



Low-Code alustojen hyödyntäminen teollisten prosessien hallinnassa

Matias Pienimäki

Opinnäytetyö, AMK

1.2025

Tieto- ja viestintätekniikan tutkinto-ohjelma

Pienimäki Matias

Low-Code alustojen hyödyntäminen teollisten prosessien hallinnassa

Jyväskylä: Jyväskylän ammattikorkeakoulu. **Tammikuu 2025**, 37 sivua

Tieto- ja viestintätekniikan tutkinto-ohjelma Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

Tiivistelmä

Tämä opinnäytetyö käsittelee Low-Code-alustojen hyödyntämistä teollisten prosessien hallinnassa, keskittyen erityisesti Mendix-alustaan. Low-code-tekniikat tarjoavat uusia mahdollisuuksia ohjelmistokehitykseen, etenkin prosessien nopeuttamiseen ja tehostamiseen vähäisemmällä koodauksen tarpeella (What is Low-Code Development? 2022).

Työn tavoitteena on kartoittaa Mendix-alustan sopivuutta ja soveltuvuutta teollisiin prosesseihin, eritoten automatisoinnin sekä digitalisoinnin näkökulmasta. Tutkimuksessa keskityttiin teknisten ominaisuuksien, käyttömahdollisuuksien ja käyttäjäystävällisyyden arvioimiseen. Käytännön osuudessa kehitettiin demonstraatiova projekti, jolla simuloitiin teollisen ympäristön prosessinhallintaa.

Työssä havaittiin Mendix-alustan mahdollistamaa nopeaa kehityssykliä ja joustavaa integraatiota jo olemassa oleviin järjestelmiin. Tulokset osoittivat low-code-alustojen soveltuvan erityisesti ympäristöihin, joissa tarvitaan nopeaa muutoskykyä ja mahdollisuutta kehittää prototyyppejä. Tulosten pohjalta voidaan suositella alustaa varsinkin pieniin ja keskisuuriin projekteihin, joiden resurssit ovat rajalliset.

Johtopäätöksenä todetaan low-code-alustojen potentiaalista toimia merkittävänä apuvälineenä teollisten prosessien digitalisaatiossa, sekä parantaa organisaatioiden reagointikykyä nopeasti muuttuvissa ympäristöissä.

Avainsanat (asiasanat)

Mendix, Low-code, Teollisten prosessien hallinta

Pienimäki Matias

Leveraging Low-Code Platforms for Industrial Process Management

Jyväskylä: JAMK University of Applied Sciences, January 2025, 37 pages

Degree Programme in Communication and information Technology. Bachelor's thesis.

Permission for open access publication: Yes/No

Language of publication: Finnish

Abstract

This thesis explores the usage of low-code platforms in managing industrial processes, with a specific focus on the Mendix platform. Low-code technologies offer new opportunities for software development by accelerating processes and enhancing efficiency with minimal coding requirements.

The objective of this study is to examine the suitability and applicability of the Mendix platform to industrial processes, especially from the perspectives of automation and digitalization. The research focuses on assessing technical features, potential use cases, and user-friendliness. The practical part of the study involved developing a demonstrative project simulating process management in an industrial environment.

The study showed that the Mendix platform's ability to enable rapid development cycles and easy integration with existing systems. The results demonstrated that low-code platforms are particularly well-suited for environments requiring high adaptability and the ability to develop prototypes quickly. Based on the results, the platform is recommended especially for small and medium-sized projects with limited resources.

In conclusion, low-code platforms show good potential as tools for the digitalization of industrial processes, while also improving organizations responsiveness in rapidly changing environments.

Keywords

Mendix, Low-code, Industrial process management

Sisältö

Lyhenteet ja selitteet	6
1 Johdanto.....	8
1.1 Digitalisaation vaikutus teollisuuden ohjelmistokehitykseen.....	8
1.2 Perinteisten ohjelmistokehitysmenetelmien haasteet teollisessa ympäristössä.....	8
1.3 Tavoitteet ja rajaukset.....	9
1.4 Tutkimuskysymykset.....	9
1.4.1 Kuinka hyvin Mendix-alusta soveltuu teollisten prosessien hallintaan?	9
1.4.2 Mitä etuja ja haasteita low-code-alustat tuovat verrattuna perinteisiin ohjelmistokehitysmenetelmiin?.....	9
1.4.3 Millaisia integraatio- ja skaalautuvuusmahdollisuuksia Mendix tarjoaa teollisuusympäristössä?.....	10
2 Teoreettinen viitekehys.....	10
2.1 Low-Code-alustat ja niiden merkitys teollisuudessa	10
2.2 Mendix-alustan keskeiset ominaisuudet ja soveltuvuus.....	11
2.3 Teollisten Prosessien hallinnan periaatteet ja vaatimukset	12
2.3.1 Keskeiset järjestelmät teollisuuden prosessinhallinnassa.....	12
2.3.2 Teollisten prosessien hallinnan keskeiset vaatimukset.....	13
2.3.3 Mendixin rooli teollisten prosessien hallinnassa	14
2.3.4 Mendix teollisuuden tukijärjestelmänä.....	14
3 Menetelmät.....	15
3.1 Tutkimusmenetelmät ja tiedonkeruu.....	15
3.2 Demonstraatiosovelluksen suunnittelu ja toteutus Mendix-alustalla	15
3.2.1 Sovelluksen tavoitteet ja toimintaperiaate	16
3.2.2 Tekninen toteutus	16
3.3 Eettinen tarkastelu	17
4 Sovelluksen esittely	17
4.1 Tarkoitus ja tavoitteet.....	17
4.2 Sovelluksen päätoiminnot ja käyttöliittymä	18
4.2.1 Päänäkymä – Säiliöiden hallinta.....	18
4.3 Historiatietojen tarkastelu	20
4.4 Tilamuutosten ponnahdusikkuna.....	22
4.5 Historiatiedon hallinnan Microflow:t	24
4.6 JSON-rakenne Microflow ja graafin populointi.....	25
4.6.1 Microflow:n toimintaperiaate.....	25

4.7	Säiliön lukitustilan päivittävä Microflow	26
4.7.1	Microflown toimintaperiaate.....	26
4.8	Tulostustoiminallisuuden toteutus	26
4.9	Tietomalli ja sen rakenne.....	30
4.10	Keskeiset havainnot alustan soveltuvuudesta	31
4.10.1	Kehityksen nopeus ja joustavuus	31
4.10.2	Käyttöliittymän suunnittelu ja käyttäjäkokemus	32
4.10.3	Tietomalli ja sen hallinta	32
4.10.4	Integraatiomahdollisuudet ja rajapinnat.....	32
4.10.5	Suorituskyky ja skaalautuvuus	33
5	Tulosten tarkastelu ja Pohdinta	33
5.1	Tulosten tarkastelu suhteessa tutkimuskysymyksiin	33
5.2	Tutkimuksen rajoitukset ja luotettavuus	34
5.3	Jatkokehitysehdotukset ja tulevaisuuden näkymät	35
6	Johtopäätökset ja suositukset	36
6.1	Työn keskeiset löydökset ja niiden merkitys	36
6.2	Suositukset Mendix-alustan hyödyntämiseksi teollisessa ympäristössä	37
	Lähteet	38

Kuviot

Kuvio 1.	Säiliöiden hallintanäkymä Studio Pro-sovelluksessa	19
Kuvio 2.	Säiliöiden hallintanäkymä varsinaisessa sovelluksessa	19
Kuvio 3.	Yksittäisen säiliön painikkeiden käyttötarkoitukset	20
Kuvio 4.	Historiatietojen ponnahdusikkuna	21
Kuvio 5.	Historiatietojen kehitysnäkymä Studio Pro-sovelluksessa	21
Kuvio 6.	Säiliöiden muokkausnäkömän ponnahdusikkuna	22
Kuvio 7.	Säiliöiden muokkasnäkömä Studio Pro-sovelluksessa.....	22
Kuvio 8.	Lukisematon säiliö	23
Kuvio 9.	Lukittu säiliö	23
Kuvio 10.	Säiliön historiatiedon tallentava microflow	24
Kuvio 11.	Säiliöhistorian graafin populointiin käytettävä Microflow	25
Kuvio 12.	Säiliön lukitustilan päivittävä Microflow.....	26
Kuvio 13.	Tulostuspainike käyttöliittymän yläreunassa.....	28
Kuvio 14.	Tulostuskomennon avaava microflow	28
Kuvio 15.	Tulostusdialogin avaava JavaScript	29

Kuvio 16. Selaimessa aukeava tulostusdialogi	29
Kuvio 17. Sovelluksen tietomalli.....	31

Lyhenteet ja selitteet

CI/CD (Continuous Integration / Continuous Deployment) = Jatkuva integraatio ja jatkuva käyttöönotto, ohjelmistokehitysmenetelmät, jotka mahdollistavat ohjelmistopäivitysten automatisoidun testaamisen ja julkaisemisen.

CMMS (Computerized Maintenance Management System) = Tietokoneavusteinen kunnossapidon hallintajärjestelmä, jota käytetään laitteistojen ja resurssien huollon hallintaan ja optimointiin.

DevOps (Development and Operations) = Ohjelmistokehityksen ja IT-toimintojen yhdistävä menetelmä, joka korostaa automaatiota, jatkuvaa testausta ja käyttöönottoa.

ERP (Enterprise Resource Planning) = Yrityksen resurssien suunnittelujärjestelmä, joka yhdistää liiketoiminnan eri osa-alueet, kuten taloushallinnon, tuotannon ja toimitusketjun hallinnan.

HMI (Human-Machine Interface) = Käyttöliittymä, jonka avulla käyttäjät voivat ohjata ja valvoa koneita tai laitteistoja teollisuusympäristössä. (TEPA Termipankki, Hakusana: HMI.)

IoT (Internet of Things) = Esineiden internet, järjestelmä, jossa laitteet ja sensorit ovat yhteydessä verkkoon ja kykenevät keräämään ja vaihtamaan tietoa. (TEPA Termipankki, Hakusana: IoT.)

JSON (JavaScript Object Notation) = Kevyt tietomuoto, jota käytetään tietojen siirtämiseen ja tallentamiseen erityisesti verkkosovelluksissa. (TEPA Termipankki, Hakusana: JSON.)

Low-code = Sovelluskehitystapa, jossa käytetään visuaalista käyttöliittymää ja konfiguraatiotyökaluja ohjelmiston kehittämiseen.

MES (Manufacturing Execution System) = Tuotannonohjausjärjestelmä, joka hallitsee ja seuraa tuotantoprosesseja reaaliaikaisesti.

Microflow = Mendix-alustalla käytettävä visuaalinen prosessionmallinnustyökalu.

On-premises = Pilvipalvelumalli, joka tarjoaa kehittäjille ympäristön sovellusten kehittämiseen, testaamiseen ja hallintaan.

PLM (Product Lifecycle Management) = Tuotteen elinkaaren hallintajärjestelmä, joka kattaa tuotteen suunnittelun, kehityksen, tuotannon, huollon ja hävityksen hallinnan. (TEPA Termipankki, Hakusana: PLM.)

PWA (Progressive Web Application) = Verkkosovellus, joka toimii kuten perinteinen mobiilisovellus, mutta on asennettavissa suoraan verkkoselaimesta.

REST (Representational State Transfer) = Arkkitehtoninen tyyli ohjelmistorajapintojen suunnitteluun, erityisesti verkkopalveluissa.

SCADA (Supervisory Control and Data Acquisition) = Teollinen valvonta- ja tiedonkeruujärjestelmä, jota käytetään prosessien valvontaan ja ohjaamiseen. (TEPA Termipankki, Hakusana: Scada.)

SOAP (Simple Object Access Protocol) = Viestiprotokolla, jota käytetään verkkopalveluiden tiedonsiirrossa erityisesti sovellusintegraatioissa.

Versionhallinta = Ohjelmistokehityksen prosessi, jossa hallitaan eri versioita ohjelmistosta ja sen muutoksista.

