalähteen liittämistä sovelluksen luontivaiheessa. (Microsoft 2024a.)



KUVA 5. Pohjaan perustuvan sovelluksen aloitusvaihtoehdot

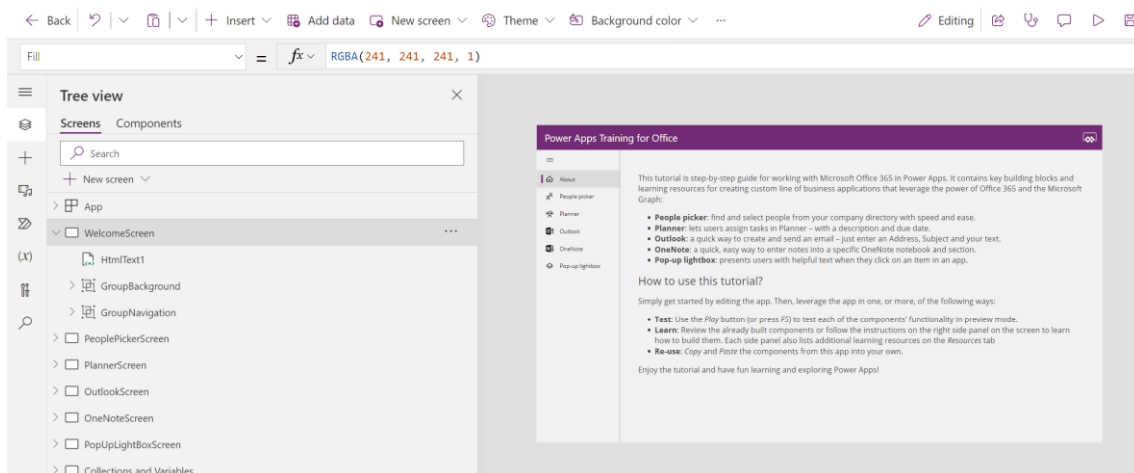
Yksinkertaistetusti voidaan todeta, että dataan pohjautuva sovellus ja pohjaan perustuva sovellus ovat toistensa vastakohtia: ensimmäisessä vaihtoehdossa sovelluksen ulkoasu luodaan datan pohjalta. Toisessa vaihtoehdossa ulkoasu suunnitellaan ensin ja sovellukseen tuotava data sovitetaan ulkoasuun. Mallipohjainen sovellus on vähemmän muokattavissa ja sen ulkoasu on pitkälti Power Appsin automaattisesti määrittämä, jonka muokkausvaihtoehdot ovat rajoitetut. (Microsoft 2024a.)

Kehitystyö on mahdollista aloittaa myös valmiin mallisovelluksen pohjalta. Malleissa toiminnallisuudet ovat jo valmiiksi rakennettuja ja käyttäjän tehtäväksi jää ainoastaan sen liittäminen haluttuun tietolähteeseen tai Office 365 -käyttäjäryhmään. Sovellusmallit ovat myös muokattavissa, jolloin käyttäjä voi poistaa niistä tarpeettomia ominaisuuksia tai lisätä niitä. (Linh 2016.) Kuvassa 6 on esitetty joitain Power Appsin tarjoamia valmiita mallipohjia, joita on yhteensä tarjolla useita kymmeniä.



KUVA 6. Power Appsin tarjoamia valmiita mallisovelluksia

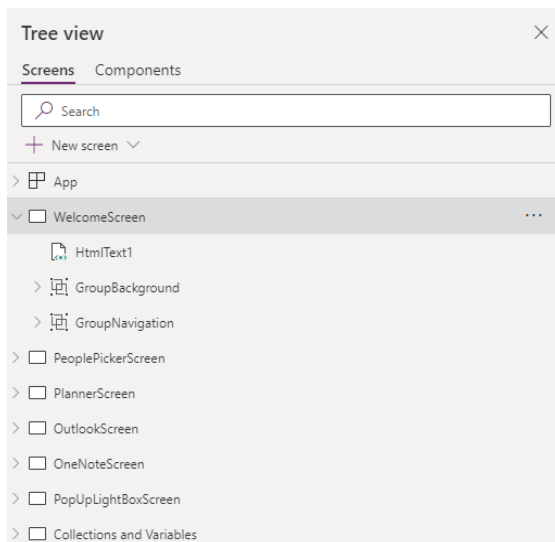
Kuvassa 7 on esitetty selaimessa toimiva graafinen käyttöliittymä, jonka kautta kehitystyö tapahtuu. Sovellukset koostuvat kontrolleista, joita käyttäjä voi lisätä ja joiden sijaintia voi muokata muokkamatta koodia. Kontrolleille voi antaa erilaisia ominaisuuksia, jotka vaikuttavat kontrollien käyttäytymiseen. Esimerkiksi nappina toimivan Button-kontrollin ominaisuus OnSelect määrittää, mitä sovelluksessa tapahtuu käyttäjän painaessa nappia. (Microsoft 2023b.)



KUVA 7. Power Appsin selainversion kehitysnäkymä

Kehitysympäristön oletusnäkökulman vasemmassa laidassa on puurakenne (kuva 8), jossa näkyy alileikkain sovelluksessa käytetyt ruudut, komponentit ja kontrollit. Kun komponentteja siirretään puunäkymässä, sovelluksen ulkoasu puunäkymän oikealla puolella muuttuu sen mukaisesti, missä osassa puuhierarkiaa komponentit ovat. Jos kehittäjä esimerkiksi siirtää jonkin komponentin puunäkymän ulkopuolelle, tulee se näkyviin puunäkymän ylä- tai alapuolelle sen perusteella, onko se puuhierarkiassa puunäkymän ylä- vai alapuolella. Puurakenteen vasemmalta puolelta voi muun muassa lisätä kontroleja, yhdistää sovelluksen tietolähteeseen, lisätä sovelluksessa käytettävää mediaa, esimerkiksi musiikkia, sekä hallita automaation liitettäviä Power Automate -ominaisuuksia. Power Automateilla on mahdollista esimerkiksi luoda automaattinen sähköposti-ilmoitus, mikäli sovelluksella tehdään uusia kirjauksia. (Microsoft 2023b.)

Puunäkymän oikealla puolella oleva kehityssikkuna näyttää sovelluksen ulkoasun reaaliajassa sen mukaan, miten sovelluksen sisältämät ruudut, komponentit ja kontrollit on näkymään sijoitettu ja mitä ominaisuuksia niille on annettu. Sovelluksen toimintoja voi testata joko painamalla oikealta ylhäältä löytyvää Play-painiketta tai pitämällä ALT-näppäintä pohjassa, jolloin esimerkiksi lomakkeiden ja nappien toiminnot voi varmistaa oikein toimiviksi. (Microsoft 2023b.)

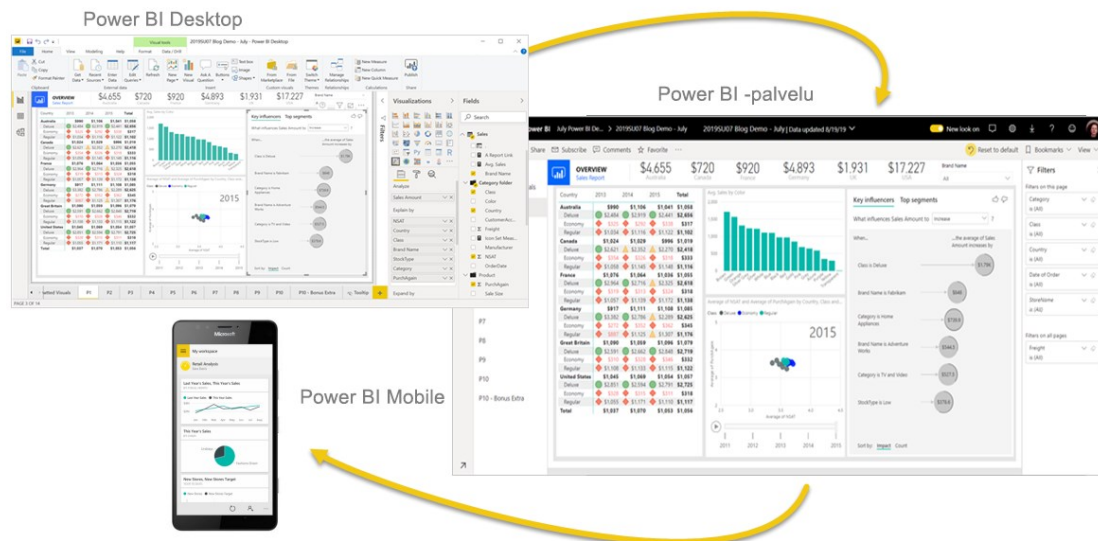


KUVA 8. Kehitettävän sovelluksen puurakenne

Ylävalikosta on mahdollista muokata nopeasti sovelluksen ulkoasua muun muassa valitsemalla valmiista teemaväreistä, vaihtaa taustan väriä tai käyttää haluamaansa kuvaa sovelluksen taustana. Ylävalikosta voi myös lisätä ruudulle kontrolleja sekä yhdistää sovellus tietolähteeseen kuten vasemmanpuoleisesta valikosta. (Microsoft 2023b.)

