Karelia-ammattikorkeakoulu
Tradenomi (AMK), Tietojenkäsittely

Generatiivisen tekoälyn hyödyntäminen low-code-kehityksessä

Microsoft Power Platform -teknologiat

Anne Ihanus

Opinnäytetyö, toukokuu 2025

www.karelia.fi

Karelia AMMATTIKORKEAKOULU	**OPINNÄYTETYÖ** **Toukokuu 2025** **Tietojenkäsittelyn koulutus** Tikkarinne 9 80200 JOENSUU +358 13 260 600

Tekijä
Anne Ihanus

Nimeke
Generatiivisen tekoälyn hyödyntäminen low-code-kehityksessä : Microsoft Power Platform -teknologiat

Tiivistelmä

Opinnäytetyön aiheena oli generatiivisen tekoälyn hyödyntäminen low-code-kehityksessä Microsoft Power Platformin tuotteilla. Tavoitteena oli saada yleiskäsitys low-codekehityksestä ja generatiivisen tekoälyn käyttämisestä sekä arvioida sen etuja ja haittoja.

Kyseessä oli toiminnallinen työ, jossa tutkittiin Power Platformin ja generatiivisen tekoälyn ominaisuuksia laatimalla kolme yksinkertaista demoa. Tuotettujen demojen teknologioiksi valittiin Power Apps ja Power Automate, tietolähteeksi Dataverse ja tekoälyavustajaksi Copilot. Demojen kehittämisen lisäksi toteutukseen kuului Power Platform -sertifiointi, tietokantataulujen laatiminen sekä generatiivisen tekoälyn tuotosten arviointi. Kehitetyillä demoilla ei ollut todellista liiketoiminnallista tarvetta, joten niiden toimintaa ei testattu kattavasti.

Yksittäisillä työkaluilla kehitetyt low-code-toteutukset sopivat parhaiten yksinkertaisten prosessien automatisointiin. Power Platform mahdollistaa useiden erilaisten teknologioiden yhdistämisen samaan toteutukseen, mikä laajentaa ratkaisujen käyttötapoja. Laajojen kokonaisuuksien kehittäminen voi kuitenkin olla haastavaa ja vaatii laaja-alaista ymmärrystä käytettävien teknologioiden ominaisuuksista. Generatiivisen tekoälyn käyttäminen voi toisaalta nopeuttaa kehitystyötä yksinkertaisissa tehtävissä, mutta toisaalta luoda virheellistä sisältöä ja tuottaa odottamattomia tuloksia.

Kieli suomi	Sivuja 37 Liitteet Liitesivumäärä

Asiasanat
ohjelmointi, tekoäly, ihmisen ja tietokoneen vuorovaikutus

Karelia
UNIVERSITY OF APPLIED SCIENCES

THESIS
May 2025
Degree Programme in
Business Information Technology

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600

Author
Anne Ihanus

Title
Using Generative Artificial Intelligence with Low-Code Development : Microsoft Power Platform Technologies

Abstract

The topic of this thesis was to utilize generative artificial intelligence in low-code development with Microsoft Power Platform technologies. The aim was to expand the knowledge about low-code development and estimate the pros and cons of using generative artificial intelligence.

The implementation of this practice-based thesis consisted of the following stages: First, the characteristics of Power Platform and generative artificial intelligence were examined through the development of three simple demo solutions. Power Apps and Power Automate were selected to be used as low-code technologies, Dataverse as a database technology and Copilot as a virtual assistant. Second, becoming certified in Power Platform certification program was included in this project. In addition, data tables for demo solutions were created, and outputs of generative artificial intelligence were evaluated.

Low-code solutions developed with singular tools are best suited for automating simple processes as they might not include all important features, but Power Platform supports integrating multiple different technologies in one solution. However, developing large solutions with low-code tools might be challenging, as they require a broad understanding of the properties of the technologies used. On the other hand, using generative artificial intelligence to assist development can accelerate the process of obtaining results, but it can also create invalid content and lead to unexpected outcomes.

Language
Finnish

Pages 37
Appendices
Pages of Appendices

Keywords
programming, artificial intelligence, human-computer interaction

Sisältö

1 Johdanto

Low-code-kehittäminen ja tekoälyn kehittyminen ovat olleet paljon esillä viime aikoina. Valitsin tämän aiheen oppiakseni enemmän sekä Microsoftin Power Platformista että tekoälystä ja hyödyntääkseni näitä oppeja työelämässä. Tekoälyn nopea kehittyminen ja sen vyöryminen arkiseenkin elämään vaativat aiheen opiskelua viimeistään nyt. Ala tarvitsee työntekijöitä, jotka osaavat hyödyntää tekoälyä ja ymmärtävät myös sen vaatimat resurssit.

Toiminnallisessa opinnäytetyössäni toteutin demoja Microsoft Power Platformin kehitystyökaluilla generatiivista tekoälyä hyödyntäen. Tämän vuoksi tietoperustassa käsitellään vain valitsemiani Microsoftin teknologioita. Opinnäytetyön rajaamisen vuoksi tekoälyn historiasta ja kielimalleista kerrotaan vain hyvin yleisellä tasolla keskittyen generatiiviseen tekoälyyn sekä tekoälyn kehityksen tuomiin ajankohtaisiin asioihin.

Raportissa käytetään joistakin termeistä sekä suomen- että englanninkielistä sanaa, sillä termi ei välttämättä ole vakiintunut suomenkieliseen muotoon. Näissä tapauksissa alkuperäinen termi mainitaan suluissa. Microsoft Learn näyttää nojaavan verkkosivujensa automaattisessa suomentamisessa hieman kömpelöön tekoälyyn, joten termien oikeellisuuden tarkastaminen Learnin omilta sivuilta ei ole luotettavasti mahdollista.

Seuraavissa kahdessa kappaleissa esitellään opinnäytetyön tietoperusta, jonka jälkeen kuvaillaan opinnäytetyön toteutusvaihe. Toteutetuissa demoissa ei ole pyritty ottamaan huomioon kaikkia liiketoiminnallisia näkökulmia, joten ne eivät kata kaikkia tilanteita esimerkiksi virheenkäsittelyn kannalta, vaan ne ovat tarkoitettu teknologioiden opetteluun. Toteutusvaiheen jälkeen kuvaillaan opinnäytetyön tulokset. Lopuksi pohditaan opinnäytetyön onnistumista ja prosessin aikana kertyneen tietämyksen vaikutuksia tuleviin ratkaisuihin.

2 Tekoäly

2.1 Mitä tekoäly on?

Tekoäly on moniulotteinen käsite, jonka menetelmät ulottuvat useaan tieteenalaan, ja vastaavasti se vaikuttaa useaan toimialaan. Lisäksi tekoälyn kehitystä ohjaavat poliittiset päätökset sekä eettiset näkökulmat – moraali ja arvot. (Ailisto, Heikkilä, Helaakoski, Neuvonen & Seppälä 2018, 6.)

Termin *Artificial Intelligence* esitteli ensimmäistä kertaa yhdysvaltalainen tietojenkäsittelytieteen professori John McCarthy vuonna 1956 (Siukonen & Neittaanmäki 2019, 25). Tuohon aikaan koneiden laskutehot olivat pieniä, mutta laskennan teorian ja neuroverkkojen tutkimisen perusteella uskottiin, että koneet saadaan oppimaan. Tekoälystä muodostui tieteenala. (Ziemann 2016.)

Tekoäly-termin ensiesittelystä lähtien informaatioteknologia on muuttanut ihmis- ja yhteiskuntaa valtavasti, ja tekoälyn määrittelyyn liittyvät menetelmät, merkitykset ja tavoitteet ovat sen myötä ajan saatossa muuttuneet. Voidaan kuitenkin todeta, että tekoäly on tietokoneen tuottamaa keinotekoista älykkyyttä. Se on ikään kuin perinteisen tietokoneen jatkeena toimiva järjestelmä tai ohjelma, jonka kyky oppia sille syötetystä datasta tekee siitä ihmiselle tuottavan avustajan. (Siukonen & Neittaanmäki 2019, 28–31.)

Euroopan komission määritelmässä tekoäly jaetaan

a) ohjelmistoihin: kasvojen- ja puheentunnistusjärjestelmiin, hakukoneisiin, kuvananalysointiohjelmistoihin ja virtuaalisiin avustajiin, sekä niin sanottuun
b) ruumiillistettuun tekoälyyn: asioiden internetiin (IoT), drooneihin, itseohjautuviin autoihin ja robotteihin. (Euroopan parlamentti 2023.)

Tekoälyn kouluttamiseksi vaaditaan valtavat määrät tietoa. Luonnollista kieltä ja kokonaisia lauseita ymmärtäviä ratkaisuja kutsutaan laajoiksi kielimalleiksi. Ne koostuvat neuroverkoista ja käyttävät syväoppimisen tekniikoita kuten

transformer-arkkitehtuuria. (IBM 2023a.) Transformerit sisältävät enkooderin ja/tai dekooderin: enkooderi luokittelee, järjestelee ja analysoi syötteen ja muuttaa sen sisäiseksi esitykseksi sanojen välisten riippuvuuksien perusteella, ja dekooderi tuottaa sisältöä ennustamalla todennäköisyyksiä syötteestä saadun tiedon perusteella (Barberá 2025, 7–8).

2.2 Generatiivinen tekoäly

Generatiivista tekoälyä voidaan kutsua myös luovaksi tai tuottavaksi tekoälyksi: se on malli, joka tuottaa uutta sisältöä. Sisältö voidaan tuottaa kirjoitetuksi tekstiksi, kuviksi, videoiksi tai ääneksi ja siksi sitä voidaankin käyttää erilaisissa avustavissa tehtävissä. Tällaisia tehtäviä voivat olla esimerkiksi tekstin muuttaminen puheeksi, puheen muuttaminen tekstiksi, erilaiset tekstinkäsittelyt kuten kääntäminen tai tiivistelmien laatiminen, sekä erilaiset virtuaaliavusteiset ohjelmistot kuten chatbotit ja ohjelmoinnin tai mediatuotannon avustaminen. (Sitra 2025.)

2.3 Tekoäly ja eettiset kysymykset

Tekoälyn etiikka voidaan tulkita joko tekoälyn kehittämiseen sovellettaviksi eettisiksi kysymyksiksi tai tekoälyn itsensä luomaksi ja sovellettavaksi toimintatavaksi (Ollila 11, 2019). Teknologiayritys IBM on käyttänyt tekoälystä termiä apuäly, koska sen eettisten periaatteiden mukaan on olennaista muistaa, että tiedon hallitsemisen arvojärjestyksessä ihminen päättää, mikä on hyväksi. Tietokone on avustaja ja suorittaa sille annetut tehtävät. (Siukonen & Neittaanmäki 2019, 29.)

Tekoälyn suunnittelu ja sen kouluttamiseen käytetyn datan laatu vaikuttavat sen tuottamaan tuloksiin. Sekä käytetty data että tekoälyn tuotos saattaa sisältää tahattomia tai tarkoituksellisia vääristymiä, jolloin sen käyttö voi johtaa vinoutuneeseen lopputulokseen, vaikka monimutkainen esitys usein näyttää täsmälliseltä ja tosiasioihin perustuvalta. (Euroopan parlamentti 2025a.) Tekoälyn tuottamasta vääristymästä käytetään myös termiä hallusinointi. Tarkoituksellinen vääristymä johtuu yleensä ilkeämielisestä toiminnasta, jossa tekoälyn

tuottamaan tietoon pyritään vaikuttamaan manipuloimalla tekoälyn käyttämää lähdetietoa. (IBM 2023b.)

Laajojen kielimallien käyttö vaatii yleensä tekoälyavustajien pääsyn käyttäjän tietoihin kuten selainhistoriaan, sähköpostiin ja muihin viestintäsovelluksiin, kalenteriin sekä jopa kolmannen osapuolen järjestelmiin. Näin laaja pääsy tietoihin lisää tietovuodon riskiä, joten käyttäjän on osattava arvioida, tuottaako tekoälyavustajan käyttäminen tarpeeksi hyötyä riskeihin nähden. (Barberá 2025, 14–15.)