1 Johdanto

Tämän opinnäytetyön tavoite on selvittää, miten hyvin Mendix-alusta soveltuu teollisten prosessien hallintaan ja mitä etuja ja haasteita low-code-alustat tarjoavat verrattuna perinteisempiin ohjelmistonkehitysmenetelmiin. Lisäksi tarkastellaan minkälaisia integraatio- ja skaalautuvuusmahdollisuuksia Mendix tarjoaa teollisuusympäristöissä. Tutkimuskysymysten avulla pyritään arvioimaan alustan soveltuvuutta teollisten prosessien hallintaan eri näkökulmista, kuten kehityksen nopeus, käyttöliittymän suunnittelu, integraatiot ja tietomallin hallinta.

1.1 Digitalisaation vaikutus teollisuuden ohjelmistokehitykseen

Digitalisaatiosta on muotoutunut yksi keskeisimmistä tekijöistä nykyaikaisen teollisuuden kehityksessä, sen merkitys on vain kasvanut viime vuosina (Viitanen 2022). Organisaatioiden täytyy sopeutua nopeasti muuttuviin markkinoihin ja käyttöönotettava uusia teknologioita, mahdollistaakseen tehokkaamman toiminnan (Forrester 2023). Tämä murros vaatii sekä organisaatioiden rakenteiden mukauttamista, mutta myös työntekijöiden digiosaamisen jatkuvaa kehitystä.

1.2 Perinteisten ohjelmistokehitysmenetelmien haasteet teollisessa ympäristössä

Teollisuuden ohjelmistokehitys on törmännyt haasteisiin, kun perinteisemmät kehitysmenetelmät eivät vastaa enää riittävän ketterästi nykyaikaisiin vaatimuksiin. Nopeampi sovelluskehitys ja resurssien tehokkaampi hyödyntäminen ovat nousseet avaintekijöiksi kilpailukyvyyn säilyttämisessä. Low-code-alustat, kuten Mendix, tarjoavat ratkaisua kohdattuihin haasteisiin. Ne mahdollistavat sovelluskehityksen nopeuden, helppokäyttöisyyden sekä skaalautuvuuden sujuvan yhdistämisen. Uusien teknologioiden käyttöönotossa on kuitenkin myös ongelmia. Onnistunut toiminta vaatii selkeitä prosesseja, toimivia työkaluja sekä tehokasta yhteistyötä toimijoiden välillä. Low-code-alustojen hyödyntäminen teollisuuden sovelluksissa edellyttää teknisten ominaisuuksien arviointia, sekä niiden käyttäjävälisyyden ja soveltuvuuden tarkastelua käytännön työssä.

1.3 Tavoitteet ja rajaukset

Tämän opinnäytetyön tavoitteena on tutkia Mendix-alustan soveltuvuutta teollisten prosessien hallintaan ja arvioida sen tuomia etuja etenkin automatisoinnissa ja digitalisoinnissa. Ensimmäinen Mendix-pohjainen teollisuuteen suuntautuva asiakasprojekti rantautui suomeen vuoden 2024 lopulla, mutta esimerkiksi keski-Euroopassa vastaavat projektit ovat todella yleisiä, joten aihe on varsin ajankohtainen. Työssä perehdytään alustojen teknisiin ominaisuuksiin, käytännön soveltamiseen sekä niiden tarjoamiin mahdollisuuksiin nimenomaan teollisuuden näkökulmasta. Käytännön osiossa kehitetään demo, jolla havainnollistetaan Mendix-alustan hyödyntämistä teollisen ympäristön prosessinhallinnassa.

Tämän tutkimuksen tuloksia voidaan hyödyntää low-code-alustojen soveltuvuuden arvioinnissa eritoten teollisuuden tarpeisiin. Työ pyrkii vastaamaan kysymyksiin siitä, miten Mendix kykenee toimimaan työkaluna, joka yhdistää nopeuden, joustavuuden ja käyttäjäystävällisyyden teollisuuden prosessien hallinnassa.

1.4 Tutkimuskysymykset

Tässä työssä käydään läpi ja arvioidaan Mendix-alustan soveltuvuutta teolliseen prosessienhallintaan kolmen tutkimuskysymyksen avulla.

1.4.1 Kuinka hyvin Mendix-alusta soveltuu teollisten prosessien hallintaan?

Teollisessa prosessien hallinnassa käytetään useita olemassa olevia järjestelmiä ja työkaluja, jotka tukevat esimerkiksi tuotannon ja resurssien ohjaamista. Tässä työssä tarkastellaan ja arvioidaan nimenomaan Mendix-alustan sopivuutta näihin tarkoituksiin, varsinkin sen visuaalisen kehitysmallin ja mukautettavuuden näkökulmista.

1.4.2 Mitä etuja ja haasteita low-code-alustat tuovat verrattuna perinteisiin ohjelmistokehitysmenetelmiin?

Low-code alustojen yleistyminen on muuttanut järjestelmien toteutustapoja. Työssä käydään läpi ja vertaillaan low-code-kehityksen toimintoja perinteisiin koodipohjaisiin vaihtoehtoihin.

1.4.3 Millaisia integraatio- ja skaalautuvuusmahdollisuuksia Mendix tarjoaa teollisuusympäristössä?

Teollisessa ohjelmistokehityksessä integraatiot muihin järjestelmiin ovat keskeisiä. Työssä arvioidaan millaisia rajapintoja Mendix tarjoaa kyseisten tarpeiden täyttämiseksi.

2 Teorettinen viitekehys

Työn teorettisessa viitekehyksessä käsitellään low-code alustoja ja niiden roolia erityisesti teollisten prosessien hallinnassa. Digitalisaation myötä ohjelmistokehityksen joustavuus ja nopeus on muodostuneet entistä tärkeämmiksi, jolloin perinteiset kehitysmenetelmät eivät aina kykene vastamaan nykypäiväisiin vaatimuksiin. Tässä luvussa tarkastellaan low-code-tekniikan keskeisiä piirteitä, Mendix alustan erityispiirteitä ja teollisten prosessien hallinnan perusteita.

2.1 Low-Code-alustat ja niiden merkitys teollisuudessa

Low-code-ympäristöt tarjoavat visuaalisen käyttöliittymän, jossa sovelluksia voidaan rakentaa ilman laajempaa syvälistä ohjelmointiosaamista. Tämä mahdollistaa nopeamman ja joustavamman kehityksen, sekä pienentää IT-osastojen työkuormaa (Forrester 2025). "Gartnerin mukaan low-code-alustojen käyttö voi nopeuttaa sovelluskehitystä jopa 50–70 % verrattuna perinteisiin menetelmiin (Gartner forecast on low-code development technologies 2023)." Teollisuudessa low-code-alustoja on käytetty erilaisissa sovelluksissa, esimerkiksi kunnossapidon hallintajärjestelmissä (CMMS) ja toimitusketjujen optimoinnissa, joissa järjestelmien nopea mukauttaminen on tärkeää. Gartnerin mukaan low-code-tekniikoiden käyttö on yleistymässä merkittävästi. On ennustettu, että vuoteen 2026 mennessä 80 % low-code-sovellusten käyttäjistä tulee olemaan IT-osastojen ulkopuolelta, kun osuus oli 60 % vuonna 2021 (Gartner forecast on low-code development technologies 2023). Lisäksi Gartner arvioi, että vuoteen 2025 mennessä 70 % uusista sovelluksista kehitetään low-code- tai no-code-tekniikoilla, mikä on huomattava kasvu verrattuna alle 25 %:iin vuonna 2023 (Gartner forecast on low-code development technologies 2023). Nämä ennusteet korostavat low-code-alustojen kasvavaa merkitystä ohjelmistokehityksessä ja niiden potentiaalia tehostaa teollisten prosessien hallintaa. Low-code-alustojen joustavuus ja mukautettavuus ovat keskeisiä etuja, mutta rajoitukset suurten tietomäärien käsittelyssä voivat heikentää suorituskykyä ja vaikuttaa soveltuvuuteen monimutkaisemmissa teollisissa sovelluksissa (Martinez & Pfister 2023).

2.2 Mendix-alustan keskeiset ominaisuudet ja soveltuvuus

Mendix on yksi markkinoiden johtavista low-code-alustoista, se mahdollistaa monimutkaisten sovellusten kehittämisen ilman perinteistä ohjelmointia (What is Low-Code Development? 2022). Muita vastaavia työkaluja ovat esimerkiksi Microsoft PowerApps, OutSystems ja Appian. Näillä kaikilla on omat vahvuutensa ja käyttökohteensa, mutta tähän työhön valittiin erityisesti Mendix sen teollisen soveltuvuuden vuoksi (Owen & Tozzi 2025). Mendixin vahvuuksia ovat muun muassa valmiit integraatiot, käyttäjäystävällinen kehitysympäristö ja joustavat käyttöönottovaihtoehdot. Sen käyttö laajemmin vaatii kuitenkin laajempaa perehtymistä etenkin, jos halutaan räätälöidä sovelluksia kattavammalla tasolla.

Mendixin visuaalinen kehitysympäristö tarjoaa drag-and-drop tyyllisen käyttöliittymän, jossa voit liikutella elementtejä ilman perinteistä koodaamista. Tämä mahdollistaa nopeamman kehityksen ja tekee siitä saavutettavampaa myös, esimerkiksi liiketoimintayksiköille. Mendix sisältää kaksi erilaista kehitysympäristöä

- Mendix Studio, joka on suunnattu ei-teknisille käyttäjille ja mahdollistaa yksinkertaisten sovellusten luomisen.
- Mendix Studio Pro, joka tarjoaa laajempia muokkausmahdollisuuksia kokeneemmille kehittäjille.

Mendix tukee myös valmiita integraatioita muun muassa SAP:n, IoT-järjestelmien ERP-ohjelmistojen ja teollisten automaatiojärjestelmien kanssa (Security Overview 2023; Top 4 Ways to Integrate SCADA and MES to ERP Systems 2020). Lisäksi alustan perusominaisuuksia voidaan myös laajentaa Java- ja/tai JavaScript-koodilla, mikä mahdollistaa monimutkaisempien toimintojen kehittämisen.

Alustan joustavuuden lisäämiseksi Mendix voidaan käyttöönottaa myös pilvipalveluna (Mendix Cloud, AWS, Azure, Google Cloud) tai on-premises-ratkaisuna. Nämä seikat tekevät siitä joustavan vaihtoehdon yrityksille, joilla on tiukat tietoturva- ja hallintavaatimukset. Valmiit sovellukset voidaan julkaista web-, mobiili- ja PWA-muodossa (Progressive Web App) ilman erillistä koodausta.

Mendixiin kuuluvat automaattinen versionhallinta ja DevOps-tuki helpottavat kehitystiimien yhteistyötä. Myös CI/CD (Continuous Integration / Continuous Deployment) mahdollistavat nopeat julkaisut ilman manuaalista asennustyötä. Tämä nopeuttaa sovellusten käyttöönottoa ja ylläpitoa.

Mendixin sisäinen microflow-rakenne on prosessinohjaustyökalu, jonka avulla sovelluksen logiikka rakennetaan visuaalisesti ilman perinteistä koodausta. Microflow mahdollistaa esimerkiksi tietojen tallennuksen, hakujen ja ehtojen käsittelyn sekä ulkoisten järjestelmien integroinnin (What is Low-Code Development? 2022).

2.3 Teollisten Prosessien hallinnan periaatteet ja vaatimukset

Teollisten prosessien hallinta tarkoittaa tuotannon, logistiikan ja kunnossapidon prosessien valvontaa ja optimointia. Yleisesti käytetyt järjestelmät, kuten MES- ja SCADA-järjestelmät, tarjoavat reaaliaikaista tietoa eri prosessien tilasta (Korkko 2023). Low-code-alustat, kuten Mendix, kykenevät täydentämään näitä järjestelmiä tarjoamalla nopeasti kustomisoitavia käyttöliittymiä sekä analytiikkatyökaluja (iFIX HMI / SCADA 2023). Järjestelmällinen prosessien hallinta on todella tärkeää teollisen toiminnan tehokkuuden ja kustannusäästöjen kannalta. Tällainen hallinta tarvitsee reaaliaikaista dataa, automaattista analytiikkaa sekä tehokkaita järjestelmiä.