3.2 Microsoft Power BI

Power BI on Microsoftin ohjelmistokokonaisuus, jonka avulla käyttäjät voivat luoda ja jakaa dataan perustuvia visuaalisia raportteja. Se koostuu kolmesta perusosasta: Power BI Desktop -työpöytäsovelluksesta, Power BI -palvelu (englanniksi Power BI Service) -nimisestä SaaS-palveluna tarjottavasta verkko-ohjelmistosta sekä mobiililaitteille tarjolla olevasta Power BI -mobiilisovelluksesta. Kuvassa 9 on esitetty perusosien tyypillinen työnkulku, jossa uuden raportin luonti aloitetaan työpöytäsovelluksella, raportti viimeistellään Power BI -palvelussa ja valmista raporttia voi tarkastella mobiilisovelluksella. (Microsoft 2024e.)



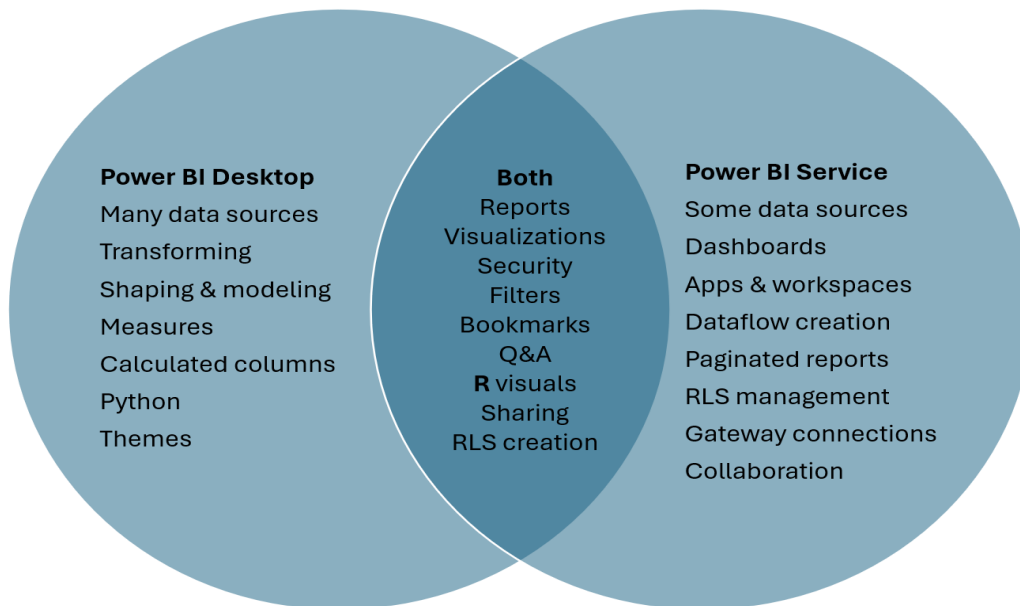
KUVA 9. Power BI:n perusosien työnkulku (Microsoft 2024d)

BI on lyhenne englanninkielisistä sanoista Business Intelligence ja sillä tarkoitetaan yrityksen olemassa olevan datan hyödyntämistä yritystoiminnassa. Sen avulla pyritään lisäämään yritysten prosessien tehokkuutta muun muassa keräämällä dataa useammasta lähteestä yhteen näkymään, josta voi seurata esimerkiksi myyntimääriä, tilauskantaa ja historiatietoja. (Microsoft 2024f).

Power BI Desktop on nimensä mukaisesti paikalliselle tietokoneelle asennettava sovellus, jonka avulla voidaan yhdistää useiden eri tietolähteiden sisältöjä. (Microsoft 2024g.) Tietolähteiden yhdistäminen ja muokkaaminen tapahtuu Power Queryn avulla, joka on Microsoftin ETL-työkalu. ETL tulee englannin kielen sanoista extract, transform ja load eli sen toiminnot koostuvat datan hausta, muuntamisesta sekä muunnetun datan lataamisesta haluttuun paikkaan, esimerkiksi tietokantaan. (Microsoft 2024h.) Tietomallin rakentamisen jälkeen työpöytäsovelluksella voi suunnitella malliin perustuvan raportin ja jakaa sen muille suoraan tai julkaisemalla sen Power BI -palvelussa (Microsoft 202g).

Power BI -palvelu on pilvipohjainen palvelu, jonka kautta voidaan muokata raportteja. Power BI -palvelun kautta on mahdollista muodostaa yhteys tietolähteisiin, mutta se on rajoitetumpaa kuin työpöytäsovelluksen kautta tehtynä. Suositeltua on tehdä raportin pohja työpöytäsovelluksen avulla, jonka jälkeen Power BI -palvelun kautta voi luoda koontinäyttöjä ja sovelluksia työpöytäsovelluksella luotuun raporttiin pohjautuen. Valmiin raportin voi jakaa suoraan Power BI -palveluun tai mobiililaitteille. Power BI -palvelu mahdollistaa yhteisten työtilojen luomisen, jossa voi olla useampi raportti ja koontinäyttö, jotka voidaan yhdistää toisiinsa ja jakaa sovelluksina. (Microsoft

2024i.) Kuvassa 10 on esitetty Power BI -palvelun ja työpöytäsovelluksen keskeisimmät erot sekä niiden yhteiset ominaisuudet Venn-diagrammin muodossa. Tiivistetysti voidaan todeta, että työpöytäsovellus keskittyy enemmän raporttien taustalla olevaan dataan ja sen käsittelyyn ja palvelun puolella keskitytään enemmän visualisointiin. (Microsoft 2024g.)



KUVA 10. Power BI Desktopin ja Power BI -palvelun vertailu

Power BI -mobiilisovellusta voidaan käyttää Android-, iOS- sekä Windows-laitteilla. Sen avulla käyttäjä voi tarkastella Power BI -palvelussa olevia raportteja suoraan sovelluksesta käyttämättä selainta. Sovellusta ei ole tarkoitettu uusien raporttien luomiseen, vaan valmiina olevien raporttien ja koontinäyttöjen tarkasteluun. (Microsoft 2024j.) Sovelluksen kautta onnistuu tietojen tarkastelu myös ilman internet-yhteyttä. Tällöin käytettävissä on aiemmin sovelluksen kautta käytetyt raportit, joita on tarkasteltu viimeisen kahden viikon aikana. Sovellus päivittää oletusarvoisesti tiedot Wi-Fi-verkon kautta kahden tunnin välein ja mobiiliverkon kautta kerran vuorokaudessa, joten lyhyen verkkokatkoksen yhteydessä käytössä on kohtalaisen ajankohtaista tietoa. Ilman verkkoyhteyttä raporteissa voi olla kuitenkin puutteita, jos jotkin osat raportista vaativat aktiivista yhteyttä palvelimeen. (Microsoft 2024k.)

4 SOVELLUKSEN JA RAPORTIN KEHITTÄMINEN

Työn ensisijaisena tavoitteena oli kehittää Power Apps -sovellus, jolla työntekijät saavat suoraan tuotannossa kirjattua tuotantoerille kohdistuneiden hukkapala-aihioiden määrän mobiilipäätteellä. Hukkapala-aihioiden käytön kirjaamisella ja seuraamisella on merkitystä esimerkiksi tuotannon vastuullisuuden kannalta, jotta raaka-aineen hävikki saadaan minimoitua. Seuranta antaa lisäksi työnjohdolle hyödyllistä tietoa raaka-aineen kulutuksesta ja siitä, että tuotannon henkilöstö muistaa hyödyntää hukkapala-aihoita.

Tuotannossa käytettävän ruostumattoman teräksen leikkauksessa oikean kokoiseksi syntyy hukkapala-aihoita, joita on mahdollista käyttää muiden tuotteiden tekemisessä. Aihoiden käytön seurantaan on ollut Excel-tiedosto (kuva 11) käytössä vuodesta 2020 asti. Tuotannon operaattorit ovat merkinneet tuotantotyön työkortille käytettyjen hukkapala-aihioiden määrän, jonka toimiston henkilökunta on siirtänyt Excel-tiedostoon tuotantoerän päättämisen yhteydessä. Sovelluksen avulla tuotannon operaattorit voivat tehdä kirjauksen hukkapala-aihioiden käytöstä suoraan tuotannossa, jolloin toimistohenkilöiden suorittama ylimääräinen työvaihe tiedon siirtämiseksi työkortilta Excelliin jää pois.