On myös huomioitava, että tietoturvariskit ovat läsnä koko tekoälymallin elinkaaren ajan. Tekoälyn suunnitteluvaiheessa valitut tietolähteet voivat sisältää sensitiivistä tietoa. Valmisteluvaiheessa tieto voi olla puutteellisesti anonymisoitua ja johtaa ongelmiin seuraavissa vaiheissa. Tietolähteiden vääristymät voivat myös johtaa tekoälyn virheelliseen toimintaan, mikä voi aiheuttaa loukkauksia yksilöiden perusoikeuksiin. Tekoälyn kouluttamisessa ja käyttämisessä käytetty tieto voi tallentua pysyvästi, jolloin myös tietovuoto on mahdollista. Tietojen epätäydellinen poistaminen käytöstä poiston yhteydessä voi johtaa pitkäkestoisiin tietoturvahaavoittuvuuksiin. (Barberá 2025, 24–25.)

Yksi merkittävä eettinen ongelma on tekoälyn ympäristövaikutukset. Ne ovat huomattavasti suuremmat kuin esimerkiksi tavallisen verkkohaun: vuonna 2024 yksi kysymys ChatGPT:lle vastasi noin kymmenen Google-haun energiankulutusta. Erityisesti generatiivisen tekoälyn koulutuksen ja laskennan vuoksi laskentatehojen tarve kasvaa, mikä johtaa datakeskusten kasvavaan energiankulutukseen. Koska eurooppalaisten kielimallien kouluttamiseen panostaminen on välttämätöntä, kehityksessä tulee huomioida sosiaalinen ja ekologinen kestävyys. (Alanko & Palo 2024.)

2.4 Tilanne nyt ja tulevaisuuden näkymät

Maailman ensimmäinen tekoälylainsäädäntö on hyväksytty EU:ssa kesäkuussa 2024. Sen tehtävänä on varmistaa EU:ssa käytettävien tekoälyohjelmistojen turvallisuus, tasa-arvoisuus, jäljitettävyys, ympäristöystävällisyys sekä

läpinäkyvyys. Asetuksen mukaan tekoälyohjelmistojen aiheuttamat riskit on arvioitava, ja riskiluokitukseen perustuen asetetaan erinäisiä velvoitteita sekä järjestelmien valmistajille että käyttäjille. (Euroopan parlamentti 2025b.)

Uuden lainsäädännön mukaan EU:ssa kielletään tekoälyn toimesta tehtävä käyttäjän manipuloiminen, sosiaalinen pisteytys, ihmisten biometrinen tunnistaminen ja luokittelu sekä reaaliaikainen biometristen tunnistusjärjestelmien käyttö julkisissa tiloissa. Nämä mainitut luokitellaan ei hyväksyttävän riskin toiminnoiksi ja kielto astui voimaan helmikuussa vuonna 2025. Biometristen tunnistusjärjestelmien käyttö voidaan kuitenkin sallia lainvalvonnallisissa tilanteissa oikeuden päätöksellä. Seuraavaan ryhmään, eli suuren riskin luokkaan kuuluvat tekoälyjärjestelmät, jotka voivat vaikuttaa kielteisesti perusoikeuksiin tai turvallisuuteen. Näitä ovat muun muassa autot, lääkinnälliset laitteet, hissit, kriittisen infrastruktuurin hallinta, työllistäminen, lainvalvonta sekä demokraattiset prosessit. Näitä tulee arvioida ennen markkinoille pääsyä sekä jatkuvasti koko tuotteen elinkaaren ajan. Lisäksi kansalaisilla on oltava mahdollisuus tehdä ilmoituksia kyseisistä järjestelmistä kansallisille viranomaisille. Näiden osalta lakia aletaan soveltaa täysin kolme vuotta lain voimaanastumisen jälkeen. (Euroopan parlamentti 2025b.)

EU:n tekoälylainsäädännössä generatiivista tekoälyä ei luokitella suuren riskin järjestelmäksi, mutta sitä koskee EU:n tekijänoikeuslaki sekä tietyt avoimuusvaatimukset. Tuotetussa sisällössä on kerrottava, mikäli se on tekoälyn tuottamaa. Tekoälyjärjestelmä on kehitettävä siten, että laittoman sisällön tuottaminen on mahdotonta. Sen tulee julkaista myös yhteenvetoja kouluttamiseen käytettävästä tiedosta, mikäli se on tekijänoikeuksin suojattua. (Euroopan parlamentti 2025b.)

Sitran asiantuntija Sanna Rekola kirjoittaa Opetushallitukselle laatimassaan artikkelissa (Opetushallitus 2025) megatrendeistä: ”Tekoäly näyttelee yhä suurempaa roolia elämän eri osa-alueilla, ja data on noussut yhdeksi aikamme tärkeimmistä talouden raaka-aineista”. Rekola pohtii, kuinka yksilöt voisivat ylläpitää media- ja teknologialukutaitoa sekä digitaalista sivistyneisyyttä, kun digitalisaatio etenee ja ”teknologia sulautuu kaikkeen”. Jokaisen tulisi ymmärtää,

kuinka verkossa toimitaan vastuullisesti ja yksityisyyttä kunnioittaen, sekä oppia tunnistamaan tiedon oikeellisuus ja vaikuttamisyritykset.

Yksi Sitran julkaisemista viimeisimmistä megatrendeistä onkin digivallan kilpailun kiihtyminen. Digitaalinen valta sisältää tiedon omistajuuden, digimaailman pelisäännöt sekä jopa teknologian tulevan kehityssuunnan. Tekoäly dominoi digitaalista ympäristöä ja herättää kysymyksiä muun muassa tiedon käyttämisestä ja ympäristövaikutuksista. Muut megatrendit liittyvät luonnon kantokyvyn vaarantumiseen, hyvinvoinnin haasteiden lisääntymiseen, demokraattisten instituutioiden kohtaamiin uhkiin sekä talouden perustan haasteisiin. Näissäkin tekoälyllä on osansa, esimerkiksi teknologian kehittymisestä ja terveysdatan luotettavasta käytöstä odotetaan kustannustehokkaita ratkaisuja hyvinvoinnin haasteisiin. (Dufva 2024.)

3 Low-code-kehitys

3.1 Low-code yleisesti

Nimensä mukaisesti low-code-kehitys tarkoittaa sovellusten kehittämistä visuaalisella työkalulla, jossa varsinaista koodaamista tarvitaan vähän tai ei lainkaan. Low-code-alustoja ovat kehittäneet muun muassa Outsystems, Google ja Microsoft. (Ite wiki 2025.) Tässä opinnäytetyössä keskityn Microsoftin kehittämän Power Platformin ominaisuuksiin.

Low-code-kehitys on perinteistä ohjelmointia nopeampaa. Tämä nopeuttaa myös testaamista ja tuloksiin reagoimista. Resursoinnin ja kustannustehokkuuden kannalta low-code-osaaminen luonnollisesti mahdollistaa myös alkeistason koodaustaidon omaavien henkilöiden rekrytoinnin. Tosin kovin monimutkaisiin ratkaisuihin se ei ole sopiva vaihtoehto, vaikka useat ratkaisut tarjoavatkin valmiita integroitumismahdollisuuksia suosituimpiin tietokantoihin ja palveluihin. (Ite wiki 2025.)

3.2 Microsoft Power Platform -tuotteita

3.2.1 Päätuotteet

Microsoft Power Platform koostuu neljästä päätuotteesta, joita ovat Power Apps selain- ja mobiilisovellusten kehittämiseen, Power Automate työnkulkujen automatisointiin, Power BI datan analysointiin ja visualisointiin sekä Power Pages verkkosivujen luomiseen. Lisäksi Microsoftin tarjoamassa on muita työkaluja, joilla voidaan parantaa edellä mainittujen tuotteiden tehokkuutta. Näitä ovat muun muassa Copilot Studio chatbottien luomiseen, yhdistimet (*connectors)* ratkaisujen yhdistämiseksi lähde- ja kohdejärjestelmiin tai palveluihin, AI Builder tekoälymallien hyödyntämiseksi sovelluksissa tai työnkuluissa, Dataverse-tietokanta, Power FX -ohjelmointikieli ja hallitut ympäristöt (*managed environments*) ratkaisujen kehittämiseksi, testaamiseksi ja ylläpitoon. (Microsoft Learn 2025a.) Seuraavaksi esittelen opinnäytetyöni kannalta olennaiset tuotteet ja komponentit.

3.2.2 Dataverse

Dataverse on Microsoft Azuressa toimiva tietokanta, joten se on maailmanlaajuisesti saatavilla. Se on suunniteltu tukemaan monenlaisia tietotyyppejä, ja se sisältää sekä relaatio- että ei-relaatiomallit. Dataversen tukee myös useita muitakin teknologioita kuten kuvien tallennusta, etsintätoimintoja, data lake -integraatioita sekä tiedostojen hallintaa. Ulkoisten tietolähteiden hallitsemisessa ja tiedon käsittelyssä Dataverse käyttää virtuaalisia tauluja, mitkä mahdollistavat reaaliaikaiset dataoperaatiot. (Microsoft Learn 2023.)

3.2.3 Power Apps

Power Apps -työkalulla voidaan nimensä mukaisesti rakentaa sovelluksia. Sovellus on mahdollista kytkeä lukuisiin erilaisiin tietolähteisiin erilaisten yhdistimien avulla – niin paikallisiin kuin eri palvelimilla sijaitseviin. Esimerkiksi Dataversen käyttö tietolähteenä mahdollistaa tietoperusteisten (model-driven)

sovellusten nopean kehittämisen. Muita yleisiä tietolähteitä ovat muun muassa Sharepoint, Dynamics 365, SQL Server ja Azure SQL. (Microsoft Learn 2025b.)

Sovelluksia kehitetään ja ylläpidetään Power Apps Studiolla, mikä toimii ainoastaan selainkäyttöliittymällä make.powerapps.com -portaalissa. Sovelluksen kehittämisen voi aloittaa valmiista pohjasta tai täysin ”puhtaalta pöydältä”, ja sovelluksia voi kehittää asetuksiltaan ja asetteluiltaan sopiviksi joko tabletille tai puhelimelle. Formaattia ei voi jälkikäteen muuttaa. (Microsoft Learn 2025c.)

Muita Power Appsin olennaisia komponentteja ja ominaisuuksia ovat:

- galleriat tietueiden visuaalista esittämistä varten,
- lomakkeet (forms) gallerian tietueiden muokkaamiseksi,
- syötetoiminnot (input controls) esimerkiksi tekstille, painikkeille ja kalentereille,
- älykkäät toiminnot (intelligent controls) esimerkiksi kameran käyttämiseksi tai GPS-paikannukseen,
- funktiot toimintojen väliseen kommunikointiin
- ulkoasun responsiivisuus sekä
- Copilot joko sovelluksen kehitysvaiheessa avustamassa kehitystä, tai Copilot ”sisäänrakennettuna” ominaisuutena sovelluksen käyttäjien hyödynnettäväksi. (Microsoft Learn 2025c.)

3.2.4 Power Automate

Power Automate on työkalu automatisoitujen työnkulkujen kehittämiseen, ja siitä on saatavilla kolme erilaista sovellusta eri käyttötarkoituksiin. Selainkäyttöliittymässä make.powerautomate.com voidaan kehittää sekä ylläpitää automaatioita, mobiilisovellus on tarkoitettu kevyempiin ylläpidollisiin toimiin ja Desktop-sovelluksella voidaan mallintaa erilaisia työpöytätoimintoja. (Microsoft Learn 2025d.)

Työnkulkujen avulla voidaan esimerkiksi kommunikoida eri järjestelmien välillä, opastaa käyttäjää prosessin etenemisessä tai automatisoida usein toistuvia

tehtäviä. Erilaisia työnkulkuja voidaan yhdistellä keskenään: esimerkiksi pilvipohjainen työnkulku voi käynnistää työpöytätyönkulun. Työnkulkuja on kolmenlaisia:

- liiketoimintaprosessien työnkulut, joilla helpotetaan sovelluksen käyttäjien työn etenemistä ja joiden vaatimukset ja vaiheet ovat organisaation määrittämiä
- pilvipohjaiset työnkulut (cloud flows*)*, jotka käynnistyvät tietystä laukaisimesta ja suorittavat sen jälkeen niihin määritellyt toiminnot yhdistetyissä tietolähteissä parametrien avulla
- työpöytätyönkulut eli ohjelmistorobotiikan prosessit työasemalla, joihin voidaan tallettaa useita peräkkäisiä toimintoja työpöytäsovelluksista tai selaimesta. (Microsoft Learn 2025e.)