2.3.1 Keskeiset järjestelmät teollisuuden prosessinhallinnassa

MES-järjestelmät (Manufacturing Execution System), joissa seurataan tuotannon etenemistä ja tehokkuutta reaaliajassa, sekä mahdollistetaan tuotannon optimoinnin ja laadunvalvonnan keräämällä dataa esimerkiksi tuotannon nopeudesta, virheistä ja resurssien käytöstä.

MOM-järjestelmät (Manufacturing Operations Management), mitkä yhdistävät tuotannonhallinnan, laadunvalvonnan, kunnossapidon ja logistiikan yhdeksi järjestelmäksi. MOM toimii MES-, SCADA- ja ERP-järjestelmien yläpuolella koordinoien niiden toimintaa. Ne mahdollistavat tuotantoprosessin täysvaltaisen optimoinnin sekä standardoinnin.

SCADA-järjestelmät (Supervisory Control and Data Acquisition), joissa valvotaan ja ohjataan teollisuuden eri automaatiojärjestelmiä, sekä kerätään tietoa esimerkiksi anturien datasta, energiankulutuksesta ja eri laitteiden statuksesta. Ne mahdollistavat myös reaaliaikaisen valvonnan ja hälytykset, tämä on keskeinen ominaisuus esimerkiksi energiateollisuudessa.

ERP-järjestelmät (Enterprise Resource Planning), mitkä ovat kokonaisvaltaisia yritysten resurssienhallinnan järjestelmiä, jotka yhdistävät talouden, henkilöstöhallinnon, tuotannon ja toimitusketjun. Ne integroituvat usein MES- ja SCADA-järjestelmien kanssa, jotta operatiivinen data voitaisiin yhdistää liiketoimintapäätöksiin.

PLM-järjestelmät (Product Lifecycle Management), jotka keskittyvät koko elinkaaren hallintaan, aina suunnittelusta tuotantoon sekä ylläpitoon. Ne mahdollistavat tuotesuunnittelun, version hallinnan ja dokumentaation hallinnan yhdessä järjestelmässä. Ne kykenevät integroitumaan ERP- ja MES-järjestelmiin, jotta tuotantoprosessit voivat hyödyntää tuotekehityksestä saatua tietoa.

2.3.2 Teollisten prosessien hallinnan keskeiset vaatimukset

Teollisuuden sovelluksille asetetaan usein korkeat vaatimukset, sillä niiden täytyy toimia luotettavasti, skaalautuvasti ja reaaliaikaisesti. Keskeiset vaatimukset ovat:

Reaaliaikainen tiedonkäsittely, sovellusten täytyy pystyä keräämään näyttämään ja analysoimaan dataa välittömästi. Tämä on tärkeää esimerkiksi kunnossapidon hallinnassa, jossa pyritään havaitsemaan laitteistovikoja ennakoivasti.

Joustavuus ja mukautuvuus, teollisuuden prosessit vaihtelevat riippuen yrityksestä, joten niitä pitää pystyä räätälöimään helposti. Nimenomaan Low-Code järjestelmät, kuten Mendix tarjoavat visuaalisia työkaluja, joiden avulla yritykset kykenevät muokkaamaan sovelluksia ilman laajempaa osaamista ohjelmoinnissa.

Integrointikyky, nykyaikaisten järjestelmien täytyy kyetä yhdistämään data eri lähteistä, esimerkiksi IoT-laitteistosta, SCADA- ja MES-järjestelmistä ja ulkoisista tietokannoista. Mendix tukee tätä mahdollistamalla REST- ja SOAP-rajapinnat ja valmiina olevia integraatiomahdollisuuksia eri tunnettuihin teollisuuden järjestelmiin.

Tietoturvallisuus ja vikasetoisuus, teollisuusjärjestelmissä käsitellään luottamuksellista tietoa, joten korkea tietoturvan taso on kriittistä ylläpitää. Mendixin tarjoamat pilvipohjaiset ja on-premises-käyttönotot mahdollistavat yrityskohtaisen turvallisuuden vaatimusten täyttämisen (What is Low-Code Development? 2022).

Automaatio ja analytiikka, järjestelmien täytyy tarjota mahdollisuus prosessien automatisointiin, esimerkiksi ennakoivaan kunnossapitoon ja tehokkuuksien optimointiin. Mendixiin pystytään liittämään eri analytiikkatyökaluja ja koneoppimismalleja, mitkä voivat auttaa esimerkiksi eri vikojen ennustamisessa (Security Overview 2023).

2.3.3 Mendixin rooli teollisten prosessien hallinnassa

Low-code-alustat, kuten Mendix, kykenevät täydentämään perinteisiä teollisuuden järjestelmiä tarjoamalla käyttäjäystävällisemmän järjestelmän datan analysoimiseen ja hallintaan. Mendixillä kyetään luomaan järjestelmiä, joilla seurataan esimerkiksi säiliöiden tai varastojen statusta ja esimerkiksi täyttöastetta. Myös kunnossapidon hallintaan tarkoitettut mobiili- tai web-sovellukset esimerkiksi huoltotiimin käyttöön sekä tuotannon ohjaus valmiilla MES- ja SCADA-järjestelmien integraatioilla ja tuotantotietojen visuaalisella esittämisellä esimerkiksi dashboard-näkymässä ovat selkeitä Mendixin vahvuuksia.

Vaikka Mendix tarjoaa valmiita komponentteja, se ei täysin korvaa perinteisiä SCADA- tai MES-järjestelmiä, mitkä vaativat syvällisempää reaaliaikaista ohjausta. Suurten tietomäärien hallinta ja käsittely Mendixillä voi olla hitaampaa, jos sitä ei erityisesti optimoida tällaisiin tarkoituksiin. Kompleksit teolliset laskentaprosessit voivat myös vaatia lisäksi Java- tai Python-koodausta, mikä voi hidastaa kokonaisprosessia kehittämistyössä.

2.3.4 Mendix teollisuuden tukijärjestelmänä

Mendix ei suoraan kykene korvaamaan SCADA- tai MES-järjestelmiä, mutta se tarjoaa joustavan ja nopeasti kehitettävän lisäkerroksen näiden rinnalle. Mendixin avulla kyetään kehittämään käyttöliittymiä, raportointityökaluja ja mobiilisovelluksia, mitkä helpottavat huomattavasti esimerkiksi datan visualisointia ja päätöksenteon tukemista.

Perinteiset menetelmät ovat useasti hitaampia ja kalliimpia, siksi Mendixin kaltainen low-code-alusta voi olla erityisen hyödyllinen varsinkin pienille ja keskisuurille yrityksille, jotka haluavat digitalisoida prosessejaan ilman mittavia investointeja (What is Low-Code Development? 2022).

3 Menetelmät

3.1 Tutkimusmenetelmät ja tiedonkeruu

Tässä tutkimuksessa käytettiin kokeellista lähestymistapaa, missä Mendix-alustan soveltuvuutta teollisten prosessien hallintaan arvioidaan käytännön demonstraation avulla. Tiedonkeruu perustuu Mendix-alustan dokumentaatioon, sekä sovelluksen kehittämiseen ja testaamiseen. Lisäksi analysoidaan Mendix:in vertautuvuutta perinteisiin ohjelmistokehitysmenetelmiin, nopeuden, joustavuuden ja ylläpidettävyyden näkökulmasta.

3.2 Demonstraatiosovelluksen suunnittelu ja toteutus Mendix-alustalla

Demonstraatiosovelluksen tarkoituksena on hallita ja valvoa teollisten säiliöiden tilatietoja visuaalisesti. Sovellus tarjoaa käyttöliittymän, jossa käyttäjät voivat tarkastella, sekä päivittää säiliöiden tilaa. Lisäksi sovellus kykenee tallentamaan historiatietoa muutoksista ja tarjoaa helposti luettavan analytiikan säiliöiden käyttöasteesta. Sovellus kehitettiin Mendix-alustalla hyödyntäen alustan omaa visuaalista mallinnustyökalua, missä tietomalli ja käyttöliittymä suunniteltiin täysin ilman perinteistä ohjelmointia. Tiedonkeruu ja menetelmät pohjautuivat Mendix-alustan viralliseen dokumentaatioon ja aiempiin tutkimuksiin ja julkaisuihin low-code alustojen käytöstä teollisuudessa. Käytännön kehitys ja testaus toteutettiin Mendix-alustalla, missä prototyyppi kehitettiin ja testattiin. Myös demonstraatiosovelluksen käyttökokemusta arvioitiin kyseisellä alustalla. Lopuksi analysoitiin kehitysnopeutta verrattuna perinteisiin ohjelmistokehitysmenetelmiin ja arvioitiin sovelluksen muokkaamisen ja ylläpitämiseen verrattuna koodipohjaisiin ratkaisuihin. Tämä yhdistelmä teoriapohjaista analyysia ja käytännön kokeilua mahdollisti kattavan arvioinnin siitä, missä määrin Mendix-pohjainen alusta soveltuu teollisuuden eri tarpeisiin.

3.2.1 Sovelluksen tavoitteet ja toimintaperiaate

Demonstraatiosovelluksen tarkoitus on hallita ja valvoa simuloitujen teollisten säiliöiden tilatietoja visuaalisesti. Sovellus toimii käyttäjäystävällisenä työkaluna, mahdollistaen reaaliaikaisen tiedon seurannan ja muutosten tallennuksen sekä historiatietojen tarkastelun visuaalisessa muodossa.

Sovellukselle asetetut keskeiset vaatimukset ennen toteutusta olivat:

- Selkeä ja helppokäyttöinen käyttöliittymä
- Reaaliaikainen tiedonseuranta ja päivitys
- Säiliöiden statusten muokkaus ilman erillistä ohjelmointia
- Muutosten historiatallennus ja tarkastelumahdollisuus
- Tietojen graafinen esittely, erillisessä näkymässä

Näiden vaatimusten pohjalta rakennettiin sovelluksen päätoiminnot:

- Säiliöiden tilatietojen hallinta, käyttäjät voivat tarkastella ja päivittää säiliöiden statusta. Näiden tietojen muokkaaminen onnistuu käyttöliittymän kautta ilman erillistä koodin kirjoittamista.
- Historiatietojen tallennus, sovellus tallentaa kaikki muutokset säiliöiden tiloissa, jolloin käyttäjät voivat tarkistella aikaisempia muutoksia ja käyttötilastoja.
- Datan visualisointi, käyttäjä kykenee tarkastelemaan säiliöiden täyttöastetta ja tilatietoja graafisessa muodossa. Sovellus tarjoaa yksinkertaisen dashboard-näkymän, missä keskeinen data visualisoidaan ja esitetään selkeästi

3.2.2 Tekninen toteutus

Sovellus kehitettiin Mendix-low-code-alustalla, mikä mahdollistaa visuaalisen kehitystyön ilman perinteisiä ohjelmointikeinoja. Mendix soveltuu eritoten teollisten prosessien hallinnan kehittämiseen sen tarjoamien työkalujen takia. Prosessi eteni seuraavasti:

1. Tietomallin suunnittelu, Sovelluksen tietomalli rakennettiin Domain model-työkalulla, mikä mahdollisti säiliöiden tietorakenteen määrittämisen sekä muutosten, että historiatietojen tallentamisen.
2. Käyttöliittymän rakentaminen, sovelluksen käyttöliittymä toteutettiin Studio Pro -ympäristössä, jossa kaikki komponentit lisättiin drag-and-drop menetelmällä. Käyttäjät voivat valita tietyn säiliön, tarkastella sen tilaa ja muokata sitä yksinkertaisessa valikossa. Sovellukseen sisältyy myös dashboard-näkymä, missä esitellään säiliöiden dataa graafisesti.
3. Logiikan ja automaatioiden toteutus, Mendixin microflow-ominaisuuksia käytettiin automatisoimaan muutosten tallentaminen. Mikäli säiliöiden tilaa muokataan järjestelmä tallentaa automaattisesti aikaleiman.
4. Integraatiot ja skaalautuvuus, sovellus luotiin skaalautuvaksi, jolloin sitä voitaisiin laajentaa esimerkiksi IoT-laitteiden tiladatan hyödyntämiseksi.