Työkortin numero	Nimike	Käytetty aihiomäärä	Aihion leveys (mm)	Aihion pituus (mm)	Neliöt (m2)	Kilot (kg)	Päivämäärä
		55	245	600	8,085	9,5	25/02/2020
		99	340	890	29,9574	35,1	24/02/2020
		22	300	450	2,97	3,5	24/02/2020
		322	160	160	8,2432	9,6	21/02/2020
		100	280	1040	29,12	34,1	20/02/2020
		8	370	300	0,888	1,0	20/02/2020
		221	270	340	20,2878	23,7	19/02/2020
		220	270	340	20,196	23,6	19/02/2020
		8	370	300	0,888	1,0	19/02/2020
		154	300	750	34,65	40,5	17/02/2020
		230	610	240	33,672	39,4	17/02/2020
		330	160	160	8,448	9,9	13/02/2020
		111	160	160	2,8416	3,3	13/02/2020
		240	280	340	22,848	26,7	12/02/2020
		260	440	300	34,32	40,2	11/02/2020
		260	220	300	17,16	20,1	11/02/2020
		179	200	300	10,74	12,6	10/02/2020
		190	125	550	13,0625	15,3	07/02/2020
		70	280	395	7,742	9,1	06/02/2020
		249	280	1350	94,122	110,1	03/02/2020
		115	280	1350	43,47	50,9	03/02/2020
		110	260	500	14,3	16,7	03/02/2020
		119	125	550	8,18125	9,6	31/01/2020
		156	290	340	15,4	18,0	28/01/2020
		290	160	160	7,4	8,7	28/01/2020
		50	125	550	3,4	4,0	27/01/2020

KUVA 11. Alkuvuodesta 2020 käyttöön otettu Excel-seurantatiedosto

Sovelluksen lisäksi haluttiin luoda Power BI -raportti perustuen sovelluksella tehtyihin kirjauksiin, jotta käytettyjen hukkapala-aihioiden määrän seuranta esimerkiksi kuukausi- ja vuositasolla olisi

nopeaa ja sujuvaa. Suodattimien avulla raportista voisi myös hakea nimikkeitä, joita yleensä valmistetaan hukkapala-aihioista. Tästä tiedosta on hyötyä esimerkiksi tuotantolaitteiden vikatilanteissa, jolloin uusien aihoiden teko ei onnistu, vaan tuotannossa joudutaan turvautumaan hukkapala-aihioiden käyttöön. Raportin tekeminen myös edesauttaa työn tarkoituksena olleen Power Platformin kokonaisvaltaisessa hyödyntämisessä pelkän sovelluksen teon sijaan.

4.1 Vaatimusmäärittely

Sovelluksen vaatimusmäärittely oli suoraviivaista, koska useamman vuoden käytössä ollut Excel-tiedosto antoi hyvän pohjan sovelluksessa tarvittaville ominaisuuksille. Sovelluksen kautta tehtävien kirjausten tulisi sisältää tuotantotilauksella oleva työkortin numero, joka luodaan tuotantotilausta vapautettaessa automaattisesti toiminnanohjausjärjestelmässä, valmistettavan nimikkeen tuotekoodi, käytettyjen aihoiden määrä, käytetyn aihion leveys ja pituus millimetreinä sekä päivämäärä. Käytetyn aihiomäärän sekä aihion leveyden ja pituuden avulla saadaan laskettua kulutetut aihiot neliöinä. Neliöiden perusteella saadaan laskettua kulunut kilomäärä kertomalla neliömäärä kertoimella 0,86, joka perustuu käytetyn teräslaadun tiheyteen ja paksuuteen. Ruostumaton teräs hankitaan kiloina, joten raportoinnin kannalta kiloiksi muokkaaminen on välttämätöntä, jotta hukkapala-aihioiden käytöllä säästetty rahamäärä saadaan laskettua.

Power Appsin sovelluspohjana voidaan käyttää suoraan Exceliä, mutta jo käytössä ollutta Exceliä ei haluttu käyttää, koska siinä oli vuosien aikana kertynyt turhaa dataa muun muassa toimimattomien kaavojen ja vanhojen laskentojen muodossa. Sovelluksella tehtävät kirjaukset ovat varsin suoraviivaisia, eikä esimerkiksi tietokannan luomista sovellusta varten nähty tarpeelliseksi. Sovelluksella tehtävien kirjausten tallennuspaikaksi valittiin Microsoftin SharePoint useiden syiden vuoksi: dokumenttien jakamiseen ja hallintaan tarkoitettu SharePoint oli jo yrityksessä käytössä esimerkiksi sisäisen Intran muodossa, joten sen käytöstä oli runsaasti kokemusta. Excel-tiedostojen käyttö nähdään yrityksessä vanhana tapana, josta pyritään mahdollisuuksien mukaan luopumaan modernimpien ratkaisuiden tieltä. Lisäksi SharePointin kautta on mahdollista selata Teams-kanavia, jolloin kirjaukset voisivat olla tallennettuna suoraan PTC-liiketoiminta-alueen Teams-kanavan alla. SharePointin luettelo-ominaisuus (englanniksi SharePoint List) mahdollistaa tiedon tallentamisen Excelin kaltaiseen muotoon, joten kirjausten tallentaminen saatiin SharePointin luetteloa käyttämällä hyvin samankaltaiseksi sinällään edelleen toimivan vanhan Excel-tiedoston kanssa.

Sovelluksessa vaadittuja ominaisuuksia uuden kirjauksen tekemiselle olivat myös tehdyn kirjauksen muokkaaminen sekä poistaminen. Sovellukseen haluttiin myös hakutoiminto, jolla tehtyjä kirjauksia voisi etsiä työkortin numeron ja nimikekoodin perusteella. Sovelluksen tulisi olla käyttöliittymältään yksinkertainen, jotta sen käyttö pieneltäkin mobiililaitteelta onnistuisi tuotantotöiden yhteydessä sujuvasti.

Power BI -raportin avulla haluttiin seurata kuukausi- ja vuositasolla käytettyjen hukkapala-aihioiden määrää sekä niiden avulla säästettyä rahamäärää. Raporttiin haettavat tiedot tulisivat SharePoint-luetteloon operaattoreiden tekemien kirjausten perusteella ja raaka-aineen hinta saataisiin toiminnanohjausjärjestelmästä. Raportissa tulisi olla suodattimet kuukausi- ja vuositason tarkastelulle sekä oma laskentansa säästetylle rahamäärälle, jonka laskenta muuttuisi raportin tarkastelijan valitsemien kuukausien ja vuosien mukaisesti.

4.2 SharePoint-luettelon luominen

Ennen sovelluksen luontia tuli luoda SharePoint-luettelo, johon operaattoreiden tekemät kirjaukset tallentuvat sekä josta Power BI -raportin data haettaisiin. Pohjana käytettiin vanhaa käytössä ollutta Exceliä, koska siinä olleet sarakkeet olivat nimeämislähtöisiä toimivia, eikä niitä näin ollen koettu tarpeelliseksi muuttaa. SharePoint mahdollistaa luettelon luonnin yhteydessä muun muassa käyttämään olemassa olevaa Excel-tiedostoa pohjana, joten luetteloon saatiin tuotua Excel-tiedostosta sekä valmiiksi nimetyt sarakkeet että vanhat kirjaukset. Vanhat kirjaukset haluttiin mukaan uuteen luetteloon, jotta Power BI -raporttiin saataisiin tiedot myös menneiltä vuosilta. Ainoat muokkaukset luettelon rakenteeseen jouduttiin tekemään käytettyjen aihoiden neliömäärien ja kilojen kanttien, koska niissä olleita kaavoja SharePoint ei osannut tuoda Excelistä. Neliö- ja kilomääriä operaattorit eivät syötä sovellukseen, vaan niiden laskenta suoritetaan operaattoreiden tekemien kirjausten perusteella suoraan luettelossa määrittämällä neliö- ja kilomäärien sarakkeisiin tarvittavat kaavat: neliömäärä lasketaan kertomalla käytetty aihiomäärä aihion leveydellä ja pituudella sekä jakamalla saatu tulos miljoonalla leveyden ja pituuden ollessa millimetreinä. Vastaavasti kilomäärä lasketaan kertomalla neliömäärä luvulla 0,86.

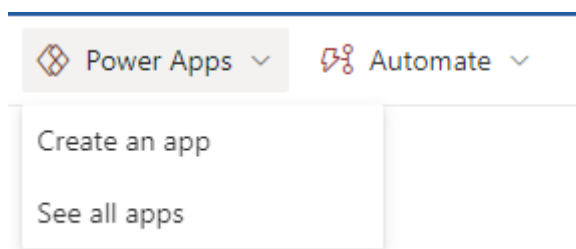
Ennen luettelon käyttöönottoa piti vielä muokata järjestystä, jossa uudet kirjaukset luettelossa tulevat näkyviin. Oletusarvona luettelossa näkyi ylimmäisenä vanhimmat kirjaukset. Tämä muokattiin siten, että uusimmat järjestykset näytetään luettelon ylimmäisenä. Mikäli luettelon pohjalta luotu sovellus ei jostain syystä toimisi, olisi uudet kirjaukset ja muokkaukset olemassa oleviin kirjauksiin tehtävä luettelon kautta ja silloin olisi käytön kannalta helpointa, että uusimmat kirjaukset olisivat heti näkyvillä luettelon avattaessa. Valmis luettelo kirjauksineen on esitetty kuvassa 12.