Myös Power Automatessa tietolähteisiin kytkeydytään yhdistimien avulla. Saatavilla on yli tuhat valmista yhdistintä ja Microsoft Power Platform tukee useita yhtäaikaisia tietoyhteyksiä, joten yhteen työnkulkuun voi tuoda tietoa useasta eri järjestelmästä. (Microsoft Learn 2025f.)

3.2.5 Power BI

Power BI on työkalu tiedon analysoimiseksi ja visualisoimiseksi. Se koostuu ohjelmistoista, palveluista ja yhdistimistä, joiden avulla voidaan kytkeytyä erilaisiin tietovarastoihin sekä suodattaa ja mallintaa tietoa sen käsittelyä varten muokkaamatta alkuperäistä tietolähdettä. (Microsoft Learn 2025g.)

Tiedonkäsittelyyn Power BI:llä on kolme erilaista sovellusta: Power BI Desktop, Power BI service sekä Power BI mobiilisovellus. Ratkaisut koostuvat kolmesta elementistä: dataseteistä, raporteista ja koontinäytöistä (dashboards). Niitä hallinnoidaan työtiloissa (workspaces) – joko henkilökohtaisessa tai jaetussa. Power BI:hin kytkettyä tai tuotua tietoa kutsutaan semanttiseksi malliksi ja yhtä semanttista mallia voidaan käyttää useassa eri työtilassa. (Microsoft Learn 2025h.)

Raportit ovat visualisointeja, kuten kaavioita tai karttoja. Niitä voi luoda itse, tuoda niitä jaetuista työtiloista tai luoda automaattisesti tietolähteeseen yhdistäessä. Koontinäytöt puolestaan ovat ikään kuin työpöytänäkymiä erilaisten raporttien esittämiseksi ja yhdellä koontinäytöllä voi esittää useita raportteja. (Microsoft Learn 2025h.)

3.2.6 Copilot

Microsoft Copilot on virtuaaliavustaja, jonka generatiivinen tekoäly on käytettävissä sekä kehittäjille että tuotteen loppukäyttäjille. Copilot voi auttaa kehittäjää suunnittelemaan tai luonnostelemaan sovelluksen tai työnkulun luonnollisella kielellä annetun komennon perusteella. Loppukäyttäjä voi esimerkiksi pyytää Copilotia analysoimaan tietoa sovelluksessa tekemistään päivittäisistä työtehtävistään. (Microsoft Learn 2025i.)

Copilot on käytettävissä avustajana koko kehitystyön ajan, jolloin siltä voi esimerkiksi kysyä ohjeita tai tietoa omasta kehitteillä olevasta tuotteesta, pyytää lisäämään toimintoja ratkaisuun tai kytkeytymään uuteen tietolähteeseen. Copilot on käytettävissä sekä Power Appsin että Power Automaten kanssa. (Microsoft Learn 2025i.)

4 Opinnäytetyön toteutus

4.1 Tavoite ja menetelmälliset valinnat

Opinnäytetyön tavoitteena oli saada yleiskäsitys low-code-kehittämisestä Microsoft Power Platform -työkaluilla sekä generatiivisen tekoälyn käytön hyödyistä ja haasteista kehitysprosessin aikana.

Tavoitteeni saavuttamiseksi suunnittelin kehittäväni muutamia erilaisia low-code-ratkaisuja Power Appsilla ja Power Automatella. Lisäksi suunnittelin integroivani Power Apps -toteutuksen sekä Power BI -visualisoinnin Microsoft

Teamsiin, ja nämä kaikki ratkaisut käyttäisivät samaa, työnkuluissa ja sovelluksissa päivittyvää Dataverse-tietolähdettä.

Tarkoitukseni oli näin selvittää, kuinka generatiivista tekoälyä voidaan hyödyntää low-code-kehittämisen tehostamisessa ja millaista sisältöä sen avulla voi tuottaa. Näiden kokeilujen pohjalta tarkoitukseni oli arvioida generatiivisen tekoälyn käytön hyötyjä ja haasteita.

Halusin myös selvittää, kuinka vaativaa komentojen laatiminen on, ja kuinka paljon aikaa tekoälyn tuotosten tarkastamiseen ja varmistamiseen kuluu. Tämän vuoksi halusin kokeilla joitakin ratkaisuja ensin ilman tekoälyn avustusta, jotta voisin arvioida tekoälyn tuotosten hyödyllisyyttä.

Toteutetut ja työvaiheineen dokumentoidut ratkaisut auttaisivat myös pohtimaan tekoälyyn liittyviä eettisiä kysymyksiä. Opinnäytetyöprosessiin liittyi oleellisesti myös PL-900: Microsoft Power Platform Fundamentals –sertifikaatin suorittaminen alustavan ymmärryksen hankkimiseksi.

4.2 Tietoperustan hankkiminen ja eri vaihtoehtojen kokeilu

Vähäisen kokemukseni vuoksi toteutusvaihe alkoi jo suunnitteluvaiheen aikana tutustumalla uudestaan näihin työkaluihin. Kokeilin, kuinka Power Appsilla ja Power Automatella kehittäminen tapahtuu ilman tekoälyn avustusta, itse tarvittavat komponentit valiten, että voisin demoja tehdessäni arvioida tekoälyn tekemiä valintoja mahdollisimman kriittisesti sekä koostaa opinnäytetyön tavoitteisiin sopivan tietoperustan kirjallisuuskatsausta varten. Suoritin myös tavoittelemani sertifikaatin, josta sain hyvää ymmärrystä kaikista Power Platformin tuotteista sekä niiden välisistä yhteyksistä ja vaatimuksista. Tiesin, että minun täytyisi selvittää eri yhdistimien yhteensopivuutta esimerkiksi integroinnin onnistumiseksi ja valita sopivat ennen demojen toteutuksen aloittamista. Testasin myös aikaisempiin kokeiluihini liittyvien Power Apps -sovelluksen ja Power BI -visualisoinnin viemistä Teamsiin.

Kiinnostuin Dataverse-tietokannasta, mikä mahdollistaisi model-driven-sovelluksen tekemisen – kumpaakaan näistä en ollut ennen käyttänyt. Kokeilin laatia Copilotin avulla erilaisia tietokantatauluja Dataverseen asiakas-, tilaus- ja tuotetietojen hallinnoimiseksi, ja tutkin taulujen ominaisuuksia ja muokattavuutta.

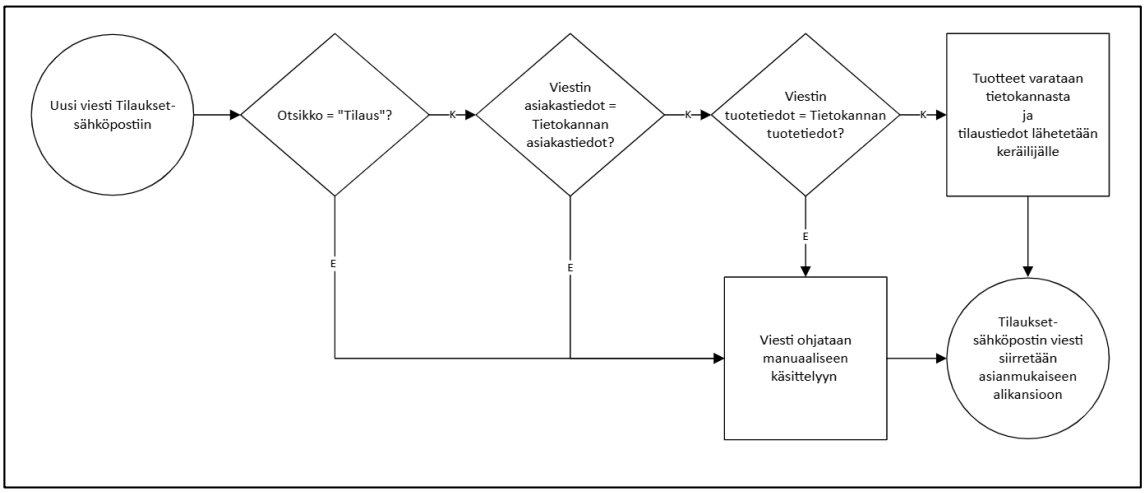
4.3 Demojen toteutus

4.3.1 Prosessikuvaukset

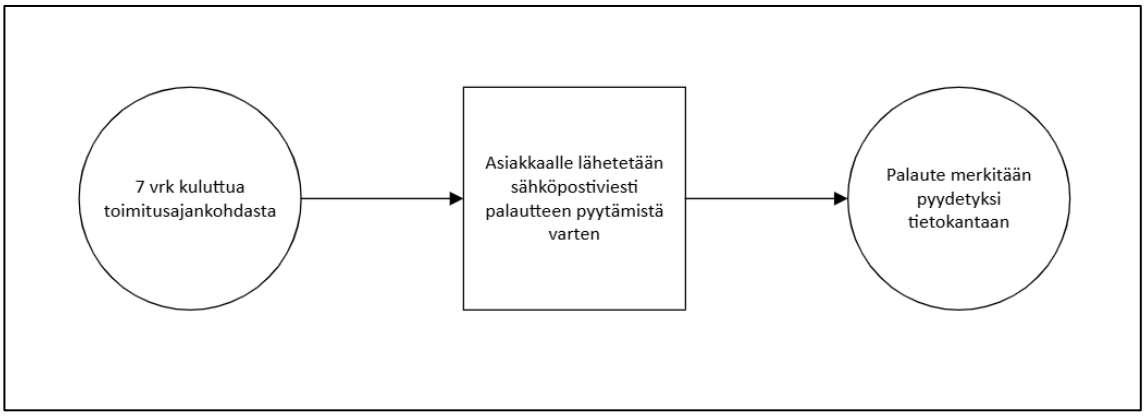
Demojen toteutusvaihe alkoi laatimalla mahdollisimman tarkat prosessikuvaukset, sillä tietokantataulujen rakenteet ja toiminnallisuudet olivat oltava valmiina ennen sovellusten ja työnkulkujen kehittämistä. Samanaikaisesti kirjasin muistiin tärkeimpiä testitapauksia, joiden avulla voisin testata ratkaisujen toimivuutta.

Eri teknologioiden testaamiseksi päädyin laatimaan kolme erilaista demoa:

1. Dataverse-tietokannan hallinnointi Power Appsin model-driven-sovelluksella, jonka taulut esittelen myöhemmin kuvassa 4,
2. Pilvipohjainen työnkulku Power Automatella tilaussähköpostin käsittelyyn (kuva 1), sekä
3. Ajastettu työnkulku Power Automatella asiakaspalautteen pyytämiseksi (kuva 2).



Kuva 1. Ylätason prosessikaavio tilauksia käsittelevästä työnkulusta.



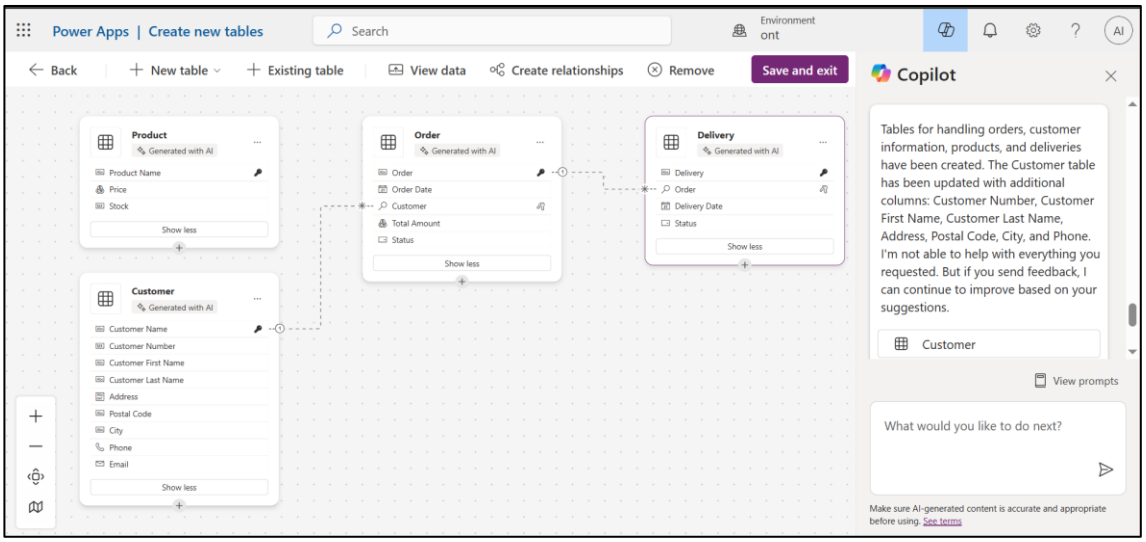
Kuva 2. Ylätason prosessikaavio asiakaspalautteen pyytämisen työnkulusta.