3.3 Eettinen tarkastelu

Työssä noudatettiin eettisiä periaatteita. Lähteitä käytettiin asianmukaisesti ja tutkimusmenelmät esiteltiin avoimesti ja läpinäkyvästi. Kaikki työssä käytetyt lähteet on merkitty lähdeluetteloon ja tekstiin viitetekniikan mukaan. Työssä ei myöskään käsitelty henkilö- tai muita arkaluonteisia tietoja.

4 Sovelluksen esittely

Tässä osiossa esitellään Mendix-alustalla luotu teollisten säiliöiden hallintasovellus, jota käytetään demonstroivana esimerkkinä low-code-alustan soveltuvuudesta nimenomaan teollisten prosessien hallintaan. Sovellus tarjoaa reaaliaikaisen käyttöliittymän teollisten säiliöiden tilahallintaan, historiallisen tietojen tarkasteluun ja analytiikkaan.

4.1 Tarkoitus ja tavoitteet

Sovelluksen tavoitteena on parantaa säiliöiden tilatietojen hallintaa ja esitellä käyttäjäystävällinen käyttöliittymä niiden tilan päivittämiseen sekä historiatietojen tarkasteluun. Sovellus on kehitetty täysin Mendix-alustalla ilman perinteistä ohjelmointia, hyödyntäen alustan sisäänrakennettua drag-and-drop työkaluja sekä microflow logiikoita.

Keskeisiä ominaisuuksia ovat:

Säiliöiden tilan hallinta, jolla käyttäjät voivat päivittää säiliöiden tilaa eri vaihtoehtojen välillä (tyhjä, käytettävissä, käytössä ja lukittu).

Historiatietojen tallennus, jossa kaikki tilaan tehdyt muutokset tallennetaan ja ne ovat tarkasteltavissa jälkikäteen tietyllä aikavälillä.

Visuaaliset raportit, säiliöiden käyttöasteita ja tilamuutoksia voidaan tarkistella graafisesti.

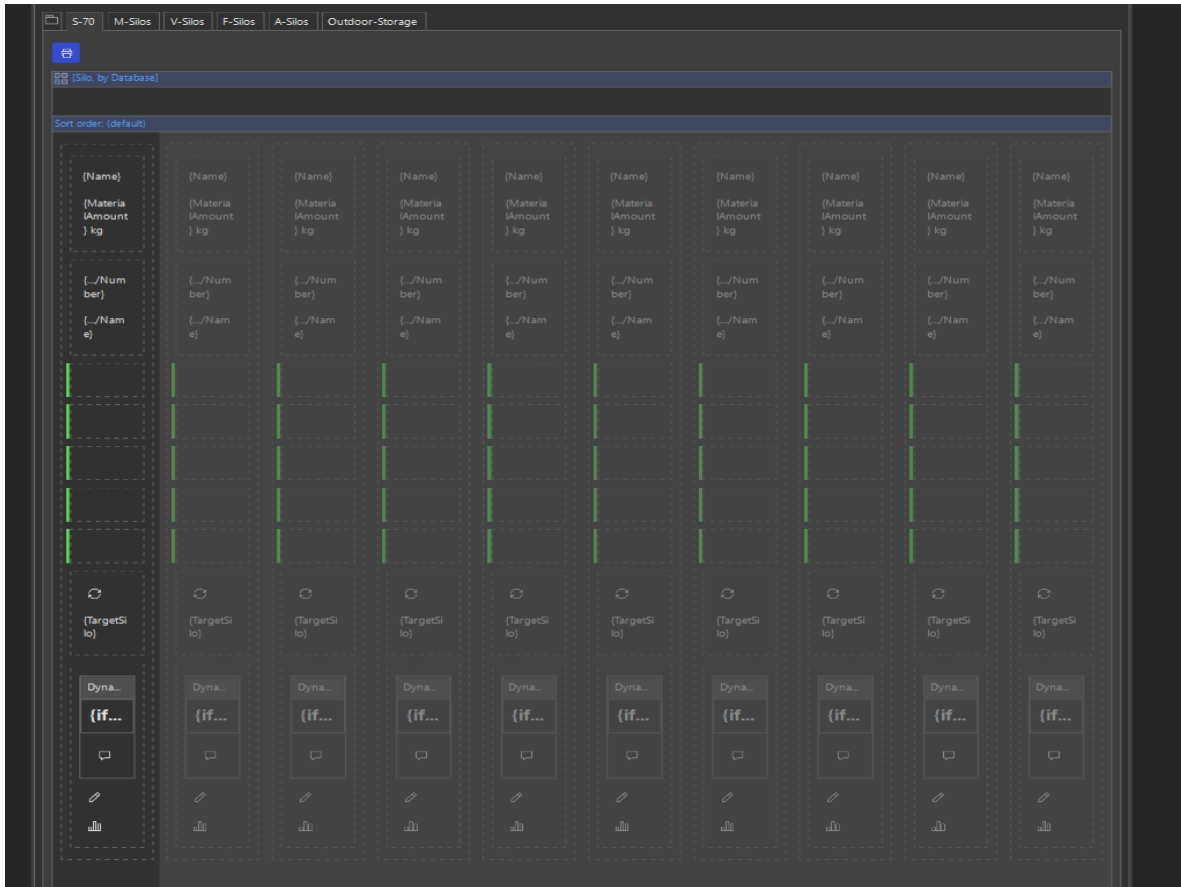
Tavoitteena on siis luoda helppokäyttöinen teollisuuden järjestelmän osa, jolla voidaan tarkastella säiliöihin liittyvää dataa jouhevasti.

4.2 Sovelluksen päätoiminnot ja käyttöliittymä

Tässä osiossa käydään läpi sovelluksen keskeiset näkymät ja niiden toiminnallisuudet. Kuvankaappauksilla havainnollistetaan sovelluksen, sekä käyttöliittymän kokonaisrakennetta. Samasta rakenteesta esitellään kaksi kuvankaappausta, toinen sovelluksen kehitysnäkymästä ja toinen loppukäyttäjälle näkyvästä sovelluksesta.

4.2.1 Päänäkymä – Säiliöiden hallinta

Kehitysnäkymä (ks. kuvio 1) esittelee käytettyjä elementtejä, joiden avulla päänäkymä on rakennettu. Sovelluksen päänäkymä (ks. kuvio 2) antaa käyttäjälle yleiskuvan kaikista säiliöistä ja niiden tämänhetkisestä tilasta. Tilatietojen seuraaminen on keskeinen osa teollisten prosessien hallintaa, sillä reaaliaikainen tieto säiliöiden täyttöasteesta ja statuksesta auttaa optimoimaan tuotannon tehokkuutta.

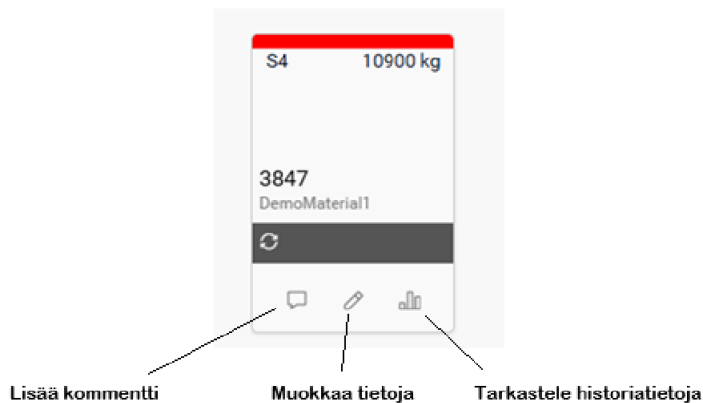


Kuvio 1. Säiliöiden hallintanäkymä Studio Pro-sovelluksessa



Kuvio 2. Säiliöiden hallintanäkymä varsinaisessa sovelluksessa

Kun käyttäjä valitsee tietyn säiliön, voidaan tarkastella kyseisen säiliön historiatietoja tai päivittää tilaa. Tilapäivityksiä voivat tehdä ainoastaan ylläpitäjät, mikä estää vahingossa tapahtuvia muutoksia. Jokaisessa näkyvässä säiliössä (ks. kuvio 3) on kolme eri painiketta, joilla mahdollistetaan säiliöiden hallinta. Painikkeiden avulla käyttäjä voi lisätä kommentteja, tarkastella historiatietoja ja muokata säiliön tietoja.



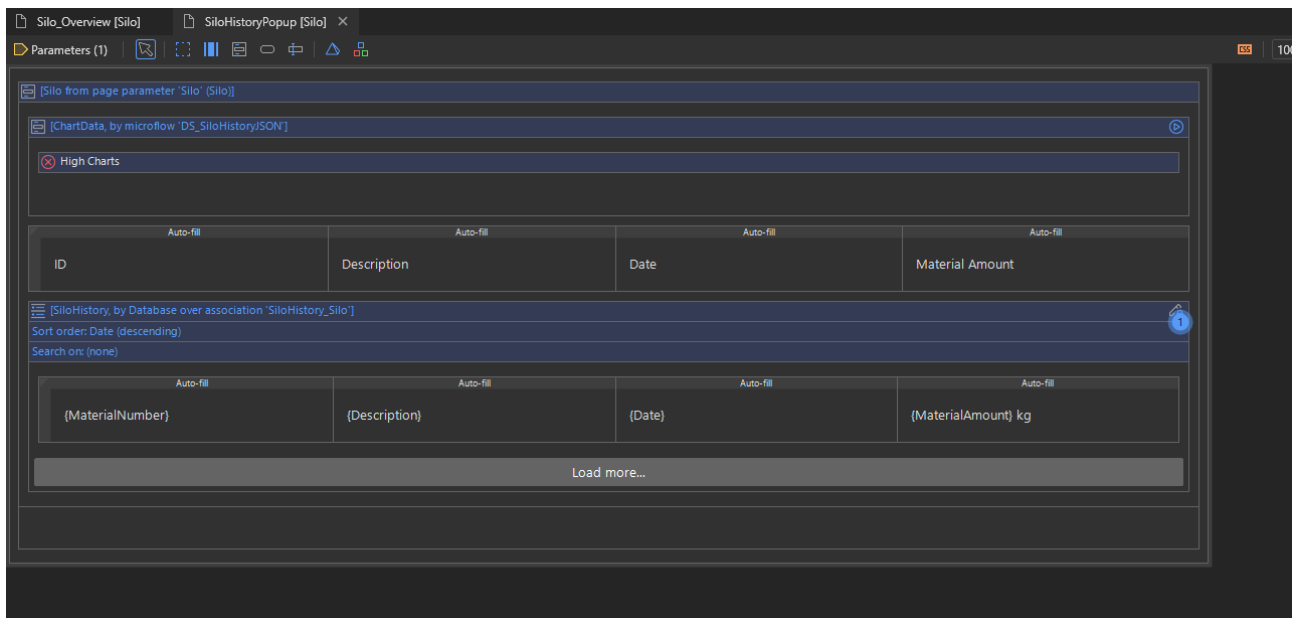
Kuvio 3. Yksittäisen säiliön painikkeiden käyttötarkoitukset

4.3 Historiatietojen tarkastelu

Historiatietojen tarkastelu on keskeinen toiminto, millä mahdollistetaan säiliöiden täyttöasteen ja tilanmuutosten analysointi. Nämä tiedot auttavat ennakoimaan huoltovälejä ja tunnistamaan mahdollisia ongelmia tuotannossa. Painamalla historiatietojen tarkastelupainiketta (ks. Kuvio 3) Sovellus avaa historiatietojen tarkasteluun luodun ponnahdusikkunan (ks. Kuvio 4), joka sisältää graafisen näkymän säiliön täyttöhistoriasta sekä listan tilanmuutoksista ajanjaksolta, joka on rajoitettu viimeiseen yhdeksään kuukauteen, tällä vältytään ylimääräisen vanhan datan säilömiseltä. Kehitysnäkymä (ks. Kuvio 5) selkeyttää ponnahdusikkunan luomisessa käytettyjä elementtejä.