Työkortin numero	Nimike	Käytetty ahiomäärä	Aihion leveys (mm)	Aihion pituus (mm)	Neliöt (m2)	Kilot (kg)	Päivämäärä
		218	340	280	20,75	24,28	7.3.2024
		312	220	300	20,59	24,09	6.3.2024
		112	225	500	12,60	14,74	6.3.2024

KUVA 12. Valmiin SharePoint-luettelon rakenne

4.3 Power Apps -sovelluksen kehittäminen

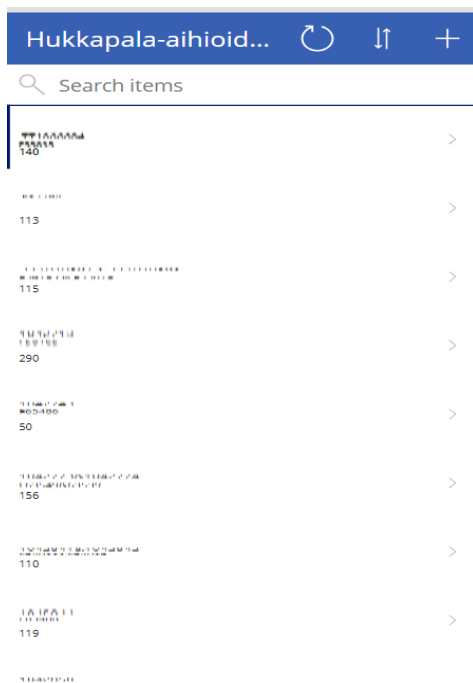
Koska suurimmalla osalla yrityksen henkilöstöstä ei ole kokemusta ohjelmoinnista, pyrittiin sovellus toteuttamaan mahdollisimman vähällä vaivalla, jotta vastaavien sovellusten luominen tulevaisuudessa olisi mahdollista mahdollisimman monen henkilön toimesta. SharePoint-luettelon yläpalkissa on Power Apps -välilehti, josta löytyy suoraan Create an App -toiminto sovelluksen luontia varten luettelon pohjalta (kuva 13).



KUVA 13. Sovelluksen luominen suoraan SharePointin luettelosta

Sovelluksen ensimmäinen versio (kuva 14) luotiin suoraan SharePointista, eli sovellus tehtiin suoraan olemassa olevan datan pohjalta. Lopputulos ei kuitenkaan ollut halutunlainen. Napin painaminen loi automaattisesti sovelluksen ja esimerkiksi luettelosta tuotavia sarakkeita ei saanut valit-

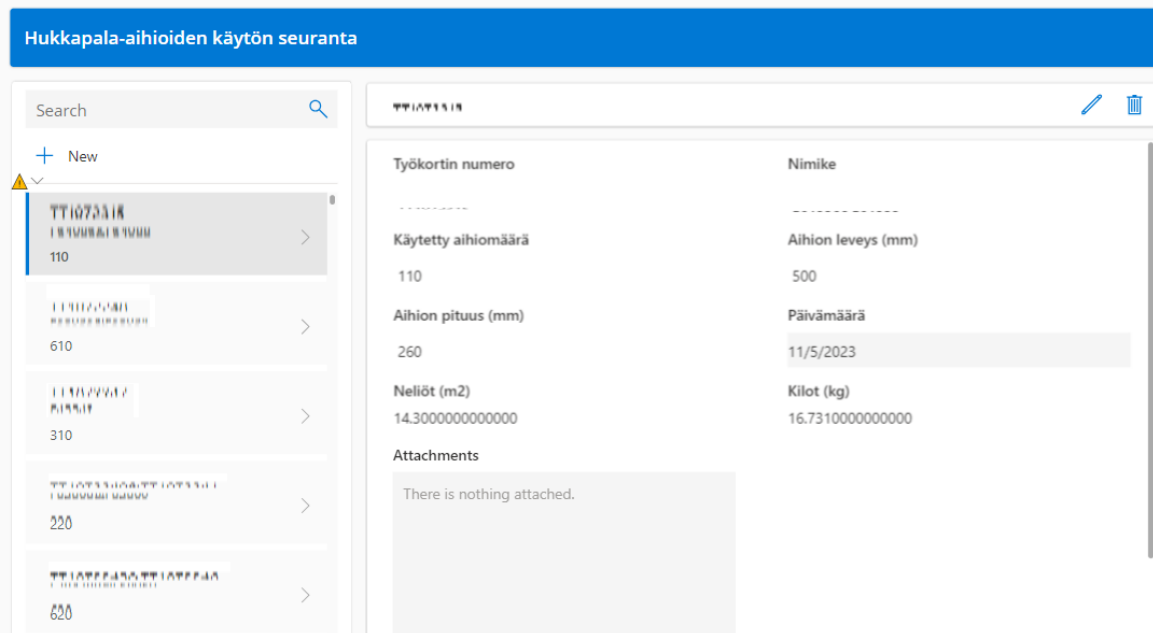
tua, vaan ne tuotiin sellaisenaan Power Appsiin. Sovelluksen kannalta turhat kilot- ja neliöt-sarakkeet tulivat myös mukaan, vaikka niissä tehtävä laskenta suoritetaan suoraan SharePointissa, eikä niihin tehdä kirjauksia sovelluksen kautta. Lisäksi sovelluksen tallennussijaintia ei erikseen kysytty, vaan se meni automaattisesti yrityksen Power Appsin oletussijaintiin, mikä ei välttämättä ole varsinkaan sovelluksen kehitysvaiheessa järkevin tallennussijainti. Myös sovelluksen käyttöliittymä jätti toivomisen varaa. Se oli toki yksinkertainen, mutta esimerkiksi hakutoiminto ei toiminut halutulla tavalla ja tehdyt kirjaukset näyttivät olevan listassa satunnaisessa järjestyksessä, eikä sovelluksessa ollut kirjausten suodatus vaikuttanut toimivan järkevällä tavalla. Sovelluksen perustana oli yksi ikkuna, jossa näytettävät tiedot muuttuivat sen perusteella, mitä sovelluksessa teki. Oletuksena näkymässä näkyi muutama tieto riveittäin luettelosta haetuista kirjauksista ja uuden kirjauksen tekeminen aukesi oletusnäkyvän päälle. Tämä versio sovelluksesta päätettiin hylätä sen vaatiman muokkausmäärän vuoksi.



KUVA 14. SharePointin luettelosta automaattisesti luotu sovellus

Sovelluksen toinen versio päätettiin luoda Power Appsin verkkosivuston kautta Sharepointin sijaan käyttämällä pohjaan perustuvaa sovellusta lähtökohtana. Sovelluksen ulkoasuksi valittiin niin sanottu dataan yhdistetty galleria (englanniksi Gallery connected to external data), jossa sovellus koostuu kahdesta pääosasta: vasemmalla olevasta galleriasta, jossa näkyy tietolähteestä halutut tiedot sekä oikealla olevasta isommasta laatikosta, josta tarkastellaan jo olemassa olevien tietueiden tarkempia tietoja tai syötetään uusia kirjauksia. Sovellus valmistui muutamassa sekunnissa

suoraan SharePointista luomisen tapaan. Myös tällä tavalla luotu sovellus jätti kysymättä sovellukseen tuotavat sarakkeet ja näin ollen sovellukseen tuli sen toiminnan kannalta tarpeettomat sarakkeet mukaan. Lisäksi uusia kirjauksia varten sovellukseen ilmestyi oma kenttänsä liitteitä varten, mitä ei SharePointin luettelossa ole mainittuna. Sovelluksen ulkoasu (kuva 15) erosi merkittävästi ensimmäisestä versiosta ja uuden kirjausten tekeminen oli mahdollista menettämättä näkymää jo olemassa oleviin kirjauksiin. Myös Power Appsin kautta luotu sovellus vaati verrattain paljon muokkaamista, jotta sen saisi halutunlaiseksi, mutta toimivamman oloisen käyttöliittymän vuoksi kehitystyötä päätettiin jatkaa tämän version pohjalta. Tässä versiossa sovelluksen käyttöliittymän vasemmalla laidassa on luettelonäkymä, jossa näkyvät SharePoint-luettelon kirjat ja sen yläpuolella mahdollisuus lisätä uusi kirjaus sekä hakutoiminto. Suurimman osan käyttöliittymästä vie oikealla laidalla oleva päänäkö, jossa on tarkemmat tiedot luettelonäkymästä valitusta kirjauksesta ja uutta kirjausta tehdessä päänäkö sisältää tyhjän lomakkeen tietojen syöttöä varten.