4.3.2 Dataverse-taulujen laatiminen

Määrittelyjen jälkeen oli vuorossa tietokantataulujen suunnittelu. Keräsin jokaisen demon prosessikaaviosta ylös asiat, jotka tietokannassa ainakin tulisi olla, ja aloitin syötteiden laatimisen Copilotille. Ensimmäinen syötteeni sisälsi pienen kirjoitusvirheen, joten kumosin ensimmäisen tuotoksen ja syötin korjatun komennon, jolloin tuotos muuttui merkittävästi. Kumosin tuotoksen vielä kertaalleen ja syötin täsmälleen saman komennon, jolloin tuotos muuttui taas edelliskertaan verrattuna erilaiseksi. Näiden epäjohdonmukaisuuksien takia päätin, etten hio syötteitä tässä vaiheessa liikaa, ja päädyin aloittamaan taulujen laadinnan seuraavalla syötteellä:

> Create tables for handling orders, customer information, products and deliveries. “Customer” should include columns “customer number”, “customer first name”, “customer last name”, “address”, “postal code”, “city”, “phone”, “email”. “Products” should include columns “product number”, "product name”, “price”, “amount in stock”. “Orders” should include columns "order number", "customer number”, “product number”, “amount ordered”, “amount reserved”, “sum”, “order date”, “delivery time”, “delivery status”.

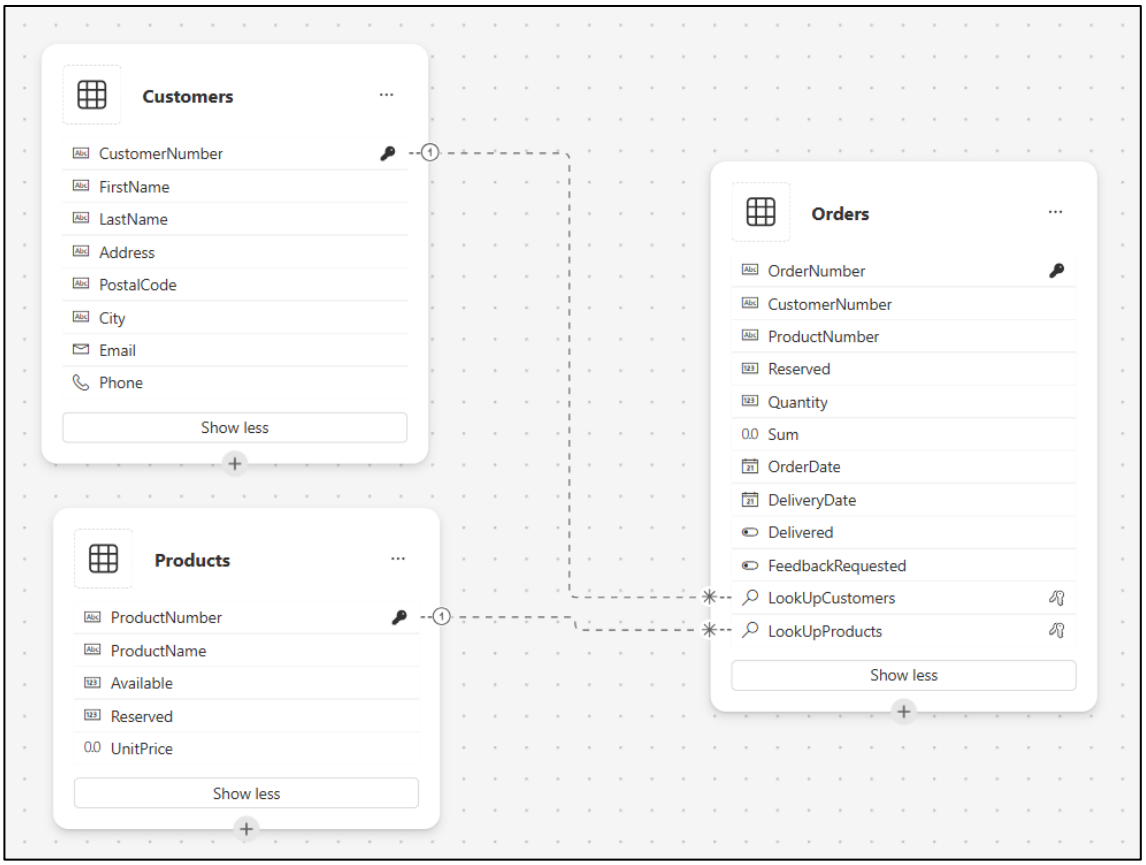
Kuvasta 3 näkyy, että Copilot laati neljä taulua annetun komennon perusteella. Vastauksessaan Copilot ilmoitti, ettei se onnistunut täysin tehtävässään.



Kuva 3. Copilotin laatimat taulut sekä ilmoitus tuotetusta sisällöstä.

Copilotin tuotoksien yhteydessä näkyi ilmoitus, että ne on luotu tekoälyllä. Lisäksi komentoikkunan alla oli muistutus varmistaa tuotoksen oikeellisuus.

Copilot oli laatinut taulujen välille riippuvuuksia, jotka eivät vaikuttaneet kovin tarkoituksenmukaisilta. Kokeilin aluksi muokata niitä itse, mutta minua jäi epäilyttämään mahdolliset piilossa olevat toiminnallisuudet ja rakenteet, koska en tuntenut Dataverseä entuudestaan. Copilotin käytön osoittautuessa liian haasteelliseksi tietokanta-asioiden avustamisessa, päätin laatia taulut itse sarake kerrallaan vaatimusmäärittelyn mukaisesti ja tietotyypit huolella valiten (kuva 4).

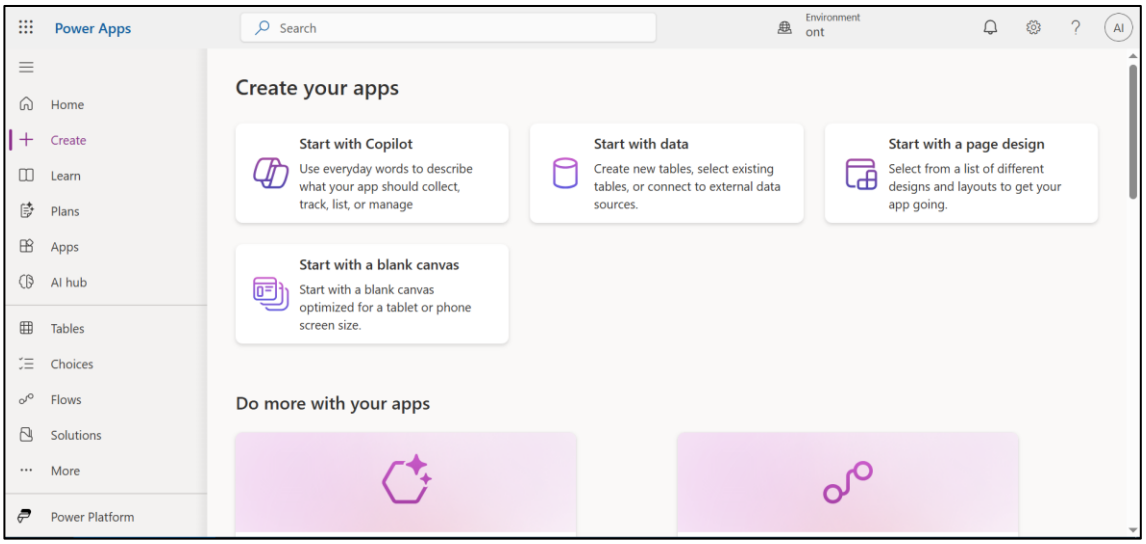


Kuva 4. Laatimani taulut sarakkeineen ja riippuvuuksineen.

Sarakkeiden nimien edessä näkyy sarakkeiden tietotyypit. Tässä vaiheessa taulut eivät vielä sisältäneet tietueita.

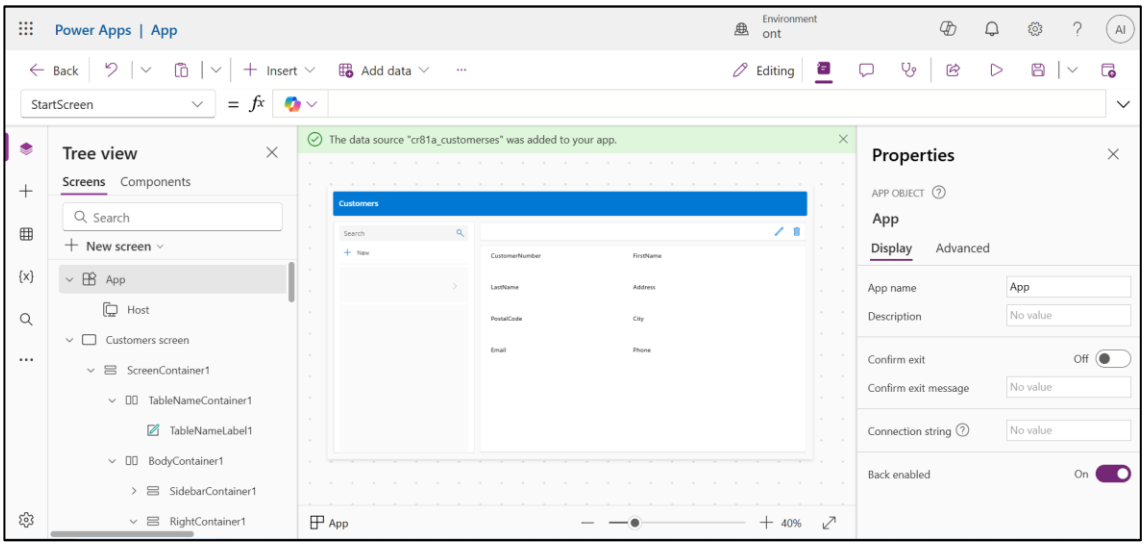
4.3.3 Power Apps

Power Appsissa kehittäjä voi valita eri vaihtoehdoista, kuinka aloittaa sovelluksen rakentaminen (kuva 5). Sovelluskehityksen voi aloittaa Copilotin avulla, tietolähteisiin kytkeytymällä tai valmista mallipohjaa käyttämällä. Sovelluksen luonnostelun voi aloittaa myös täysin tyhjästä näytöstä.



Kuva 5. Erilaisia vaihtoehtoja Power Apps -kehittämisen aloittamiseksi.

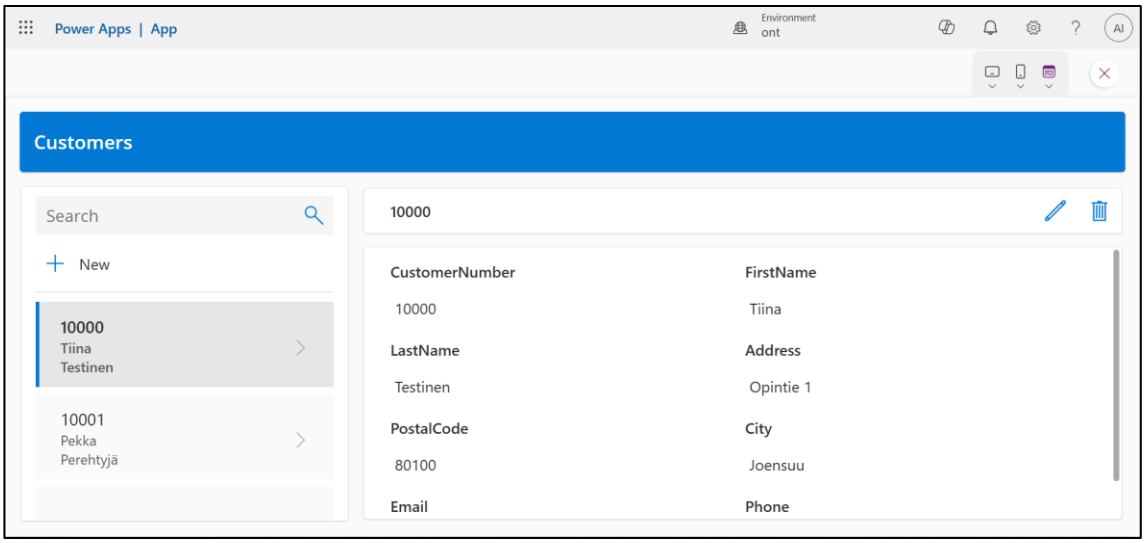
Taulujen luomisen jälkeen loin model-driven-sovelluksen ”Customers”-taulun tietueiden laatimista varten, jolloin perustoiminnallisuus luodaan automaattisesti valitun tietokantataulun rakenteiden mukaiseksi. Kuvan 6 vasemman reunan puunäkymästä voi valita tietyn elementin muokattavaksi. ”Properties”-paneelista voi muokata valitun elementin toiminnallisuuksia tai muita teknisiä asetuksia.