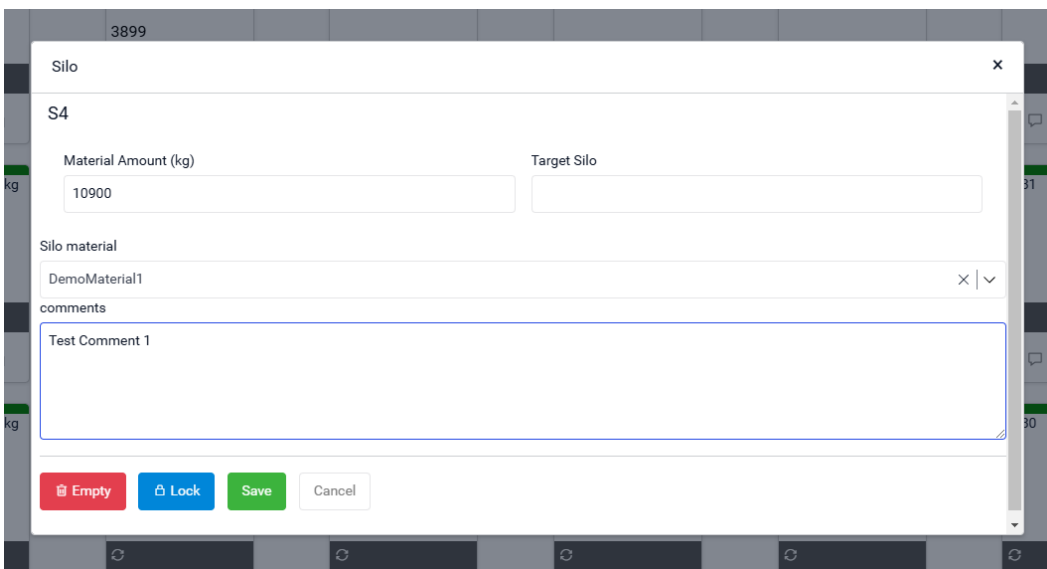
Kuvio 4. Historiatietojen ponnahdusikkuna



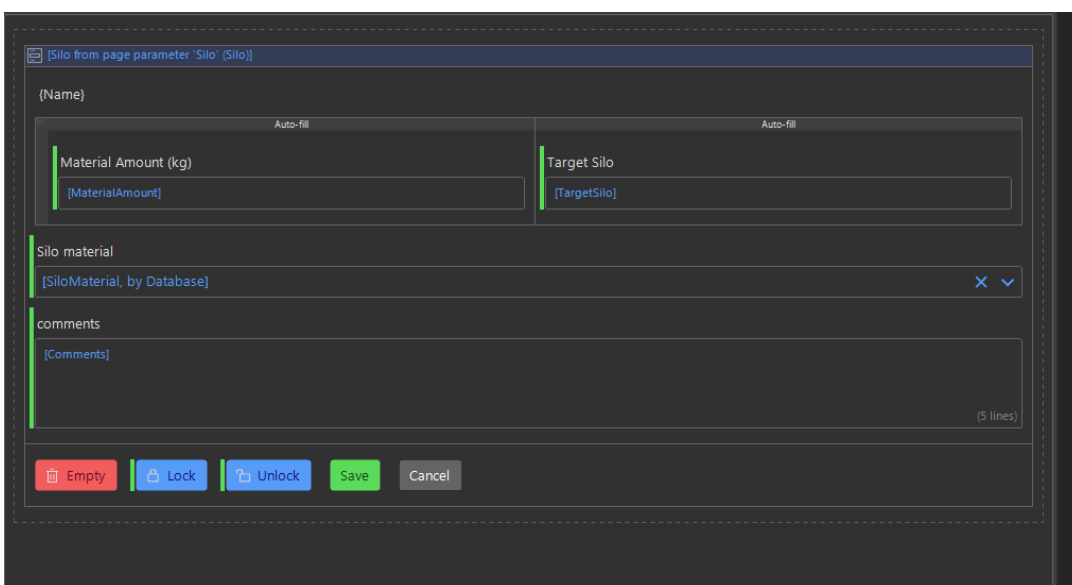
Kuvio 5. Historiatietojen kehitysnäkymä Studio Pro-sovelluksessa

4.4 Tilamuutosten ponnahtusikkuna

Säiliöiden tilan hallinta on olennainen osa sovelluksen toimintoja. Se mahdollistaa säiliöiden sisältämän materiaalin ja käyttötilan reaaliaikaisen päivittämisen. Painamalla tietojen muokkauspainiketta (ks. Kuvio 3) Sovellus avaa ponnahtusikkunan (ks. Kuvio 6), jossa kyetään muokkaamaan materiaalin määrää, vaihtaa kohdesäiliötä ja lisätä kommentteja, tallennettaessa muokattu data on suoraan nähtävillä ilman viivettä. Kehitysnäkymä (ks. Kuvio 7) havainnollistaa kehitystyössä käytetyt elementit.

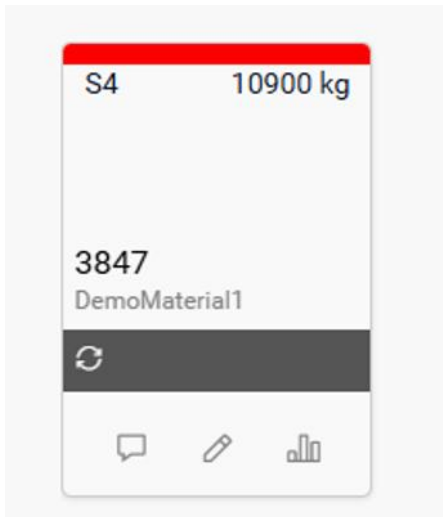


Kuvio 6. Säiliöiden muokkausnäkömän ponnahtusikkuna

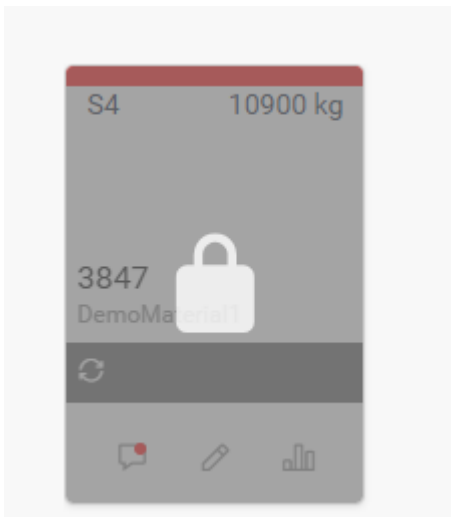


Kuvio 7. Säiliöiden muokkasnäkömä Studio Pro-sovelluksessa

Tyhjäyspainiketta (ks. Kuvio 6) painamalla asetetaan säiliön materiaalimäärä nolnaan, sekä tyhjä-tään materiaali, jolloin historiatietojen ponnahdusikkuna (ks. Kuvio 4) näyttää automaattisen tekstin, joka kertoo säiliön tulleen tyhjätyksi. Lukituspainiketta painamalla aktivoitu lukitustila estää säiliön käytön, jolloin se ei ole enää valittavissa. Lukitun säiliön tila näkyy myös visuaalisesti käyttäjälle (ks. Kuvio 9), jolloin käyttäjät eivät voi vahingossa suorittaa siihen liittyviä toimenpiteitä.



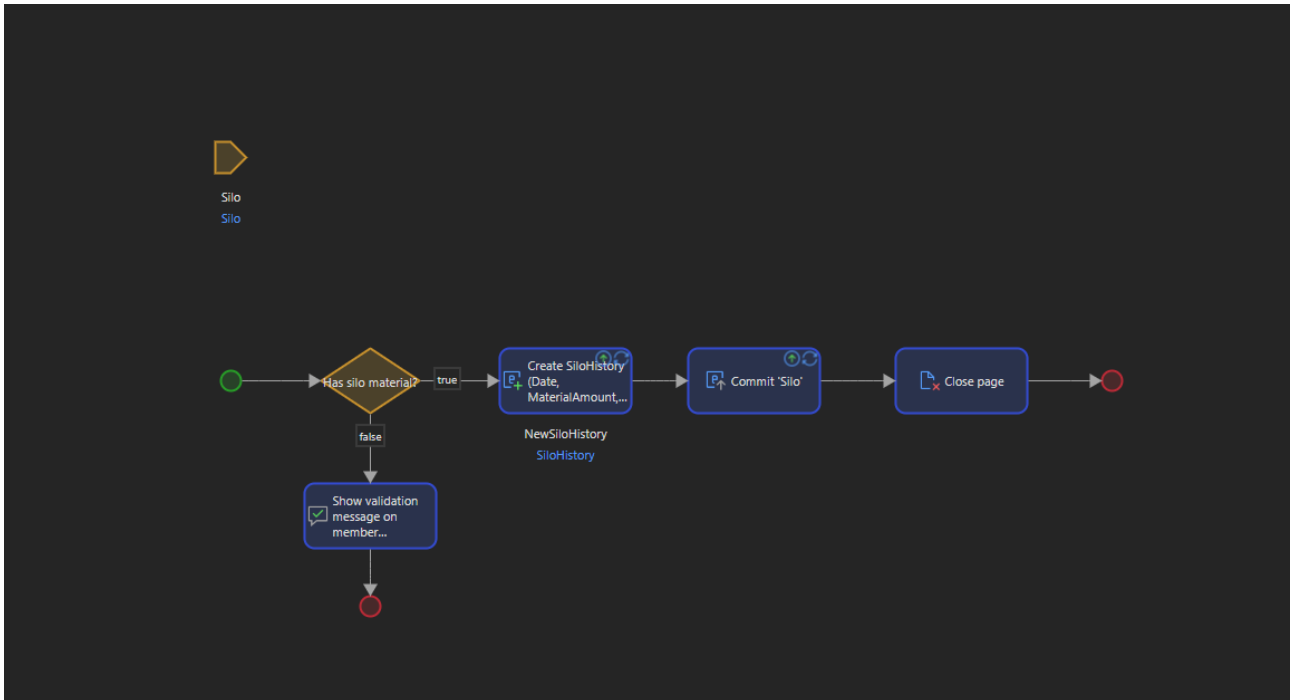
Kuvio 8. Lukisematon säiliö



Kuvio 9. Lukittu säiliö

4.5 Historiatiedon hallinnan Microflow:t

Säiliöiden historiatiedon hallinnan logiikasta vastaavat microflowt, joiden avulla uuden datan rakentaminen mahdollistetaan. Kun käyttäjä painaa tallennuspainiketta (ks. Kuvio 6.) sovellus käynnistää microflow-logiikan (ks. Kuvio 10.), joka tallentaa uuden historiatiedon tietokantaan.