KUVA 15. Power Apps -verkkosivuston kautta luotu sovellus

4.4 Luettelonäkymä sovelluksessa

Power Appsin automaattisesti luoma Gallery-kontrolli sovelluksen vasemmalla laidalla sisälsi virheilmoituksen, jonka mukaan käytössä ollut Search-funktio ei välttämättä toimi oikein, kun luettelo sisältää yli 500 riviä. Luomishetkellä luettelossa oli noin 800 riviä kirjauksia. Power Apps sallii rajan noston 2000:een, mutta sen määrän yli hakutoiminto ei ainakaan tällä hetkellä tuloksia näytä. Tämä

aiheuttaa sen, että mikäli haettava kirjaus on luettelossa esimerkiksi rivillä 2010, ei se näy sovelluksen kautta luettelossa edes hakutoiminnon kautta. Koska 2000 kirjausta tulisi jossain vaiheessa todennäköisesti täyttymään, ei rajan nostaminen 2000:een olisi pysyvä ratkaisu. Tämän vuoksi hakutoiminto muokattiin käyttämään Filter-funktiota, joka kykenee näyttämään periaatteessa rajattoman määrän rivejä.

Luettelossa olevat rivit oli suodatettu työkortin numeron perusteella aakkosjärjestykseen, mutta todellisuudessa rivit vaikuttivat olevan satunnaisessa järjestyksessä. Sovelluksen toiminnan kannalta loogisinta olisi, että uusimmat kirjaukset näkyisivät listalla ylimpänä, kuten ne oli jo aiemmin muokattu näkymään Sharepointin luettelossa. Sekä hakufunktion muuttaminen että rivien suodattamisen korjattiin Gallery-kontrollin Items-ominaisuuden kautta. Items-ominaisuus määrittelee datalähteen, mistä galleriassa näytettävät asiat haetaan, ja sen avulla voidaan lisäksi lajitella ja suodattaa gallerian sisältöä eri funktioita käyttäen. Luettelossa näytettävien rivien järjestykseksi asetettiin Sort-funktion avulla päivämäärä laskevassa järjestyksessä ja Filter-funktion avulla hakukentän kautta voi etsiä joko työkortin numeron tai nimikkeen perusteella haluamaansa kirjausta. Filter-funktiossa käytettiin StartsWith-funktiota, koska se pystyy ohittamaan 2000 näytettävän rivin rajoituksen. Suodatuksen voi tehdä myös etsimällä haettua tekstiä mistä tahansa kohtaa kirjauksia, mutta tällöin näytettävät rivit on rajoitettu 2000 kappaleeseen. Esimerkkikoodi valmiista Filter-funktiosta on esitetty kuvassa 16.

```
Sort(  
    Filter(  
        'Hukkapala-aihioiden käytön seuranta',  
    StartsWith(  
        'Työkortin numero', SearchInput.Text)  
    Or  
    StartsWith(  
        'Nimike', SearchInput.Text)  
    ),  
    Päivämäärä, Descending  
)
```

KUVA 16. Luettelossa näytettävien kirjausten suodatus

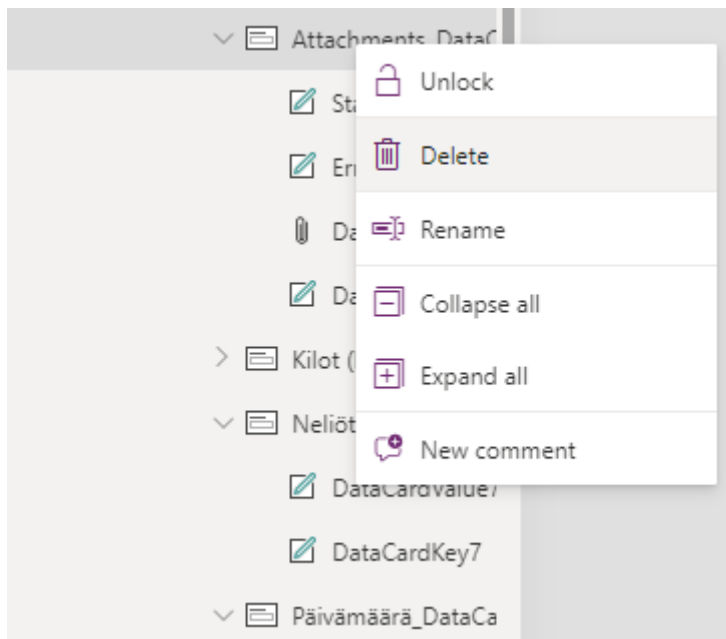
Kahden edellä mainitun muokkauksen lisäksi luettelossa otsikkotasolla näytettäviä tietoja piti muokata. Oletuksena listauksessa näkyivät työkortin numero, tuotteen nimike sekä käytetty aihiomäärä. Kirjauksia selatessa käytetty aihiomäärä ei ole oleellinen tieto, vaan sitä tärkeämpi tieto on nähdä

kirjauksen päivämäärä, jota käytetään myös luettelon järjestyksen määrittämisessä. Tekstilaatikoissa näytettävät tiedot määritetään Text-ominaisuuden kautta. Gallerian ollessa yhdistettynä Sharepoint-luetteloon voi tekstikentän määrittää näyttämään mitä vain arvoa luettelosta viittaamalla siihen ThisItem-operaattorilla, esimerkiksi ThisItem.SarakkeenNimi. Näin ollen aihion määrän muuttaminen päivämääräksi saatiin aikaan yksinkertaisesti muuttamalla tekstilaatikon Text-ominaisuuden ThisItem.Käytetty aihiomäärä muotoon ThisItem.Päivämäärä. Sovellus käytti oletuksena päivämäärästä muotoa kuukausi, päivä, vuosi. Tämä saatiin muokattua suomalaiseen muotoon lisäämällä Text-ominaisuuden ThisItem-operaattorin perään "dd.mm.yyyy".

4.5 Päänäkymä sovelluksessa

Sovelluksen päänäkymä on toteutettu Power Appsin Formin avulla, joka on nimensä mukaisesti lomakepohja, jolle tuodaan halutut tiedot yhdistetystä datalähteestä. Lomakepohjalla olevia tietoa sisältäviä kenttiä kutsutaan DataCardiksi. Formilla on omat kontrollinsa tiedon näyttämiseen ja sen muokkaamiseen sekä uusien kirjausten syöttämiseen. Sovelluksessa oli oletuksena kaikki kolme kontrollia ja ne haluttiin siinä myös pitää. Oletusnäkyvässä vasemmalta kirjauksen valitessa näkyi tehdyn kirjauksen tiedot, eikä niitä pystynyt muokkaamaan. Tällöin Display Form -kontrolli on aktiivisena. Oikealta löytyvä muokkauspainike asettaa Edit Form -kontrollin Edit-tilaan, jolloin valittua kirjausta pystyy muokkaamaan. Vasemmalta löytyvä uuden kirjauksen painike asettaa Edit Form -kontrollin New-tilaan, jolloin päänäkymän lomake tyhjenee tiedoista ja käyttäjä voi täyttää kentät uusilla tiedoilla. Uuden kirjauksen tekeminen asettaa lomakkeen jälleen Display Form -kontrollin aktiiviseksi.

Lomakepohja sisälsi oletuksena ylimääräisiä kenttiä, jotka haluttiin poistaa sovelluksesta. Neliö- ja kilometrisarakkeet olivat turhia sovelluksen toiminnan kannalta sekä sovelluksen luomisvaiheessa automaattisesti lisätty liitteiden lisäämisen mahdollistava kenttä. Näiden poistaminen onnistui yksinkertaisesti kehitysympäristössä vasemmalta löytyvän puurakenteen kautta painamalla poistettavaa DataCardia hiiren oikealla painikkeella ja valitsemalla Delete (kuva 17). Poistaminen onnistui nopeasti, eikä sovellukseen jäänyt esimerkiksi virheviestejä puuttuvista osista, joten Power Apps poisti automaattisesti poistettaviin osiin liittyvät viittaukset sovelluksen muista osista.



KUVA 17. Lomakkeen DataCardin poistaminen

Päivämäärän sisältävä DataCard vaati muutoksia hieman samaan tapaan kuin aiemmin tehdyt muokkaukset luettelonäkymässä. Päivämäärän valinnassa ja näyttämisessä Power Apps hyödyntää DatePicker-kontrollia. Oletuksena päivämääräkentässä oli viikon ensimmäisenä päivänä sunnuntai sekä päivämäärä näkyi muodossa kuukausi, päivä, vuosi. Päivämäärän sai muokattua kuntoon DataCardin Advanced-välilehdeltä muokkaamalla StartOfWeek-ominaisuuden muotoon StartOfWeek.Monday ja päivämäärän oikeaan muotoon asettamalla Format-ominaisuuden muotoon “dd.mm.yyyy”.