Kuva 6. Model-driven-sovellus muokkausnäkymässä.

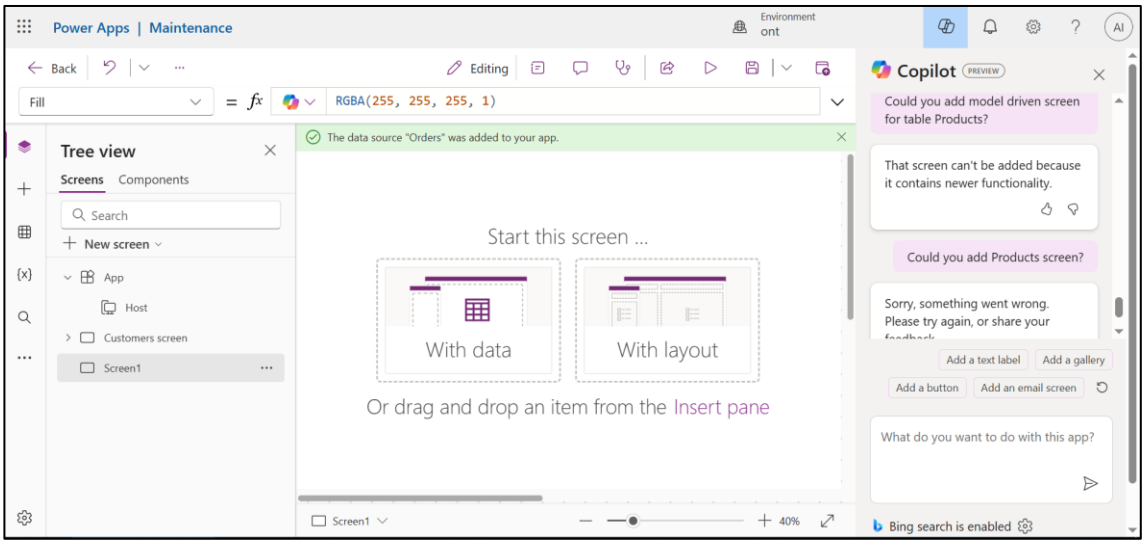
Sovellusta testatessa tauluun pystyi lisäämään uusia tietueita tai muokkaamaan olemassa olevia visuaalisen käyttöliittymän kautta. Kuvassa 7 näkyy kaksi

lisäämääni tietuetta. Vasemman yläkulman ”+”-painikkeesta voi lisätä uuden tietueen, joka listataan vasemman sivupalkin näkymään. Valittua tietuetta voi myös muokata käyttöliittymän kautta. Testinäkymän yläpuolella on mahdollisuus valita esikatselutila joko tabletille, puhelimelle tai selaimelle.



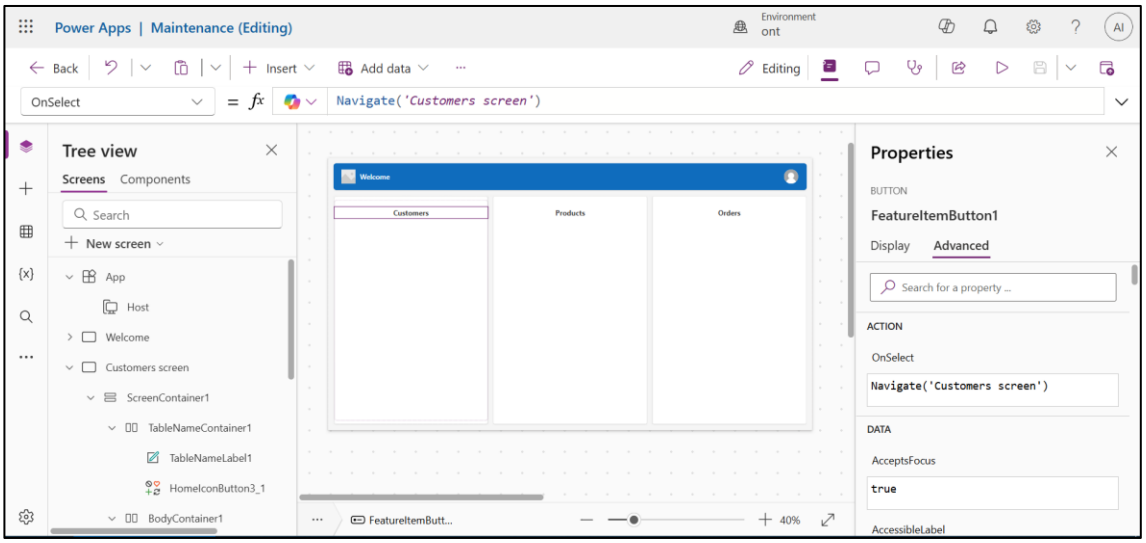
Kuva 7. ”Customers”-taulusta luotu model-driven-sovellus testaustilassa.

En muokannut sovelluksen oletusmuotoista ulkoasua tai toiminnallisuuksia, sillä sovelluksen käyttötarkoituksena oli vain tuottaa ja ylläpitää demojen tarvitsemaa tietoa sekä tutkia eri elementteihin määriteltyjä toimintoja. Lisäsin sovellukseen myös ”Products”- ja ”Orders”- taulut tietolähteiksi, sekä pyysin Copilotia laatimaan sovellukseen näytön ”Products”-taululle. Copilot ei onnistunut tehtävässään, mutta lisäsi kuitenkin oletusarvoisen näytön sovellukseen (kuva 8):



Kuva 8. Copilotin epäonnistunut tuotos ”Screen1” sekä ilmoitus tuotetusta sisällöstä.

Päätin lisätä uuden näytön tiedot itse. Muokkasin ”Screen1”-näytön nimen ja valitsin yllä olevassa kuvassa näkyvältä työpöydältä vaihtoehdon ”Start this screen with data”, jonka jälkeen sovellus pyysi minua valitsemaan halutun taulun tietokannasta. Tämän jälkeen ”Customers”-näytön kaltainen näyttö valmistui, mutta yllättäen sen toiminnallisuudet katosivat, eikä sitä voinut enää muokata. Avasin sovelluksen uudelleen muokkaustilaan, jolloin näyttö oli kokonaan kadonnut. Pohdin, oliko tämä yllättävä virhe aiheutunut Copilotin epäonnistumisesta näytön luomisessa, joten en pyytänyt enää Copilotia luomaan mitään sisältöä Power Appsissa, vaan loin näytöt manuaalisesti ja yhdistin ne tarkoituksenmukaisiin tauluihin. Lopuksi lisäsin vielä aloitusnäytön ”Welcome”, josta voi navigoida johonkin kolmesta aiemmin mainitusta taulusta, ja määritin sovelluksen käynnistymään kyseiseltä näytöltä (kuva 9).

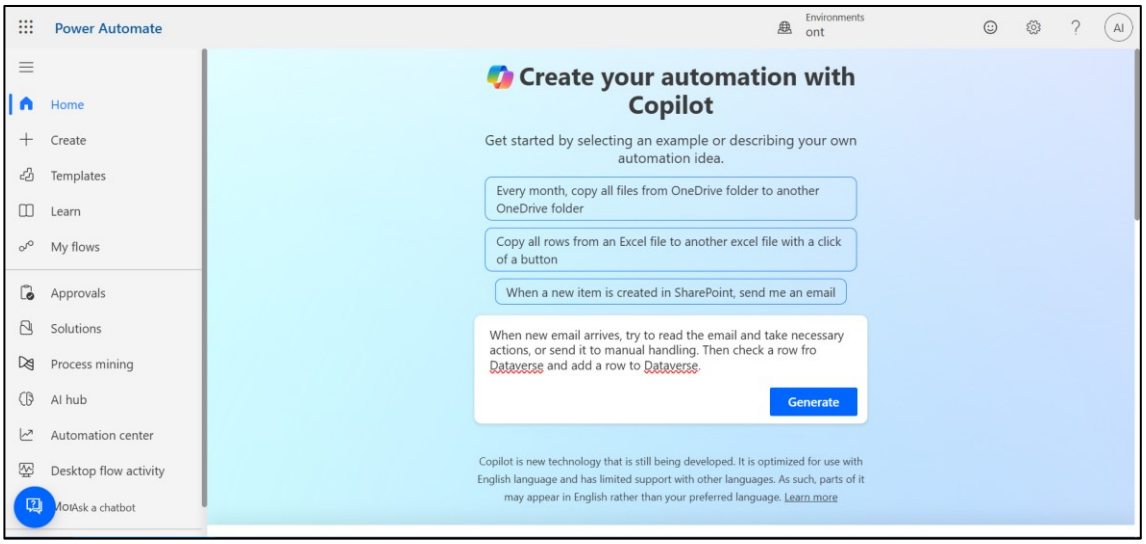


Kuva 9. Sovelluksen aloitusnäyttö muokkaustilassa.

Yllä olevan kuvan oikeassa reunassa on ”Properties”-paneeli, jossa elementtien toimintoja voidaan määritellä. Kuvan tilanteessa on määritelty, että ”Customers”-painikkeen valitseminen avaa sovelluksen ”Customers screen” -näkymässä.