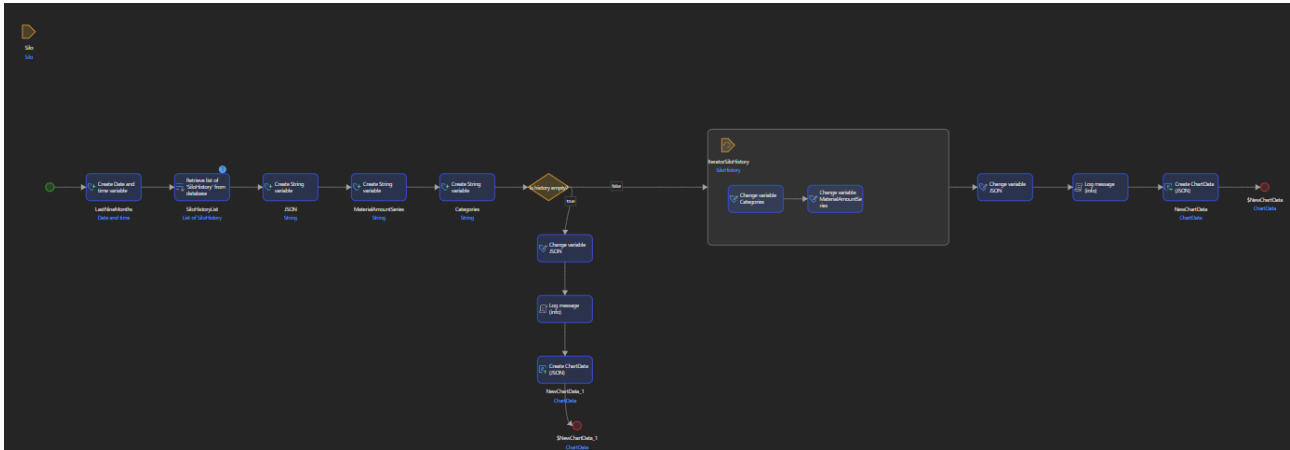


Kuvio 10. Säiliön historiatiedon tallentava microflow

Microflow koostuu seuraavista keskeisistä vaiheista: **Ehtotarkistus:** Ensimmäinen lohko ”Has silo material” tarkistaa, onko säiliöllä materiaalimäärää. Jos arvo on tyhjä, prosessi keskeytyy ja käyttäjälle näytetään ilmoitus. **Historiatietojen luonti:** Jos säiliössä on materiaalia, microflow jatkaa ”Create SiloHistory” -toimintoon, jossa luodaan uusi historiatieto-kokonaisuus. Tämä kokonaisuus tallentaa päivämäärän (Date), materiaalimäärän (MaterialAmount) ja säiliön tunniste (Silo). **Tietojen tallennus:** Kun historiatieto on luotu, se tallennetaan tietokantaan ”Commit Silo”-toiminolla, mikä varmistaa uuden päivityksen näkyvyyden käyttöliittymässä. **Näkymän sulkeminen:** Kun tiedot on tallennettu onnistuneesti ”Close page” -toiminto sulkee näkymän ja palauttaa käyttäjän takaisin pääsivulle.

4.6 JSON-rakenne Microflow ja graafin populointi

Graafin luominen ja populointi sovelluksessa vaatii strukturoitua tiedonkäsittelyä, jossa JSON-muotoista dataa käytetään tiedon oikeaoppiseen järjestämiseen ja visualisointiin. Tämä microflow vastaa datan muuntamisesta ja valmistelusta, jotta se voidaan esittää graafisessa muodossa käyttöliittymässä (ks. Kuvio 11.)



Kuvio 11. Säiliöhistorian graafin populointiin käytettävä Microflow

4.6.1 Microflow:n toimintaperiaate

Tämä Microflow suorittaa useita datan käsittelyvaiheita, jotka mahdollistavat säiliöhistorian näyttämisen graafisessa muodossa. **Tiedonkeruu:** Microflow aloittaa luomalla tyhjän listan, johon graafissa näytettävät arvot tallennetaan. Se hakee säiliöhistorian tiedot tietokannasta ja järjestää ne aikajärjestykseen. **JSON-rakenteen muodostaminen:** Luodaan JSON-muuttuja, johon lisätään graafissa tarvittavat tiedot. Microflow muuntaa säiliöhistorian arvot JSON-rakenteeksi, jotta ne voidaan näyttää käyttöliittymässä. JSON sisältää esimerkiksi seuraavat tiedot päivämäärä, materiaali määrä ja muutostyyppi (lisäys, tyhjennys)

Tietojen suodatus ja muokkaus: Microflow tarkistaa, ovatko tietyt ehdot täyttyneet ennen datan lisäämistä listaan. Muuttujia käsitellään ja suodatetaan, jotta graafi näyttää vain olennaiset muutokset. **Tietojen valmistelu käyttöliittymää varten:** Lopuksi JSON-tietorakenne muunnetaan käyttöliittymän tukemaan muotoon ja syötetään graafin tietolähteeksi. Data lähetetään käyttöliittymän visualisointikomponentille, joka piirtää graafin kerättyjen arvojen perusteella.

4.7 Säiliön lukitustilan päivittävä Microflow

Säiliön lukitseminen on olennainen toiminto, joka estää säiliön käytön ja päivityksen, kunnes lukitus poistetaan. Tämä Microflow päivittää säiliön lukitustilan ja varmistaa, että muutos näkyy käyttöliittymässä reaaliajassa (ks. Kuvio 12).



Kuvio 12. Säiliön lukitustilan päivittävä Microflow

4.7.1 Microflown toimintaperiaate

Tämä Microflow on yksinkertainen mutta tärkeä prosessi, joka sisältää seuraavat vaiheet: **Lukitustilan muuttaminen:** Microflow päivittää Silo-objektin kentän "IsLocked", jolloin säiliö merkitään lukituksi. Lukitus estää säiliön tilanmuutokset ja mahdolliset lisätoimenpiteet, kunnes se avataan uudelleen. **Näkymän sulkeminen:** Kun säiliö on päivitetty, "Close page" -toiminto sulkee käyttöliittymän muokkausnäytön ja palauttaa käyttäjän takaisin pääsivulle. Päivitetty lukitustila näkyy säiliön käyttöliittymän tilanäkymässä.

4.8 Tulostustoiminnallisuuden toteutus

Tulostustoiminto on olennainen osa sovellusta, sillä sen avulla käyttäjä voi luoda paperiversioita säiliöiden tilannekuvista ja raporteista. Tulostusprosessi perustuu Mendix-alustalla toteutettuun

JavaScript-toimintoon, joka avaa selaimen tulostusikkunan käyttäjän painaessa tulostuspainiketta sovelluksessa. Tämän lisäksi Microflow varmistaa tulostustoiminnon oikean käynnistymisen ja käyttöliittymä mukautuu automaattisesti siten, että tulostettava näkymä sisältää vain olennaiset tiedot.

Tulostusprosessi käynnistyy, kun käyttäjä painaa tulostuspainiketta sovelluksessa (ks. Kuvio 13). Tämä käynnistää Microflow:n (ks. Kuvio 14), jonka ainoa tehtävä on kutsua JavaScript-funktiota `JS_OpenPrintDialog()` (ks. Kuvio 15). Tämä funktio suorittaa selaimen sisäisen `window.print()`-komenton, joka avaa tulostusikkunan ja mahdollistaa käyttöliittymän tulostamisen paperille tai tallentamisen PDF-muotoon (ks. Kuvio 16). Mikäli tulostuksen aikana ilmenee virheitä, funktio kirjaa ne selaimen kehittäjäkonsoliin, jotta mahdolliset ongelmat voidaan debugata.

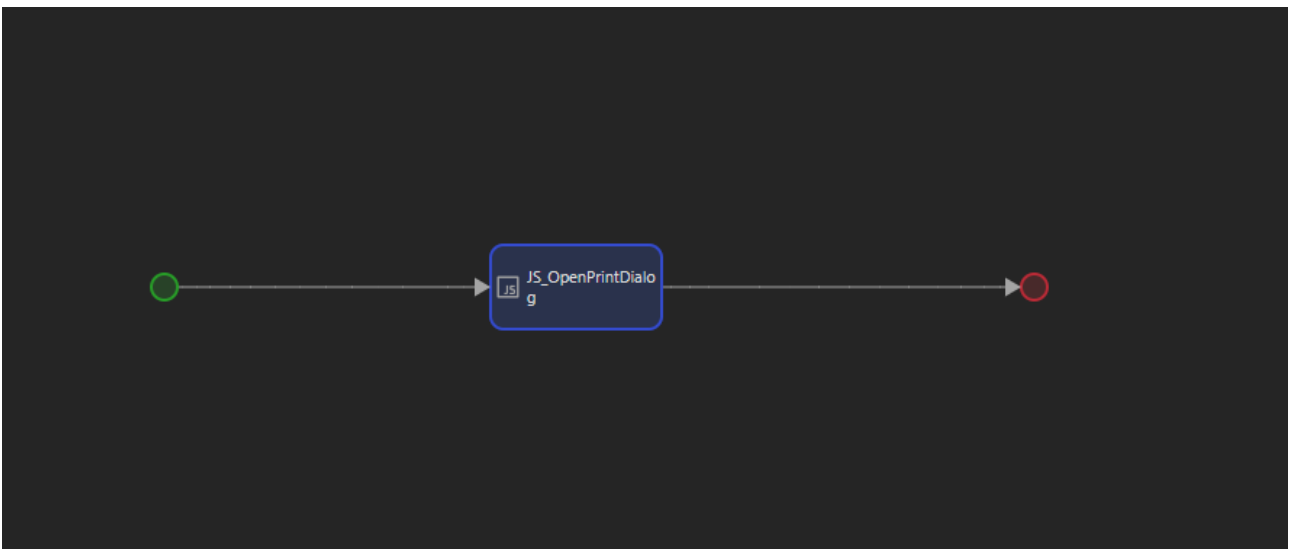
Käyttöliittymä on suunniteltu niin, että tulostusnäkymä mukautuu automaattisesti näyttämään vain tarvittavat elementit ilman ylimääräisiä painikkeita ja navigointivalikoita. Kun käyttäjä avaa tulostusikkunan, hän voi valita tulostettavat sivut, asettelun ja tulostusmuodon. Tulosteen voi tämän jälkeen joko lähettää fyysiselle tulostimelle tai tallentaa PDF-muodossa esimerkiksi arkistointia varten.

Tulostustoiminnon kehityksessä voidaan ottaa huomioon useita parannuksia. Yksi keskeinen optimointimahdollisuus on lisätä erillinen CSS-tyylitiedosto, joka määrittelee, mitkä elementit näkyvät tulosteessa ja mitkä piilotetaan. Tämä voisi parantaa tulosteen selkeyttä ja luettavuutta. Lisäksi tulostustoimintoa voisi laajentaa hyödyntämällä Mendixin sisäistä PDF-generointia, joka mahdollistaisi paremmin muotoiltujen raporttien luomisen suoraan ilman selaimen tulostustoimintoa. Toinen kehitysmahdollisuus on dynaamisen otsikoinnin lisääminen, jolloin tulostettava dokumentti voisi sisältää esimerkiksi tulostusajankohdan ja käyttäjän nimen, mikä helpottaisi tietojen arkistointia ja seuranta.

Tämän kokonaisuuden ansiosta tulostustoiminnallisuus on helppokäyttöinen ja integroitu osaksi sovelluksen käyttöliittymää ilman monimutkaisia konfigurointeja. Mikäli tulostustoimintoa halutaan kehittää edelleen, voidaan harkita automaattisia PDF-raportteja tai tulostusnäkyvän visuaalista optimointia. Tämä varmistaisi, että sovelluksen käyttäjät voivat hyödyntää tulostustoimintoa tehokkaasti eri tilanteissa.