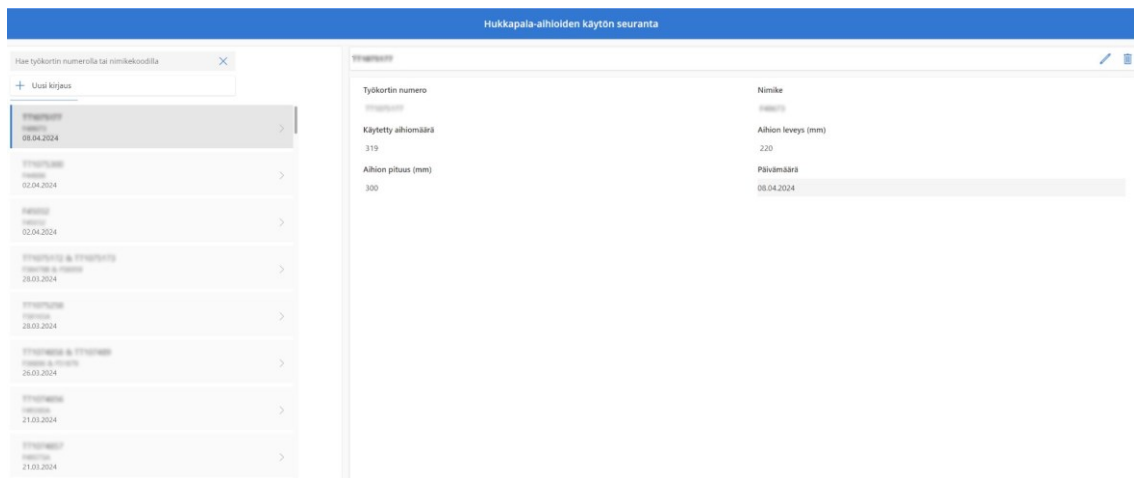
Oletuksena yksikään lomakkeen kentistä ei ollut pakollinen täytettäväksi uutta kirjausta tehdessä. Kentät määritettiin pakollisiksi jokaisen DataCardin Advanced-välilehdeltä asettamalla Required-ominaisuus True-tilaan. Muokkauksen jälkeen vaaditut kentät toimivat oikein, mutta kentän ollessa tyhjä ruutuun ilmestyvä geneerinen virheviesti oli suomea ja englantia sekaisin. Koska sovelluksen haluttiin olevan täysin suomenkielinen, täytyi virheilmoitukset muokata sopivammiksi. Tämä toteutettiin muokkaamalla sovelluksessa jokaisella DataCardilla valmiiksi ollutta ErrorMessage-viestikenttää. Viestikentän Text-ominaisuus oli muodossa Parent.Error, jolloin virheviestiksi muodostuu Power Appsin oletusarvona oleva DataCardin nimi ja sen perään teksti “is required”. Kuvassa 18 on esitetty työkortin numero -kentän muokattu virheviestin koodi, jossa tarkistetaan, onko Parent.Error -ominaisuus tyhjä ja mikäli se ei ole tyhjä, näytetään virheviesti, jossa pyydetään syöttämään vaadittuun kenttään tiedot.

Text

```
If(Not(IsBlank(Parent.Error)), "Syötä työkortin numero.")
```

KUVA 18. Muokatun virheviestin koodi

Viimeisenä muokattavana osana oli sovelluksen ulkoasu. Käyttöliittymä oli poistettujen kenttien jälkeä looginen, mutta päänäkyä jäi ruudun alaosasta erittäin tyhjän näköiseksi erityisesti käytettäessä sovellusta tietokoneen näytöllä (kuva 19).

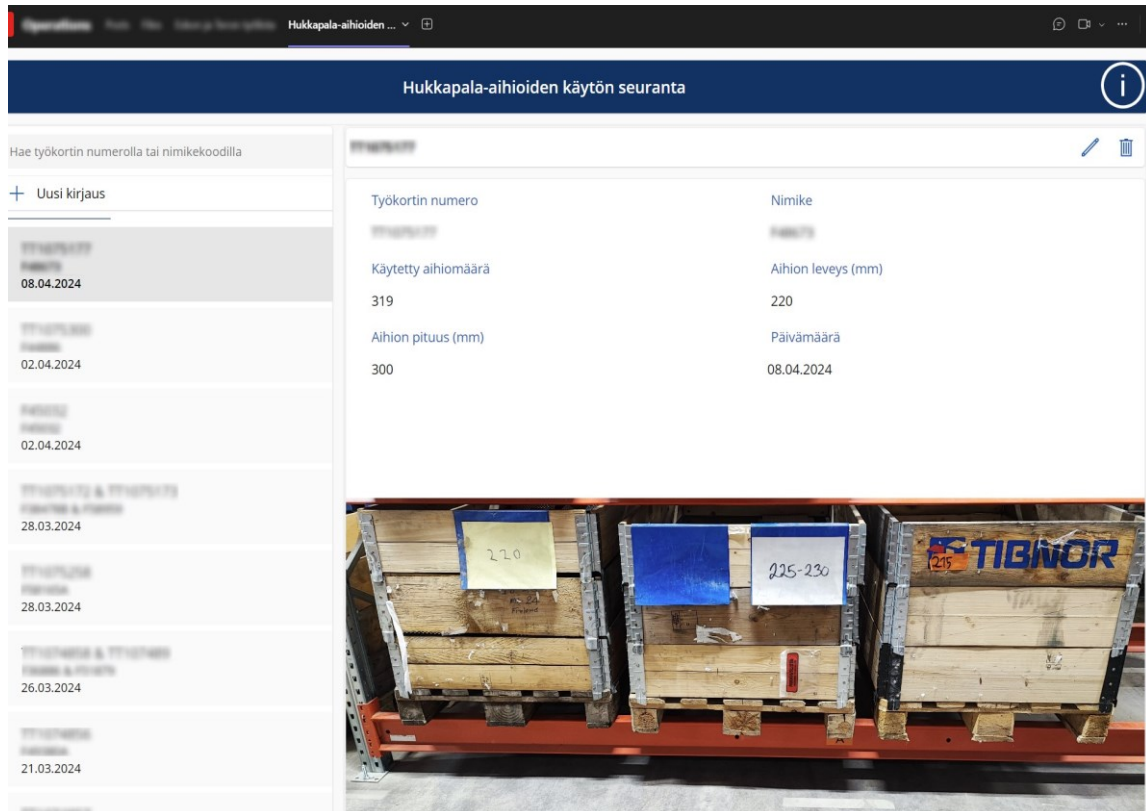


KUVA 19. Sovelluksen käyttöliittymä toiminnallisten muokkauksien jälkeen

Tyhjän tilan täyttämiseksi päänäkyään lisättiin kuva hukkapala-aihoita sisältävistä pakkauslaivoista. Kuvan lisääminen onnistui kehitysympäristössä ylälaidasta löytyvällä Insert Image -toiminnolla ja viemällä kuva puuhierarkiassa päänäkyän lomakkeen alapuolelle. Sovelluksen teemaväriksi valikoitui tummansininen ja valkoinen. Teeman vaihtaminen tehtiin kehitysympäristön ylälaidasta löytyvästä Theme-valikosta, jossa on noin 20 valmista väriteemaa valittavana. Lisäksi sovelluksen ulkoasuun tehtiin lukuisia pienempiä muokkauksia esimerkiksi fonttikokoon ja eri komponenttien kokoon liittyen.

Valmis sovellus julkaistiin käyttöön kehitysympäristön oikeassa yläkulmassa olevasta Publish-painikkeesta. Julkaistua sovellusta voi käyttää suoraan nettiselaimesta eri laitteilla tai lisäämällä sovelluksen Teams-kanavalle. Selaimen kautta tapahtuva käyttö vaatii linkin sovellukseen sekä linkin aukaisijalta käyttöoikeuden sovellukseen. Teams-kanavalle integroidun sovelluksen käyttämiseen

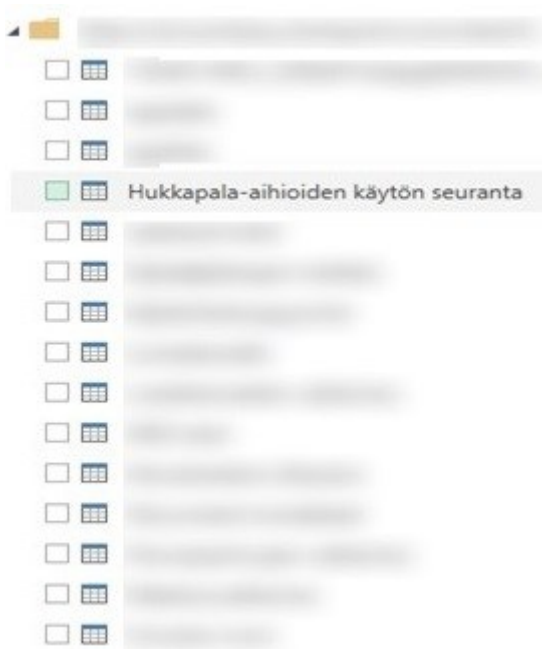
riittää Teams-kanavan jäsenyys, ellei sovelluksen käyttöä erikseen haluta rajoittaa tietyille ryhmälle. Teams-kanavalle integroitu sovellus on esitetty kuvassa 20.