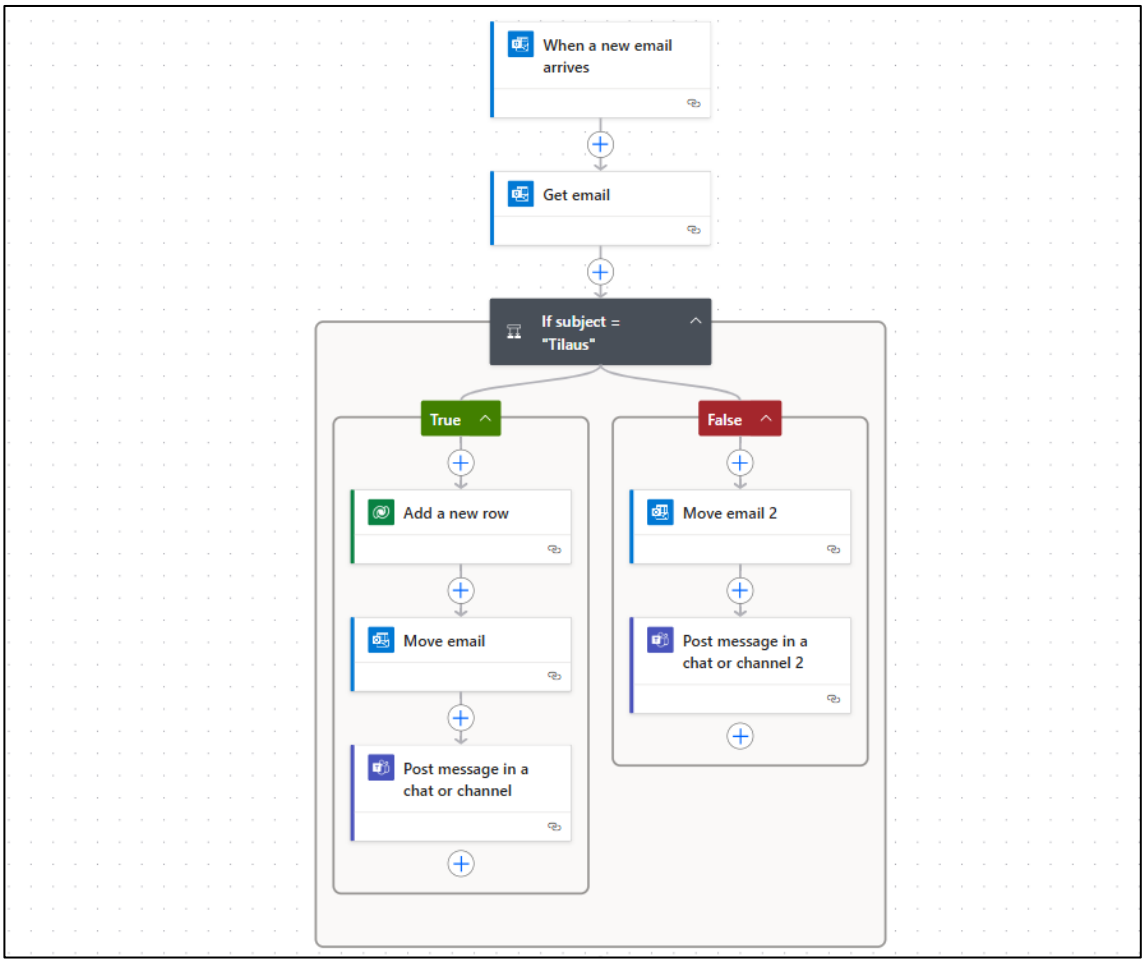
4.3.4 Power Automate

Kuten Power Appsinkin, myös Power Automate -työnkulku on mahdollista aloittaa Copilotin avustuksella. Kehittäjän on myös mahdollista valita erilaisia mallipohjia tai aloittaa kehitys tietyn käynnistimen perusteella. Aloitin kuvailemalla Copilotille suunnitellun työnkulun päävaiheita, ja monen kokeilun jälkeen vaikutti siltä, että Power Automaten puolella Copilot toimii huonommin kuin Power Appsin. Myös syötteen pituus etusivulla oli rajattu vain 250 merkkiin, kun se Power Appsin puolella oli 5000. Lopulta päädyin seuraavanlaiseen syötteeseen (kuva 10):



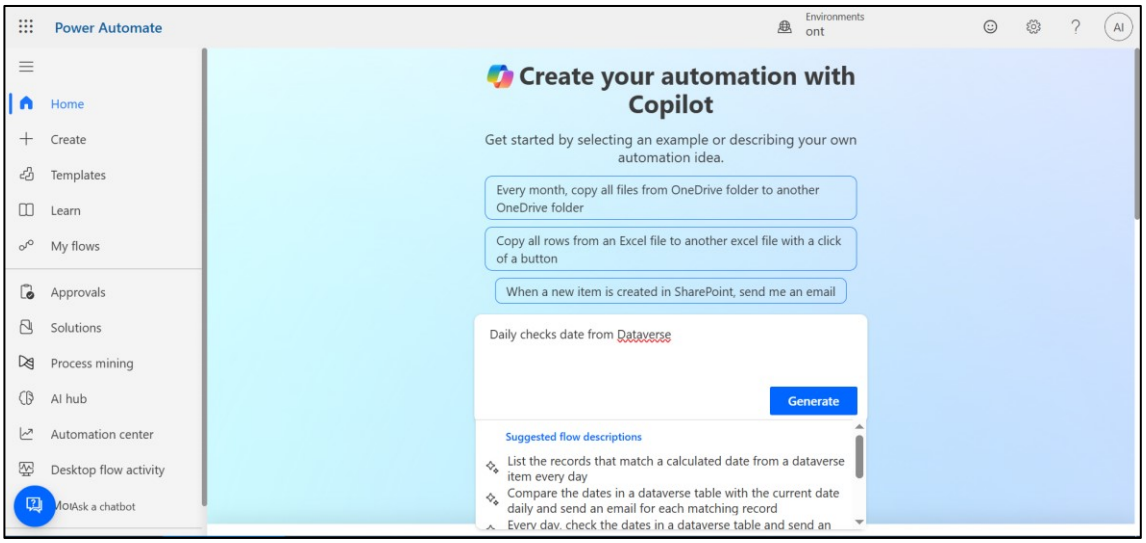
Kuva 10. Copilotille syöttämäni komento työnkulun laatimiseksi.

Tällaisella syötteellä sain Copilotin tuomaan työnkulkuun mahdollisimman monta erilaista toimintoa. Tarkoituksenani oli lisätä mahdollisesti puuttuvat toiminnot itse. Tämän jälkeen valitsin Copilotin ehdotuksista kattavimman, ja tarkastin Dataversen ja Outlookin yhdistimien tarvitsemat yhteydet. Copilotin työnkulun luomisen jälkeen käyttö ei olisi ollut kehitysympäristön vanhassa versiossa mahdollista, ja koska uusi versio oli mielestäni monimutkaisempi käytettävyydeltään, päätin jatkaa ilman tekoälyapuria ja lisäsin työnkulkuun vielä muita tarvittavia toimintoja (kuva 11).



Kuva 11. Copilotin osittaisella avustuksella laadittu Tilaus-työnkulku.

Viimeisenä demona laadin vielä ajastetun työnkulun asiakaspalautteen pyytämiseksi. Syötin tekstikenttään muutaman sanan, jonka perusteella Copilot listasi muutamia valmiita mallipohjia (kuva 12).

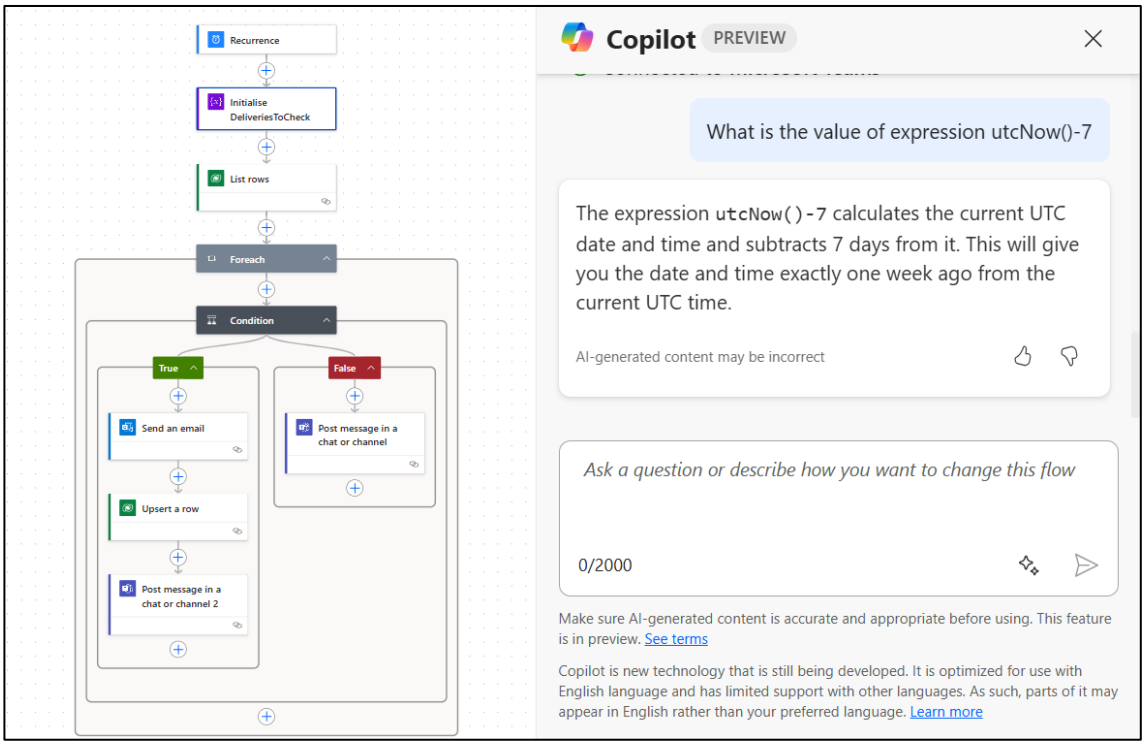


Kuva 12. Copilotin ehdottamia vaihtoehtoja.

Valitsin yllä olevassa kuvassa näkyvistä ehdotuksista vaihtoehdon ”Compare the dates in a dataverse table with the current date daily and send an email for each matching record”. “Generate”-napin painaminen näytti tekoälyn esittämät vaihtoehdot työnkulun pohjaksi.

Valitsin tarpeisiini paremmin sopivan vaihtoehdon, tarkastin jälleen yhdistimien vaatimat yhteydet ja annoin tekoälyn luoda työnkulun. Tämän jälkeen minun tuli määritellä kaikkiin aktiviteetteihin tarvittavat parametrit, kuten taulun nimi, sähköpostitiedot sekä ehtolause. Käynnistimen pakolliset parametrit – käynnistymisen taajuus ja lukumäärä – olivat valmiiksi määriteltyinä tekoälyn toimesta, mutta lisäsin vielä valinnaisina parametreinä aikavyöhykkeen ja käynnistymisajan.

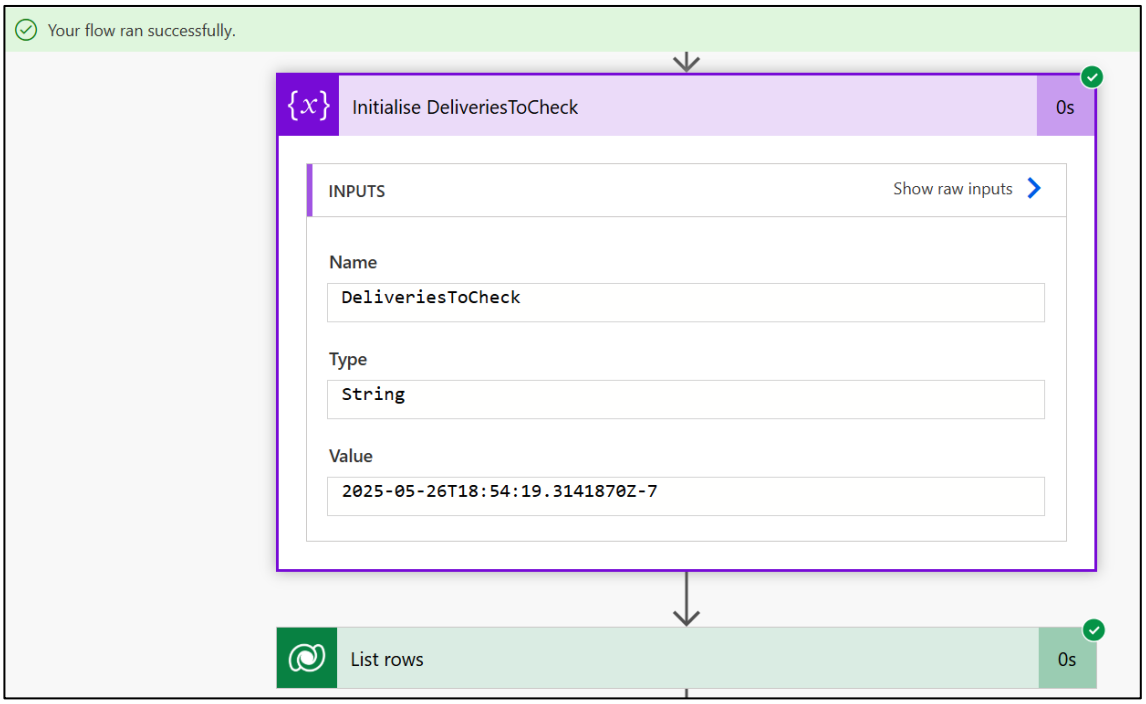
Määrittelin myös muuttujan, jota tarvitaan etsimään viikko sitten toimitettuja tilauksia. Päätin tarkistaa käyttämäni kaavan oikeellisuuden, joten vaihdoin kehitysnäkymän uuteen versioon ja kysyin asiaa Copilotilta (kuva 13). Copilotin vastauksen perusteella kirjoittamani kaava vaikutti asianmukaiselta.



Kuva 13. Työnkulku ja Copilotin komentopaneeli.