Kuvio 13. Tulostuspainike käyttöliittymän yläreunassa



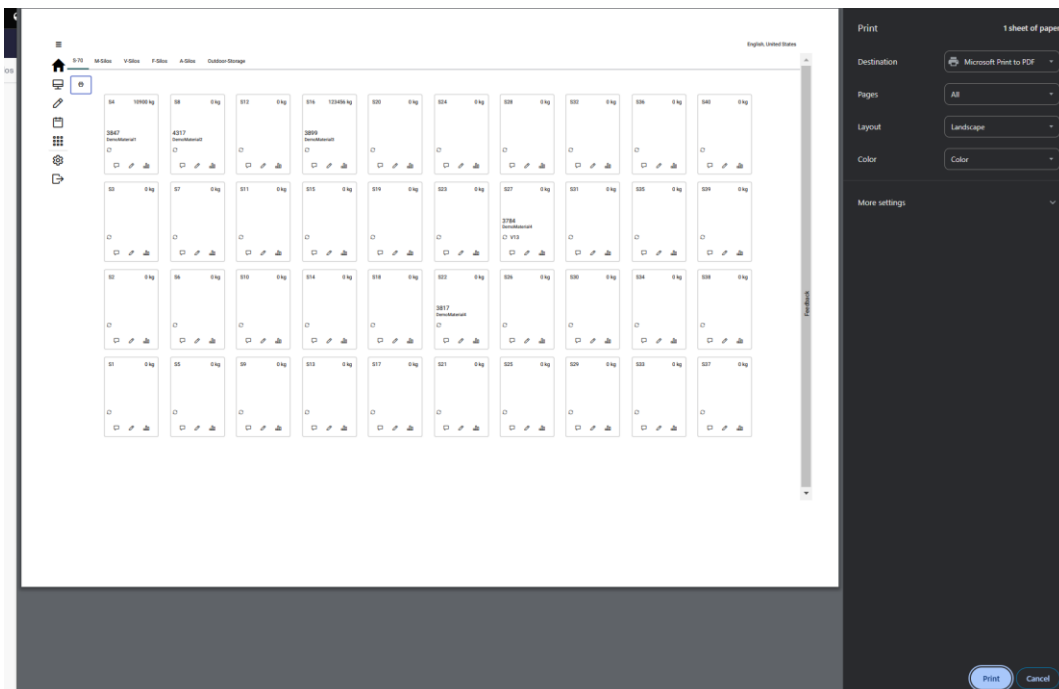
Kuvio 14. Tulostuskomennon avaava microflow

```

Settings Code
1 // This file was generated by Mendix Studio Pro.
2 //
3 // WARNING: Only the following code will be retained when actions are regenerated:
4 // - the import list
5 // - the code between BEGIN USER CODE and END USER CODE
6 // - the code between BEGIN EXTRA CODE and END EXTRA CODE
7 // Other code you write will be lost the next time you deploy the project.
8 import "mx-global";
9 import { Big } from "big.js";
10
11 // BEGIN EXTRA CODE
12 // END EXTRA CODE
13
14 /**
15  * @returns {Promise.<void>}
16  */
17 export async function JS_OpenPrintDialog() {
18     // BEGIN USER CODE
19     try {
20         // Open the browser's print dialog
21         window.print();
22     } catch (error) {
23         // Log any errors that might occur
24         console.error("Error in JS_OpenPrintDialog:", error);
25         throw error;
26     }
27     // END USER CODE
28 }
29

```

Kuvio 15. Tulostusdialogin avaava JavaScript



Kuvio 16. Selaimessa aukeava tulostusdialogi

4.9 Tietomalli ja sen rakenne

Sovelluksen tietomalli (ks. Kuvio 17) on keskeinen osa järjestelmää, sillä se määrittää, miten säiliöihin liittyvät tiedot tallennetaan ja käsitellään. Tietomallin suunnittelussa on pyritty huomioimaan skaalautuvuus ja tehokas tiedonhallinta, jotta se tukee teollisten prosessien hallintaa mahdollisimman hyvin. Tietomalli koostuu useista toisiinsa liittyvistä entiteeteistä, jotka kattavat säiliöiden perustiedot, materiaalit, historiatiedot sekä analytiikkadatan.

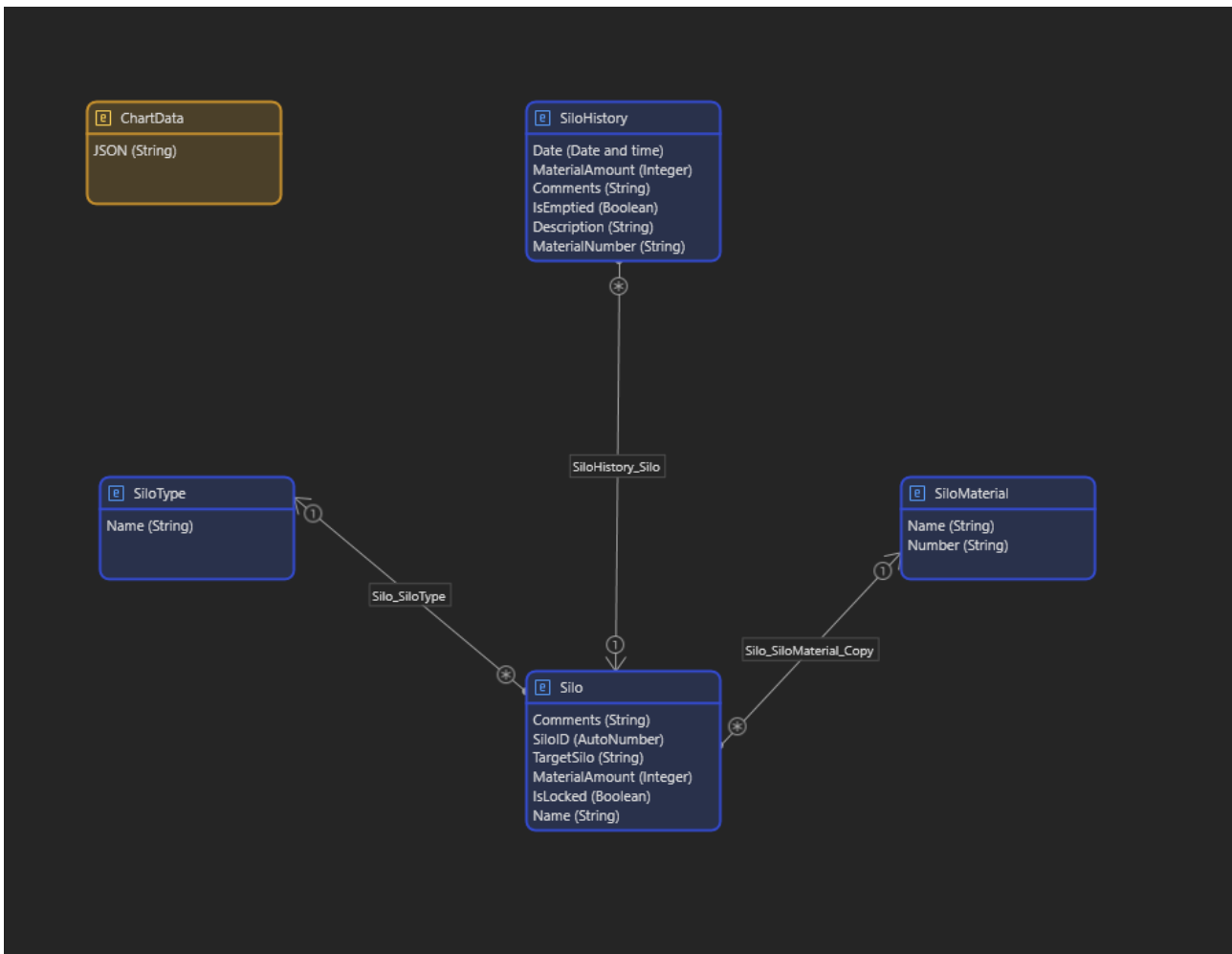
Tietomallin ydin on Silo-entiteetti, joka sisältää kaikki keskeiset tiedot yksittäisestä säiliöstä. Jokaisella säiliöllä on yksilöllinen tunniste (SiloID), jonka avulla järjestelmä voi kohdistaa päivitykset oikeaan kohteeseen. Säiliön tila määritellään useiden attribuuttien avulla, kuten MaterialAmount (materiaalimäärä), Comments (kommentit) ja IsLocked (lukitustila). Lisäksi säiliöihin liittyy tieto kohdesäiliöstä (TargetSilo), mikä mahdollistaa materiaalsiirtojen seurannan.

Historiatietojen hallinta on toteutettu SiloHistory-entiteetin avulla, johon tallennetaan säiliön aikaisemmat tilamuutokset, materiaalmäärät, kommentit ja tyhjennyshistoria. Jokainen muutos luo uuden historiatietueen, mikä mahdollistaa ajantasaisen seurannan ja analyysin säiliön käytöstä.

Säiliöt voidaan luokitella eri tyyppeihin SiloType-entiteetin avulla, mikä mahdollistaa erilaisten säiliöiden hallinnan esimerkiksi materiaalin, käyttötarkoituksen tai sijainnin perusteella. Säiliöissä säilytettävät materiaalit puolestaan määritellään SiloMaterial-entiteetissä, jossa jokaisella materiaalilla on oma tunnisteensa ja nimensä. Tämä rakenne tukee materiaalinhallinnan integraatioita ja jäljitettävyyttä.

Järjestelmään sisältyy myös ChartData-entiteetti, joka tallentaa JSON-muodossa tiedot, joita käytetään graafien ja analytiikan muodostamiseen. Tämä mahdollistaa tietojen tehokkaan esittämisen visuaalisessa muodossa ilman monimutkaisia tietokantakyselyitä reaaliajassa.

Kokonaisuutena tietomalli on suunniteltu siten, että se tukee teollisten säiliöiden hallintaa, historiadatan seurantaan sekä analytiikkaa, samalla mahdollistaen joustavan laajennettavuuden tulevaisuuden kehitystarpeisiin. Seuraavassa kuvassa esitellään tietomallin rakenne ja sen eri entiteettien väliset suhteet.



Kuvio 17. Sovelluksen tietomalli

4.10 Keskeiset havainnot alustan soveltuvuudesta

Tutkimuksen seurauksena voidaan tarkastella, kuinka hyvin Mendix-alusta soveltuu teollisten prosessien hallintaan perustuen kehityksen nopeuteen, käyttöliittymän suunnitteluun, tietomallin hallintaan, integraatioihin ja suorituskykyyn. Näiden näkökulmien avulla voidaan arvioida alustan soveltuvuutta eri käyttötarkoituksiin sekä tunnistaa sen vahvuudet ja mahdolliset rajoitteet.

4.10.1 Kehityksen nopeus ja joustavuus

Mendix-alustan suurimpia vahvuuksia on sen nopea kehityssykli, joka mahdollistaa sovellusten kehittämisen ilman laajaa ohjelmointiosaamista. Sovelluksen kehittämisessä havaittiin, että visuaalinen mallintaminen ja komponenttipohjainen lähestymistapa nopeuttivat kehitystä merkittävästi verrattuna perinteisiin ohjelmointimenetelmiin.

Testisovelluksen toteutuksen aikana oli nähtävissä, että prototyyppien luominen ja testaaminen oli huomattavasti nopeampaa kuin perinteisillä kehitysmenetelmillä. Tämä teki alustasta erityisen soveltuvan tilanteisiin, joissa tarvitaan nopeita muutoksia ja itseään toistavaa kehitystä. Toisaalta havaittiin, että monimutkaisemmat ja mukautetummat toiminnot edellyttivät Java- tai JavaScript-laajennuksia, mikä voi hidastaa kehitystä ja lisätä ylläpitotyötä.

4.10.2 Käyttöliittymän suunnittelu ja käyttäjäkokemus

Mendixin tarjoama käyttöliittymän rakennustyökalu mahdollisti nopean UI-komponenttien käyttöönoton ja mukauttamisen ilman perinteistä front-end-kehitystä. Testisovelluksen päänäkymä säiliöiden hallinnalle suunniteltiin drag-and-drop-menetelmällä, mikä mahdollisti responsiivisen käyttöliittymän ilman erillistä CSS-muotoilua. Käyttöliittymä oli kuitenkin joissakin tapauksissa rajoittunut Mendixin valmiisiin komponentteihin, mikä tarkoitti, että laajemmin räätälöidyt elementit oli toteutettava erillisinä widgeteinä tai muokkaamalla Mendixin valmiita UI-komponentteja. Tämä voi olla haaste, jos sovellus vaatii yksityiskohtaisia käyttöliittymän muokkauksia.