KUVA 20. Valmis sovellus integroituna Teams-kanavalle

4.6 Power BI -raportin kehittäminen

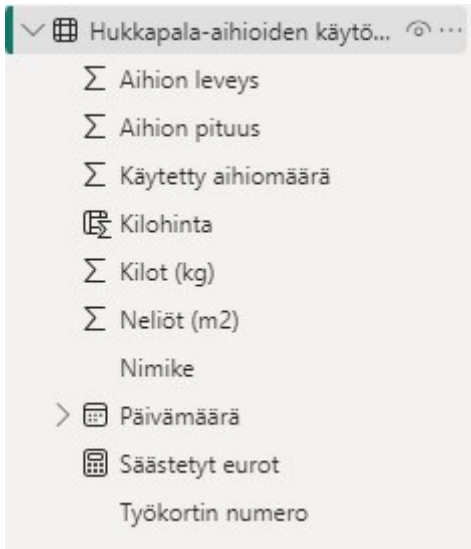
Power BI -raportin kehittäminen aloitettiin käyttämällä Power BI Desktop -työpöytäsovellusta. Raportin teon ensimmäinen vaihe oli yhdistää sovellusta varten luotu SharePoint-luettelo datalähteeksi, josta raportille tulevat tiedot haettaisiin. Yksittäistä luetteloa ei ainakaan toistaiseksi ole mahdollista yhdistää datalähteeksi suoraan, vaan Power BI hakee kaikki sille syötetystä osoitteesta löytyvät SharePoint-luettelot, joista voi valita joko yksittäisen tai halutessaan useamman luettelon (kuva 21).



KUVA 21. Oikean SharePoint-luettelon valinta raportin tietolähteeksi

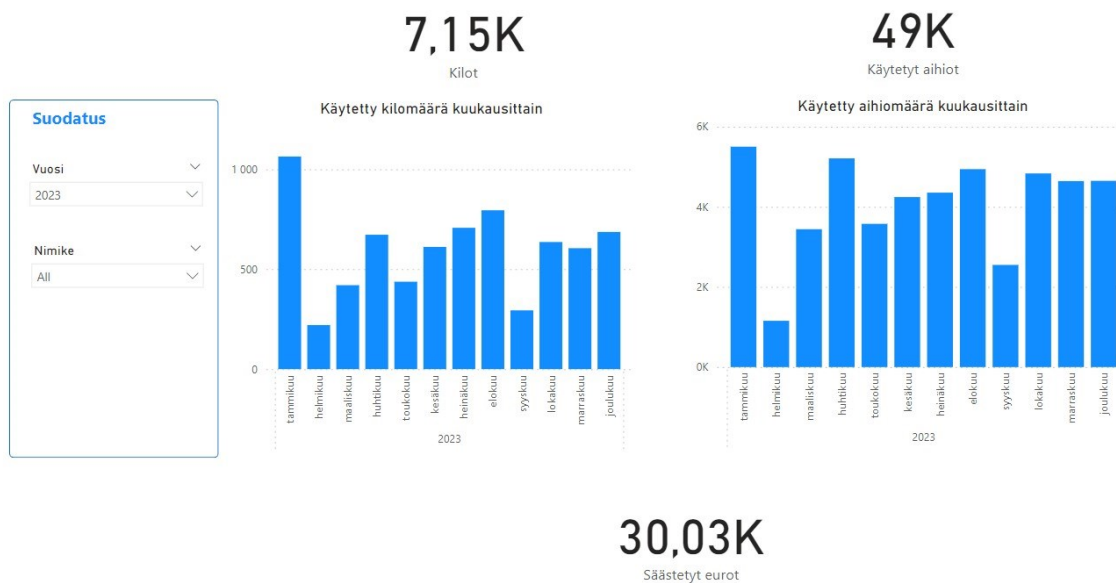
Onnistuneen datalähteen yhdistämisen jälkeen aloitettiin datan visualisointi. Tuotua dataa jouduttiin kuitenkin muokkaamaan ennen sen hyödyntämistä raportoinnissa. Kaikki SharePoint-luettelosta tuodut sarakkeet olivat tietotyybiltään tekstikenttiä, jonka perusteella raportin vaatimaa laskentaa ei voi suorittaa. Esimerkiksi päivämääräkentän arvon tunnistaminen oikein on erittäin tärkeää raportin toiminnan kannalta, koska sen perusteella voidaan tarkastella esimerkiksi kuukausitasolla käytettyjen hukkapala-aihioiden määrää. Päivämääräkentän lisäksi aihion leveyden, aihion pituuden, käytetyn aihiomäärän, käytettyjen kilojen sekä käytettyjen neliöiden kentät piti muuttaa numeromuotoisiksi kentiksi sekä niiden laskentatyyppi summaksi. Laskentatyyppiin täytyy olla summa, jotta voidaan tarkastella esimerkiksi vuositasolla käytettyjen hukkapala-aihioiden neliöiden määrää.

Dataan piti tehdä myös lisäyksiä, jotta raportista saatiin kaikki halutut ominaisuudet irti. Käytössä olleesta toiminnanohjausjärjestelmästä saatiin ruostumattoman teräksen neliöhinta, joka tuotiin ylimääräisenä sarakkeena olemassa olevan datan yhteyteen. Neliöhinnan ja käytettyjen aihoiden neliömäärän perusteella pystytään laskemaan hukkapala-aihioiden käytöllä säästetty rahamäärä siihen verrattuna, jos kaikki hukkapala-aihiot olisivat menneet roskeeseen. Valmis tietomalli puurakenteena on esitetty kuvassa 22.



KUVA 22. Raportin tietomalli puurakenteen muodossa

Koska raportista haluttiin mahdollisimman selkeä, helppokäyttöinen ja helppolukuinen, valittiin tietojen näyttämiseen pylväskaaviot sekä niiden tueksi datakortit. Pylväskaavioita raportille (kuva 23) luotiin kaksi kappaletta, joista toinen näyttää valitulla aikavälillä käytettyjen hukkapala-aihioiden kokonaiskilomäärän kuukausittasolla ja toinen käytettyjen aihioiden kokonaismäärän. Pylväskaavioiden X-akselilla on kuukaudet ja Y-akselilla lukumäärät. Kaavioiden tueksi tehdyt datakortit sisältävät numeroarvona valitun aikavälin kokonaiskilomäärän, käytettyjen aihioiden kokonaismäärän sekä euromäärän, joka hukkapala-aihioiden käyttämisellä on säästetty.



KUVA 23. Valmis Power BI -raportti

Raportilla näytettävän tiedon suodattamista varten raportin vasempaan laitaan lisättiin kaksi osittajaa, joiden avulla raportin käyttäjä voi valita tarkastelujaksoksi haluamansa vuoden tai vuodet sekä suodattaa tulokset koskemaan ainoastaan tiettyjä nimikkeitä. Kun osittajasta valitsee vuoden, raportti näyttää sekä pylväskaavioissa että datakorteissa oletusarvoisesti koko vuoden tiedot. Käyttäjä voi tarkastella tietyn kuukauden tietoja klikkaamalla pylväskaaviosta tiettyä kuukautta, jolloin kaikki raportin osiot päivittyvät näyttämään valitun kuukauden tietoja. Useamman kuukauden tietojen tarkastelu onnistuu painamalla ensin control-näppäimen pohjaan näppäimistöstä ja tämän jälkeen klikkaamalla hiirellä halutut tarkastelukuukaudet pylväskaaviosta.

5 POHDINTA

Opinnäytetyön tavoitteena oli luoda ylijäävän raaka-aineen hyötykäytön seurantasovellus sekä sovelluksella tehtyjen kirjausten pohjalta visuaalinen raportti hyödyntäen Microsoftin vähäkoodisia Power Platform -kehitystyökaluja. Työ toimi eräänlaisena pilottiprojektina toimeksiantajayritykselle sen sisäisessä sovelluskehityksessä antaen kuvaa vähäkoodisen kehitystyön potentiaalista nykyisten työmenetelmien modernisoimisessa.

Microsoft Power Platform ei ollut kehityksen kannalta aiemmin tuttu, joten työn tekeminen lähti liikkeelle etsimällä perustietoa alustan työkaluista ja niiden toiminnallisuuksista. Microsoftin omat nettisivut tarjosivat suhteellisen hyvät sekä laajat oppaat ja hyviä ohjeita löytyi muutenkin laajasti internetistä, muun muassa YouTubesta. Oppaat antoivat mielestäni hyvät lähtökohdat onnistua yksinkertaisten ratkaisujen luomiseen myös sellaisille henkilöille, joilla ei ole aiempaa kokemusta ohjelmoinnista. Työkaluihin tutustumisen jälkeen kehitystyö oli suhteellisen nopeaa ja suoraviivaista, mikä on yksi vähäkoodisen kehityksen periaatteista. Kokonaistyöaikaa sovelluksen luomiseen kului muutamia kymmeniä tunteja.