Kuvan 13 oikeassa reunassa näkyy Copilotin vastaus kysymykseeni. Vasemmassa reunassa näkyy työnkulku, joka käynnistyy tiettyyn aikaan, ja päättyy viestin lähettämiseen Teams-kanavalle. Copilot luonnosteli työnkulun lukuunottamatta True-haaran kahta viimeisintä aktiviteettiä ja False-haaran sisältöä, jotka lisäsin työnkulkuun itse.

Muokkasin tietokantojen dataa testausta varten sopivaksi, mutta testaus päätyi odotusten vastaisesti Else-haaraan. Tarkastelin työnkulun raporttia (kuva 14), ja yllätykseksenі muuttujan määrittelyssä apuna käyttämäni Copilotin vastaus ei ollutkaan ollut pätevä.

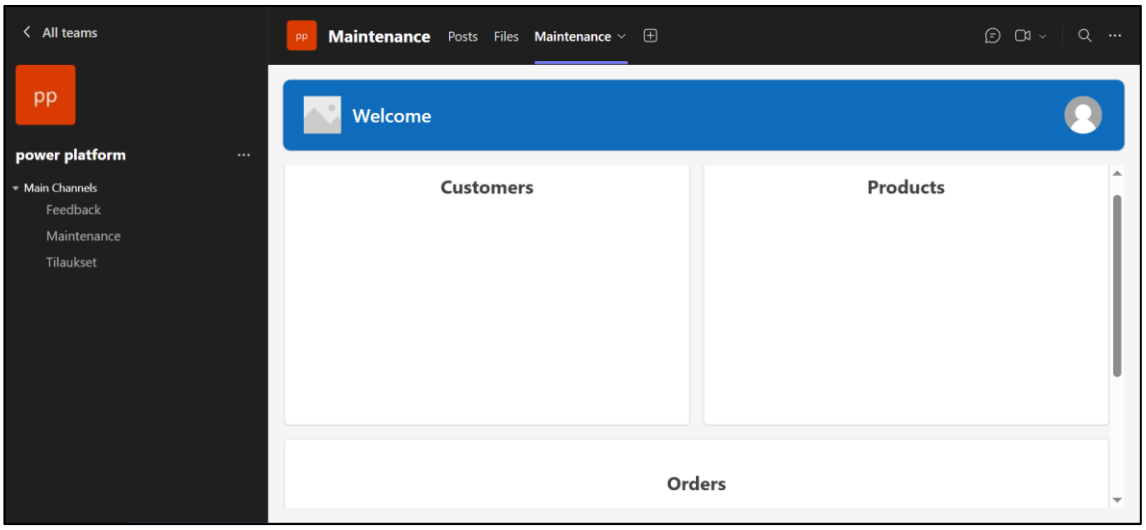


Kuva 14. Yksi työnkulun vaiheista raporttinäkymässä.

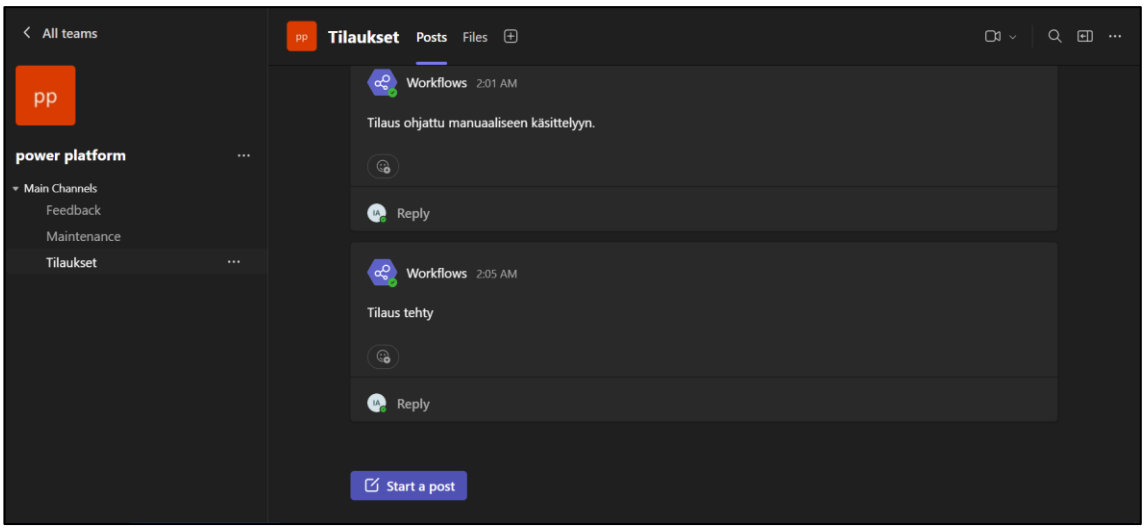
String-tietotyyppisestä eli tekstimuotoisesta päivämääräarvosta ei voi vähentää lukua 7. Päivämäärät ovat kuitenkin Dataverse-tietokannassa aina tekstimuotoisena – tätä Copilot ei ottanut huomioon vastauksessaan. En muuttanut työnkulkuja enää tämän havainnon jälkeen.

4.4 Teamsiin integrointi

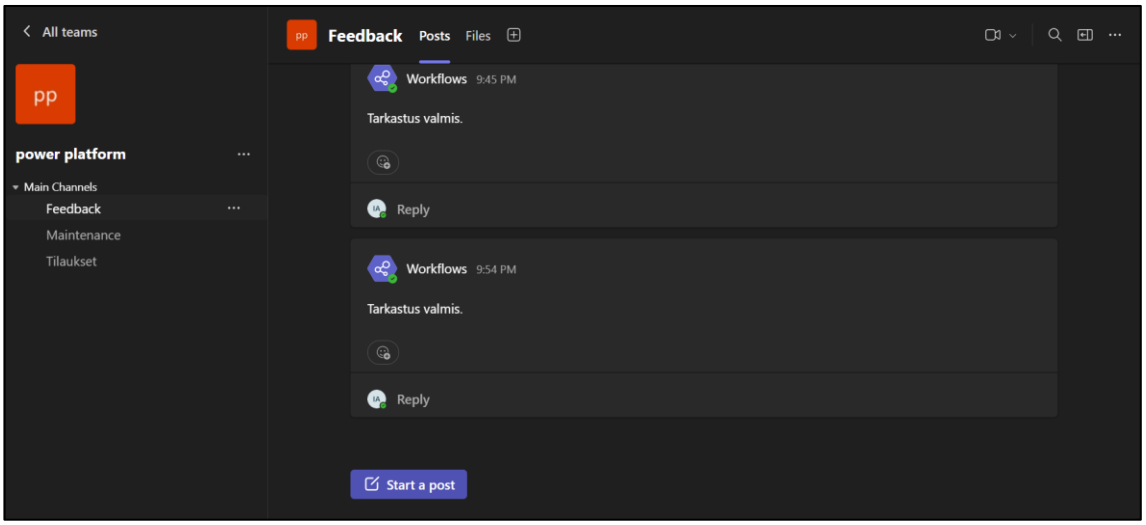
Integroin tämän opinnäytetyön Power Apps -toteutuksen (kuva 15) sekä työnkuluista lähetettävät ilmoitukset (kuvat 16 ja 17) Microsoft Teamsiin. Yhdistimien avulla tallennetut sovellukset ovat avattavissa halutulle kanavalle, jossa niitä voi käyttää määritettyjen käyttöoikeuksien mukaisesti. Aloitin sovelluksen viemisen lisäämällä kanavalle uuden välilehden, jonne pystyin valitsemaan minkä tahansa sovelluksen, johon käyttöoikeuteni riittivät. Välilehden nimi muuttui automaattisesti sovelluksen nimeksi.



Kuva 15. Microsoft Teamsiin integroitu Power Apps -sovellus.



Kuva 16. Tilaukset-kanavalle lähetettyjä ilmoituksia tilausten käsittelyn työnkulun tuloksista.



Kuva 17. Feedback-kanavalle lähetettyjä ilmoituksia palautteen pyytämisen työnkulun tuloksista.

Minun oli vielä julkaistava sovellus, että sain viimeisimmän version näkymään Teamsissa oikein. Teams muistutti myös säätämään sovelluksen käyttöoikeuksia, että se olisi saatavilla halutulle kohderyhmälle. Power BI -visualisointeja en ehtinyt tekemään.

4.5 Testaaminen

Minulla ei ollut tarkoitus testata tuotoksia kattavasti, sillä demojen tarkoitus oli tutustua Power Platformin eri tuotteisiin ja testata generatiivisen tekoälyn käyttöä. Laadin kuitenkin joitakin testitapauksia demojen suunnitteluvaiheen ohjenuoraksi.

Power Apps -demon tarkoituksena oli toimia pääasiassa Power Platformin toiminnallisuuksien nopean testaamisen mahdollistajana, sillä sovelluksen avulla tietokannan hallinnointi on nopeaa ja visuaalista. Generatiivisen tekoälyn kokeilujen viedessä merkittävästi aikaa, en ehtinyt testaamaan eri tapauksia ja siksi todellisia testituloksia ei ole dokumentoitu.

5 Tulokset

Opinnäytetyön tuotoksena syntyi kolme demoa, joiden kehittämisessä sain kokeilla sertifikaatissa opiskeltuja tietoja. Tietoperustan keräämisen aikana ymmärrykseni käytettävissä olevista vaihtoehdoista ja niiden yhdistämismahdollisuuksista kasvoi merkittävästi. Power Apps -sovelluksen lisääminen Teamsiin nosti mielestäni ratkaisun arvoa tuomalla sen helposti saataville valituille käyttäjille. Power Platform on monipuolinen alusta, ja olisin halunnut yhdistellä mahdollisimman monia eri komponentteja keskenään, mutta tuotteiden lukumäärä, generatiivisen tekoälyn käytön haasteet ja Dataversen mukaan ottaminen vastoin alkuperäistä suunnitelmaani ja aikataulutustani rajoittivat toteutuksen lopullista laajuutta ja yhteensopivuutta.

Kaikki tuotokset käyttivät Dataverse-tietokantaan laatimiani tauluja, ja kaikkien tuotosten käytössä hyödynsin generatiivista tekoälyä mahdollisimman erilaisilla tavoilla. Erityisesti Power Appsilla luonnosten laatiminen Copilotin avustuksella oli hyödyllistä low-code-kehittämisen opettelun kannalta, sillä siten pääsin nopeasti tutkimaan eri elementtien toiminnallisuuksia ja logiikkaa. Näitä muokkaamalla pystyin nopeasti näkemään muutoksen vaikutuksen kokonaisuuteen ja opin tiettyjä säännönmukaisuuksia ominaisuuksien määrittelyssä.

Mikäli taulut olivat hyvin valmisteltuja, testitietueiden tuottaminen Copilotin avulla oli nopeaa ja helppoa. Käytin Copilotia sekä luonnosteluun että antamaan neuvoja myöhemmissäkin kehityksen vaiheissa. Copilot ei kuitenkaan osannut kaikissa tilanteissa kohdistaa vastaustaan erityisesti Power Platformin tuotteisiin, vaan vastaukset saattoivat olla muuhun teknologiaan paremmin sopivia tai yleispäteviä. Copilot ei myöskään keskustelussa kysynyt tarkentavia kysymyksiä, vaan ilmoitti joko onnistuneensa tai epäonnistuneensa, ja pyysi lähettämään palautetta. Luonnosteluvaiheessa Copilot antoi muutamia vaihtoehtoja valittaviksi, mutta varsinaista vuoropuhelua sekään ei ollut.

Mitä pidempiä komentoja annoin, sitä todennäköisemmin Copilot epäonnistui tuotoksessaan. Etenkin prosessin alussa, eli taulujen luontivaiheessa, oli

haastavaa laatia sellaiset komennot, että ratkaisun kaikki osat tulisivat mahdollisimman suurella todennäköisyydellä toimimaan joko yhdessä tai erikseen, koska rakensin kokonaisuutta kuitenkin vain pieni pala kerrallaan. Totesin, että Copilotin käyttö yhdistettynä vähäiseen aiempaan tietämykseen esimerkiksi Dataversesta oli haastavaa ja mahdollisti todennäköisesti tahattomia, mutta ei-toivottuja tuloksia joko prosessien logiikkaan tai tiedon korruptoitumiseen.