Valmis Power Apps -sovellus saavutti sille määritetyt tavoitteet: sovelluksella voi tehdä uusia kirjauksia ja sillä voi tarkastella, muokata sekä poistaa vanhoja kirjauksia. Sovellus on ollut rajoitetussa testikäytössä kevään 2024 aikana, eikä sen toiminnassa ole havaittu ongelmia. Testijakson aikana vanhasta Excel-kirjanpidosta on luovuttu ja kaikki kirjaukset tehdään nykyään sovelluksen kautta SharePointiin. Sovellus on tarkoitus julkaista koko tuotantohenkilöstön käyttöön kesän 2024 aikana.

Myös Power BI -raporttiin saatiin halutut ominaisuudet toteutettua ja sen yksinkertaisuuden vuoksi kehitystyö oli sovellusta nopeampaa ja siihen kulunut kokonaistyöaika jäi alle kymmeneen tuntiin. Käytetyn datan yksinkertaisuuden sekä sen perusteella tehdyn laskennan yksinkertaisuus ei välttämättä antanut hyvää kuvaa laajemman raportoinnin vaatimasta työstä, mutta sen avulla saatiin pintaraapaisu Power BI -työkalujen käyttöön. Toimeksiantajayrityksessä vuoden 2024 alkupuolella käynnistyneen raportoinnin toimittajavaihdoksen vuoksi raporttia ei olla vielä julkaistu yleisesti saataville.

Kokonaisuudessaan työn toteuttaminen antoi melko hyvän kuvan Power Platformin tarjoamista mahdollisuuksista. Sen avulla ei saa luotua erityisen monimutkaisia sovelluksia, mutta se tuntuisi soveltuvan erinomaisesti helposti omaksuttavien työkalujen ansiosta yrityksen sisällä tapahtuvaan kehitystyöhön, jolla saadaan nopeutettua esimerkiksi nykyisin Excelissä tapahtuvaa työtä. Säännöllisesti julkaistavien uusien ominaisuuksien ja päivitysten perusteella Microsoft tulee panostamaan Power Platformin kehitystyöhön myös jatkossa.

Jatkokehityksen suhteen ei olla vielä tehty päätöksiä, mutta yksi mielenkiintoinen kokeilu olisi tuoda raportille tietoa suoraan toiminnanohjausjärjestelmästä SharePointin luettelon lisäksi. Toiminnanohjausjärjestelmästä saataisiin tieto muun muassa raaka-aineen kokonaiskulutuksesta, jonka avulla raportilla voitaisiin tarkastella muun muassa hukkapala-aihioiden ja normaalien aihoiden käytön suhdetta tuotannossa. Sovelluksen osalta on mietitty vastaavanlaisen sovelluksen kehittämistä työkalujen huollon seurantaan. Power Apps mahdollistaa kirjausten yhteyteen lisättävät liitteet, jolloin esimerkiksi huoltoon tulevasta työkalusta ja siihen liittyvästä viasta voisi sovelluksen kautta lisätä kuvia helpottamaan huoltotyötä.

LÄHTEET

Attido 2024. Low-code. Hakupäivä 14.3.2024. <https://www.attido.com/fi/low-code/>

Binunsky, Dimitri 2019. How Low-Code and No-Code Solutions Compare to RAD. Hakupäivä 23.3.2024. <https://devops.com/how-low-code-and-no-code-solutions-compare-to-rad/>

Gartner 2024. Enterprise Low-Code Application Platforms. Hakupäivä 6.3.2024. <https://www.gartner.com/reviews/market/enterprise-low-code-application-platform>

IBM 2024. What is Low-Code? Hakupäivä 13.3.2024. <https://www.ibm.com/topics/low-code>

Kissflow 2024. The History of Low-Code Platforms: How Development Changed. Hakupäivä 7.3.2024. <https://kissflow.com/low-code/history-of-low-code-development-platforms/>

Kyanon Digital 2021. 10 Challenges of Enterprise Low-code. Hakupäivä 14.3.2024. <https://medium.com/@kyanon.digital/10-challenges-of-enterprise-low-code-e801a6691447>

Microsoft 2022. Overview of creating apps in Power Apps. Hakupäivä 4.2.2024. <https://learn.microsoft.com/en-us/power-apps/maker/>

Microsoft 2023a. Microsoft Power Fx -yleiskatsaus. Hakupäivä 4.2.2024. <https://learn.microsoft.com/fi-fi/power-platform/power-fx/overview>

Microsoft 2023b. Yleiskatsaus sovellusten luomisesta Power Appsissa. Hakupäivä 10.2.2024. <https://learn.microsoft.com/fi-fi/power-apps/maker/>

Microsoft 2023c. Mikä on Microsoft Dataverse? Hakupäivä 13.3.2024. <https://learn.microsoft.com/fi-fi/power-apps/maker/data-platform/data-platform-intro>

Microsoft 2024a. Mitä on vähäkoodinen kehitys? Hakupäivä 11.2.2024. <https://powerapps.microsoft.com/fi-fi/what-is-low-code/>

Microsoft 2024b. Microsoft Power Platform. Hakupäivä 4.2.2024. <https://www.microsoft.com/fi-fi/power-platform#products>

Microsoft 2024c. Microsoft Power Apps. Hakupäivä 4.2.2024. <https://www.microsoft.com/fi-fi/power-platform/products/power-apps>

Microsoft 2024d. Build apps through conversation. Hakupäivä 20.2.2024. <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/ai-conversations-create-app>

Microsoft 2024e. Power BI:n kuvaus. Hakupäivä 18.2.2024. <https://learn.microsoft.com/fi-fi/power-bi/fundamentals/power-bi-overview>

Microsoft 2024f. What is business intelligence? Hakupäivä 18.2.2024 <https://powerbi.microsoft.com/en-us/what-is-business-intelligence/>

Microsoft 2024g. Power BI Desktopin ja Power BI –palvelu vertailu. Hakupäivä 19.2.2024. <https://learn.microsoft.com/fi-fi/power-bi/fundamentals/service-service-vs-desktop>

Microsoft 2024h. What is Power Query? Hakupäivä 19.2.2024. <https://learn.microsoft.com/en-us/power-query/power-query-what-is-power-query>

Microsoft 2024i. Mikä on Power BI -palvelu? Hakupäivä 19.2.2024. <https://learn.microsoft.com/fi-fi/power-bi/fundamentals/power-bi-service-overview>

Microsoft 2024j. Mitä Power BI -mobiilisovellukset ovat? Hakupäivä 19.2.2024. <https://learn.microsoft.com/fi-fi/power-bi/consumer/mobile/mobile-apps-for-mobile-devices>

Microsoft 2024k. Tietojen tarkastelu offline-tilassa Power BI -mobiilisovelluksissa. Hakupäivä 19.2.2024. <https://learn.microsoft.com/fi-fi/power-bi/consumer/mobile/mobile-apps-offline-data>

Microsoft 2024l. Rapid application development for beginners. Hakupäivä 8.3.2024. <https://powerapps.microsoft.com/en-us/rapid-application-development-rad/>

Phillips, James 2016. Microsoft PowerApps and Flow are generally available starting tomorrow. Hakupäivä 4.2.2024. <https://blogs.microsoft.com/blog/2016/10/31/microsoft-powerapps-flow-generally-available-starting-tomorrow/>

Proventia Oy 2024a. Yritys. Hakupäivä 2.2.2024. <https://www.proventia.com/yritys>

Proventia Oy 2024b. Proventian lämpöeristeet. Hakupäivä 3.2.2024. <https://www.proventia.com/index.php?391>

Proventia Oy 2024c. Valokuva. Lämpöeristeratkaisut. Hakupäivä 3.2.2024. <https://www.proventia.com/lampoeristeet/lampoeristeratkaisut>

Salesforce 2024. Einstein 1 Platform. Hakupäivä 9.3.2024. <https://www.salesforce.com/eu/products/platform/overview/>

Tran, Linh 2016. Create an app from PowerApps templates. Hakupäivä 20.2.2024. <https://powerapps.microsoft.com/es-es/blog/using-app-template/>

Quixy 2023. Exploring the Pros and Cons of Low-Code Development. Hakupäivä 17.2.2024. <https://quixy.com/blog/pros-and-cons-of-low-code-development/>

Staples, Bill 2015. Introducing Microsoft Power Apps. Hakupäivä 4.2.2024. <https://blogs.microsoft.com/blog/2015/11/30/introducing-microsoft-powerapps/>