Copilotin viimeisin tuotos oli aina mahdollista kumota. Power Platform myös ilmoitti asianmukaisesti ja selkeästi, jos tuotos oli tekoälyn laatima. Käyttäjää myös muistutettiin varmistamaan tekoälyn tuotoksen oikeellisuus. Copilot ei kuitenkaan ilmoittanut kehitysympäristössä näkyvästi, minkälaista tietoa tekoälylle ei kannata syöttää, mikä altistaa teknologiaa tuntemattoman kehittäjän syöttämään järjestelmään sensitiivistä tietoa. Tekoälylle tarjoamieni syötteiden sisältämiä tietoturvariskejä minun ei itse tarvinnut arvioida, sillä loin käytettävät tietolähteet kuvitteellisella datalla.

Toteutuksessa minut yllätti generatiivisen tekoälyn tuotosten tarkkailuun kuluneen ajan todellinen määrä. Alkuperäinenkin oletukseni oli, että komentojen muotoilu tulisi viemään paljon aikaa, mutta todellinen käytetty aika viivästytti jo valmiiksi tiukkaa aikataulua. Toinen yllättävä seikka oli se, että Power Automaten puolella Copilot vaikutti toimivan huonommin kuin Power Appsin. Syötteiden sallittu enimmäiskirjainmääräkin oli erilainen, eikä Copilot mielestäni myöskään ymmärtänyt Power Automaten kohdalla usean lauseen syötteitä kovin hyvin. Lisäksi huomasin harjoitellessani tietokantataulujen laatimista Copilotin avulla, että joillakin kerroilla Copilot saattoi tauluja luodessaan lisätä niihin valmiiksi myös tietueita ilman erillistä pyyntöä, mutta ei kuitenkaan aina.

6 Pohdinta

6.1 Aiempi osaaminen ja odotukset

Aikaisempi kokemukseni Power Platformista oli hyvin vähäistä ja yli kahden vuoden takaa. Tuolloin harjoittelin Power Appsilla hyvin yksinkertaisen sovelluksen tekemistä ja tietolähteenä käytin Excel-taulukkoa. En ollut miettinyt ympäristöjä, siirrettävyyttä tai yhdisteltävyyttä. Olin myös tehnyt hyvin yksinkertaisia ja lyhyitä työnkulkuja Power Automatella ohjelmistorobotiikan opintoihin liittyen.

Generatiivista tekoälyä olin kokeillut ainoastaan yhden kerran. En koskaan ole ollut innostunut tekoälystä, ja olen suhtautunut kielteisesti etenkin generatiivisen tekoälyn käytön nopeaan yleistymiseen ilman kokonaisvaltaista ymmärrystä sen mahdollisista vaikutuksista. Olen ollut sitä mieltä, että generatiivisen tekoälyn käyttäminen saattaa heikentää yksilön ongelmanratkaisutaitoja ja jopa vääristää oppimista, sillä tuotoksen oikeellisuudesta ei voi olla varma. Odotin kuitenkin, että generatiivinen tekoäly osaisi yhdistellä Power Platformin komponentteja toisiinsa mahdollistaen usean ratkaisun integroimisen yhteen kokonaisuuteen.

6.2 Työn onnistumisen arviointi

Aiempaan osaamiseeni verraten opinnäytetyö laajensi kokonaisvaltaista ymmärrystäni etenkin Power Platformin tuotteista, mutta myös yleisesti tekoälystä. Totesin kuitenkin, että ennakko-odotukseni laajojen kokonaisuuksien yhteensovittamisesta generatiivisen tekoälyn avulla olivat liian kunnianhimoiset. Generatiivisen tekoälyn käyttö ei ollut oikotie onneen, vaan pikemminkin enemmän aikaa vievä ratkaisu.

Prosessilla oli tiukka aikataulu, ja olin valinnut suunnitelmaani useita erilaisia teknologioita kokeiltavaksi, joista Power BI -visualisointi jäi kokonaan toteuttamatta. Sen sijaan keskityin kokonaisuutta ajatellen liian pitkäksi aikaa Dataversen tutkimiseen, mikä vei aikaa muilta työvaiheilta. Olisin halunnut perehtyä

enemmän Dataversen ominaisuuksiin ja selvittää sen soveltuvuutta erilaisten operaatioiden suorittamiseen.

Jos aloittaisin suunnitelmavaiheen uudelleen, valitsisin erilaisia komponentteja lopulliseen ratkaisuuni. Kokonaisvaltaisen ymmärrys kertyy vain harjoituksen kautta, mutta ajankäytöllisesti uudelleenorientoituminen ei ollut mahdollista. Olisin halunnut valita Power Pages -teknologian tilausten lähettämisen työnkulkuun, vaikka minulla ei ole siitä lainkaan aiempaa kokemusta. Power Pages -toiminnon liittäminen Tilaus-työnkulun käynnistimeksi olisi todennäköisesti ollut low-code-toiminnallisuuksien opettelun kannalta hyödyllisempää kuin sähköpostiviestistä käynnistyvä nykytoteutukseni, sillä sen avulla olisin voinut harjoitella kattavammin eri teknologioiden yhteensovittamista.

6.3 Lopuksi

Low-code-kehittäminen mahdollistaa nopeasti melko yksinkertaisten kokonaisuuksien rakentamisen. Valitun teknologian ominaisuuksista on kuitenkin oltava tarpeeksi tietoa ennen vaatimusmäärittelyn tekemistä, aivan kuten perinteisenkin ohjelmistokehittämisen ollessa kyseessä.

Generatiivisen tekoälyn käytössä on sekä haasteita että hyötyjä. Mikäli käytettävät teknologiat eivät ole kehittäjälle entuudestaan tuttuja, tuotosten oikeellisuuden varmistaminen voi olla hyvinkin haastavaa tekoälyn tuottaessa välillä varsin odottamattomia vastauksia.

Generatiivisen tekoälyn avulla on kuitenkin mahdollista kokeilla nopeita ratkaisuja esimerkiksi low-code-arkkitehtuurin opiskeluun. Se on hyvä työkalu myös yksinkertaisiin, kehitystä nopeuttaviin toimenpiteisiin, kuten ulkoasun yhtenäistämiseen, jolloin tulokset ovat helposti varmistettavissa. On kuitenkin oltava tarkkana, millaista tietoa avustajalle syöttää, sillä kerran syötetty tieto voi olla olemassa koko tekoälymallin elinkaaren ajan. En ehkä itse myöskään laskisi tekoälyavusteista sovellusta henkilökohtaisesti käyttämiini tietojärjestelmiin. Mielestäni on hieman ongelmallista, että low-code-alustan käyttö ei välttämättä vaadi ohjelmointitaitoja, mutta generatiivisen tekoälyn käyttö on mahdollista –

onko niin sanottu kansalaiskehittäjä tarpeeksi valveutunut tekoälynkään toimintaperiaatteista, saati sen sisältämistä tietoturvariskeistä?

Tekoälyn kehittyminen on nopeaa, mutta eettisten ongelmien määrä tuskin poistuu samalla vauhdilla. Millaista dataa tekoälylle saa tarjota ratkaisua kehittäessä? Tekoälyn tuottamaan koodiin liittyy myös tekijänoikeuskysymyksiä. Kuka datan omistaa komennon syöttämisen jälkeen? Nämä kysymykset tulevat ohjaamaan tekoälyn käyttöäni tulevaisuudessakin.

Lähteet

Ailisto, H., Heikkilä, E., Helaakoski, H., Neuvonen, A. & Seppälä, T. 2018. Tekoälyn kokonaiskuva ja osaamiskartoitus. Valtioneuvoston kanslia.

Alanko, K. & Palo, E. 2024. Ratkaiseeko tekoälyasetus generatiivisen tekoälyn ympäristöhaasteet? Sitra: Puheenvuoro.15.8.2024. Blogi. https://www.sitra.fi/blogit/ratkaiseeko-tekoalyasetus-generatiivisen-tekoalyn-ymparistohaasteet/. 26.4.2025.

Barberá, I. 2025. AI Privacy Risks & Mitigations – Large Language Models (LLMs). European Data Protection Board. https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf. 11.5.2025.

Dufva, M. 2024. Megatrendit 2024. Sitra: Puheenvuoro. 4.1.2024. https://www.sitra.fi/blogit/megatrendit-2024/. Blogi. 30.5.2025.

Euroopan parlamentti. 2023. Mitä tekoäly on ja mihin sitä käytetään? https://www.europarl.europa.eu/topics/fi/article/20200827STO85804/mita-tekoaly-on-ja-mihin-sita-kaytetaan. 1.5.2025.

Euroopan parlamentti. 2025a. Tekoäly: mahdollisuuksia ja uhkia. https://www.europarl.europa.eu/topics/fi/article/20200918STO87404/tekoaly-mahdollisuuksia-ja-uhkia. 7.5.2025.

Euroopan parlamentti. 2025b. EU:n tekoälysäädös on ensimmäinen laatuaan. https://www.europarl.europa.eu/topics/fi/article/20230601STO93804/eu-n-tekoalysaados-on-ensimmainen-laatuaan. 30.5.2025.

IBM. 2023a. What are large language models (LLMs)? https://www.ibm.com/think/topics/large-language-models. 9.5.2025.

IBM. 2023b. What are AI hallucinations? https://www.ibm.com/think/topics/ai-hallucinations. 26.4.2025.

Ite wiki. 2025. Low code -ohjelmistokehitys. https://www.itewiki.fi/opas/low-code-ohjelmistokehitys/. 12.4.2025.

Microsoft Learn 2023. Why choose Microsoft Dataverse? https://learn.microsoft.com/en-us/power-apps/maker/data-platform/why-dataverse-overview. 16.5.2025.

Microsoft Learn. 2025a. Explore Microsoft Power Platform. https://learn.microsoft.com/en-us/training/modules/introduction-power-platform/2-explore-microsoft-power-platform. 12.4.2025.

Microsoft Learn. 2025b. Describe Power Apps. https://learn.microsoft.com/en-us/training/modules/introduction-power-apps/2-describe-power-apps. 22.4.2025.

Microsoft Learn. 2025c. Build a basic canvas app. https://learn.microsoft.com/en-us/training/modules/introduction-power-apps/6-build-basic-canvas-app. 22.4.2025.

Microsoft Learn. 2025d. Explore the different Power Automate apps. https://learn.microsoft.com/en-us/training/modules/introduction-power-automate/3-explore-different-power-automate-apps. 9.5.2025.

Microsoft Learn. 2025e. Describe the capabilities of Power Automate. https://learn.microsoft.com/en-us/training/modules/introduction-power-automate/2-describe-capabilities-power-automate. 9.5.2025.

Microsoft Learn. 2025f. Describe the components of a cloud flow. https://learn.microsoft.com/en-us/training/modules/introduction-power-automate/4-describe-components-cloud-flow. 9.5.2025.
Microsoft Learn. 2025g. Describe using Power BI to build data-driven analytics. https://learn.microsoft.com/en-us/training/modules/introduction-power-bi/2-describe-using-power-bi-build-data-driven-analytics. 9.5.2025.
Microsoft Learn. 2025h. Explore the different Power BI elements. https://learn.microsoft.com/en-us/training/modules/introduction-power-bi/3-explore-different-power-bi-elements. 9.5.2025.
Microsoft Learn. 2025i. Use Artificial Intelligence to increase productivity. https://learn.microsoft.com/en-us/training/modules/introduction-power-platform/5-use-artificial-intelligence-increase-productivity. 9.5.2025.
Ollila, M.-R. 2019. Tekoälyn etiikkaa. Helsinki: Otava.
Rekola, S. 2025. Tulevaisuuden megatrendit koulutyötä tukemassa. Opetushallitus. https://www.oph.fi/fi/opettajat-ja-kasvattajat/tulevaisuuden-megatrendit-koulutyota-tukemassa. 11.5.2025.
Sitra. 2025. Dictionary. Generative AI; creative AI; productive AI. https://www.sitra.fi/en/dictionary/generative-ai-creative-ai-productive-ai/. 1.5.2025.
Siukonen, T. & Neittaanmäki, P. 2019. Mitä tulisi tietää tekoälystä. Jyväskylä: Docendo Oy.
Ziemann, M. 2016. Tekoäly 60 vuotta – Koneen luultiin voittavan ihmisen shakissa ennen vuotta 1967. Yle. 14.7.2016. https://yle.fi/a/3-9022952. 20.4.2025.