4.10.3 Tietomalli ja sen hallinta

Sovelluksen tietomalli oli helppo rakentaa Mendixin visuaalisen tietomallinnustyökalun avulla. Sovelluksen säiliöhistorian tallennus ja hakutoiminnot toteutettiin nopeasti microflow-logiikoilla ilman monimutkaisia SQL-kyselyitä. Tämä teki tietokantatoiminnoista tehokkaita ja helppokäyttöisiä myös kehittäjille, joilla ei ole vahvaa tietokantaosaamista. Haasteeksi nousi kuitenkin suurten tietomäärien käsittely, sillä Mendix suorittaa tietokantaoperaatiot taustalla omalla prosessillaan, mikä voi aiheuttaa suorituskykyhaasteita verrattuna suoraan tietokantakyselyyn, jos tietomäärät kasvavat merkittävästi. Tämä tarkoittaa, että isojen datasettien tehokas käsittely vaatii optimoituja hakukyselyitä ja mahdollisesti erillisiä tietovarastoja (Martinez & Pfister 2023).

4.10.4 Integraatiomahdollisuudet ja rajapinnat

Mendix-alustan vahvuutena havaittiin olevan sen laajat integraatiomahdollisuudet, erityisesti REST- ja SOAP-rajapinnat, joiden avulla sovellus voitaisiin yhdistää muihin teollisuuden järjestelmiin, kuten MES-, SCADA- ja ERP-järjestelmiin (Top 4 Ways to Integrate SCADA and MES to ERP

Systems 2020). Sovellus tukee natiivisti IoT-laitteiden ja ulkoisten tietokantojen yhdistämistä (Security Overview 2023), mikä tekee siitä varteenotettavan vaihtoehdon teollisen datan reaaliaikaiseen käsittelyyn.

4.10.5 Suorituskyky ja skaalautuvuus

Sovelluksen suorituskyky havaittiin riittäväksi pienille ja keskisuurille järjestelmille. Vaikka alusta oli tehokas pienimuotoisessa projektissa, sen suorituskyky voi heikentyä merkittävästi, kun käsitellään suuria tietomääriä, tämä vaatisi optimointia tai lisätyökalujen käyttöä suorituskyvyn parantamiseksi (Martinez & Pfister 2023). Mendix käyttää automaattista versionhallintaa ja DevOps-tukea, mikä helpottaa järjestelmän jatkuvaa kehittämistä ja päivityksiä.

Pilvipohjaiset käyttöönnotot (Mendix Cloud, AWS, Azure) tarjoavat hyvän skaalautuvuuden, mutta on-premises-asennuksissa saattaa tulla haasteita suorituskyvyn optimoinnissa. Suurten tietomäärien käsittelyssä voi esiintyä viiveitä. Tämä tarkoittaa, että yritysten, joilla on suuret tietomäärät ja korkeat suorituskykyvaatimukset, tulisi harkita tietokantakerroksen erillistä optimointia. Demonstraatiosoveluksessa pilvipohjaisia metodeja ei käytetty, mutta ne voisivat olla hyödyllisiä (Martinez & Pfister 2023).

5 Tulosten tarkastelu ja Pohdinta

Lopuksi voidaan analysoida tutkimuksen tuloksia suhteessa asetettuihin tutkimuskysymyksiin sekä tarkastellaan tutkimuksen rajoituksia ja luotettavuutta. Lisäksi esitetään mahdollisia jatkokehitysehdotuksia ja arvioidaan Mendix-alustan soveltuvuutta laajemmin tulevaisuuden kehitystarpeiden näkökulmasta.

5.1 Tulosten tarkastelu suhteessa tutkimuskysymyksiin

Tutkimuksen tavoitteena oli arvioida Mendix-alustan soveltuvuutta teollisten prosessien hallintaan ja vastata seuraaviin tutkimuskysymyksiin:

Kuinka hyvin Mendix-alusta soveltuu teollisten prosessien hallintaan?

Mitä etuja ja haasteita low-code-alustat tuovat perinteisiin ohjelmistokehitysmenetelmiin verrattuna?

Millaisia integraatio- ja skaalautuvuusmahdollisuuksia Mendix tarjoaa teollisuusympäristössä?

Mendix soveltuu erityisesti pienille ja keskikokoisille teollisille sovelluksille, joissa vaaditaan nopeaa kehitystä. Sovelluksen toteutuksessa havaittiin, että low-code-ympäristö mahdollisti nopean kehityksen ja käyttöliittymän rakentamisen ilman perinteistä ohjelmointia. Monimutkaisemmissa skenaarioissa, joissa tarvitaan erityisiä liiketoimintasääntöjä ja raskasta laskentaa, Mendix saattaa kuitenkin vaatia lisäkehitystä Java- tai JavaScript-laajennusten avulla.

Toiseen tutkimuskysymykseen liittyen havaittiin, että low-code-kehityksen suurimmat edut olivat nopeus, joustavuus ja helpompi ylläpidettävyys verrattuna perinteiseen ohjelmistokehitykseen. Testisovelluksessa käytettiin Mendixin visuaalista kehitystyökalua, joka mahdollisti nopean prototyyppien rakentamisen ja käyttöönoton. Tämä osoittaa, että Mendix voi merkittävästi nopeuttaa ohjelmistokehitystä ja tehdä siitä saavutettavampaa myös käyttäjille, joiden tekninen taitotaso ei ole korkea. Haasteiksi havaittiin kuitenkin käyttöliittymäkomponenttien rajallisuus ja tietokantaky-selyiden optimoinnin ongelmat suurissa tietomäärissä.

Kolmanteen tutkimuskysymykseen liittyen havaittiin, että Mendix tarjoaa hyvät integraatiomahdollisuudet REST- ja SOAP-rajapintojen kautta, mikä mahdollistaa yhteydenoton teollisuuden MES-, SCADA- ja ERP-järjestelmiin. Koska testisovellusta ei kuitenkaan yhdistetty ulkoisiin järjestelmiin, nämä havainnot perustuvat teoreettiseen tarkasteluun, työkokemukseen ja Mendixin dokumentaatioon. Skaalautuvuuden osalta Mendix on toimiva vaihtoehto pilvipohjaisille sovelluksille, mutta on-premises-asennuksissa voi olla tarpeen tehdä lisäoptimointeja.

5.2 Tutkimuksen rajoitukset ja luotettavuus

Tämä tutkimus perustui Mendix-alustan testaamiseen yksittäisessä sovelluksessa ilman todellista integraatiota muihin järjestelmiin. Näin ollen tulokset antavat realistisen kuvan alustan perusominaisuuksista, mutta käytännön skaalautuvuutta ja suorituskykyä suurilla tietomäärillä ei testattu varsinaisessa tuotantoympäristössä.

Toinen rajoitus liittyy käyttäjäkokemukseen ja pitkän aikavälin ylläpidettävyyteen. Koska sovellus kehitettiin lyhyessä ajassa eikä sitä testattu laajassa tuotantokäytössä, käyttöliittymän ja tietokantamallin tehokkuuden pitkän aikavälin vaikutuksia ei voitu arvioida kattavasti. Tämä tarkoittaa, että jatkotutkimuksissa olisi hyödyllistä tarkastella Mendix-alustan soveltuvuutta pitkällä aikavälillä oikeassa tuotantoympäristössä.

Luotettavuuden kannalta tutkimus nojasi sekä käytännön testaukseen että Mendixin dokumentaatioon, mikä antaa tasapainoisen kuvan alustan ominaisuuksista. Tulokset ovat hyödyllisiä erityisesti organisaatioille, jotka harkitsevat Mendixin käyttöönottoa osana teollisten prosessien digitalisointia.

5.3 Jatkokehitysehdotukset ja tulevaisuuden näkymät

Tutkimuksen perusteella voidaan suositella ainakin seuraavia kehitysehdotuksia ja jatkotutkimusaiheita:

Integraatiot ja järjestelmäyhteydet, koska testisovellusta ei yhdistetty muihin järjestelmiin seuraava vaihe voisi olla integrointi esimerkiksi MES- tai SCADA-järjestelmään ja sen vaikutuksen analysointi. Lisäksi voitaisiin tutkia IoT-laitteiden yhdistämistä Mendix-sovellukseen, jolloin voitaisiin saada reaaliaikaista dataa teollisuuden laitteista tai sensoreista.

Suorituskyvyn optimointi, jatkossa voitaisiin testata suurten tietomäärien käsittelyä ja arvioida Mendix-sovelluksen skaalautuvuutta laajemmassa tuotantoympäristössä. Eritoten tietokantahakujen ja analytiikkatietojen tehokkuutta voitaisiin tutkia, jotta nähdään järjestelmän toimintaa suurilla datamäärillä.

Low-code vs. perinteinen kehitys, Olisi hyödyllistä vertailla Mendix-alustalla kehitettyä sovellusta perinteisellä ohjelmistokehityksellä toteutettuun ratkaisuun, jotta voidaan selvittää tarkemmin kehitysnopeuden ja ylläpidon eroavaisuuksia (Owen & Tozzi 2025).

6 Johtopäätökset ja suositukset

Päätteeksi tiivistetään tämän opinnäytetyön keskeiset löydökset ja arvioidaan niiden merkitys teollisten prosessien hallinnan näkökulmasta. Lisäksi esitetään suosituksia siitä, miten Mendix-alustaa voidaan hyödyntää teollisessa ympäristössä tehokkaasti ja millaisiin käyttötapauksiin se soveltuu parhaiten.

6.1 Työn keskeiset löydökset ja niiden merkitys

Tutkimuksen ja testaamisen perusteella havaittiin, että Mendix-alusta tarjoaa nopean ja joustavan ratkaisun teollisten prosessien hallintaan, erityisesti tilanteissa, joissa sovellusten kehitystä täytyy mukauttaa ja kehittää ilman laajaa ohjelmointiosaamista. Testisovelluksen toteutus osoitti, että low-code-kehityksen suurimmat edut ovat nopeus, helppokäyttöisyys ja mahdollisuus kehittää visuaalisia käyttöliittymiä ilman erillistä front-end-kehitystä.

Yksi löydöksistä oli myös se, että Mendix tarjoaa laajat integraatiomahdollisuudet REST- ja SOAP-rajapintojen avulla, mikä mahdollistaa sen liittämisen muihin teollisiin järjestelmiin, kuten MES- ja SCADA. Koska testisovellusta ei yhdistetty ulkoisiin järjestelmiin, tämä havainto perustuu täysin Mendixin dokumentaatioon, alustaan liittyvään tarkasteluun ja omakohtaiseen työkokemukseen.

Toinen keskeinen havainto oli suorituskyvyn ja skaalautuvuuden merkitys suurissa tietomäärissä. Mendixin tietokantarakenne helpottaa kehitystyötä, mutta suurten datasettien käsittelyssä voi esiintyä suorituskykyhaasteita, mikäli tietomallia ja hakukyselyitä ei optimoida. Tämä tarkoittaa, että vaikka Mendix on tehokas työkalu pienten ja keskisuurten sovellusten kehittämiseen, se ei välttämättä ole järkevin ratkaisu erittäin suurille ja monimutkaisille järjestelmille ilman lisäoptimointeja.

Käyttöliittymän osalta Mendix todettiin helppokäyttöiseksi ja visuaalisesti joustavaksi, mutta joissakin tapauksissa valmiiden komponenttien rajoitteet saattoivat estää täysin räätälöityjen näkymien toteuttamisen ilman lisäkehitystä. Tämä tarkoittaa, että Mendix soveltuu erityisesti yrityksille, jotka haluavat nopean ratkaisun, mutta joilla ei ole tarvetta erityisen yksilölliselle käyttöliittymälle.

6.2 Suositukset Mendix-alustan hyödyntämiseksi teollisessa ympäristössä

Nopeat prototyyppisovellukset, Mendix soveltuu teollisuuden sisäisten työkalujen ja prototyyppi-sovellusten kehittämiseen, joissa tarvitaan nopeita muutoksia, sekä yrityksille, jotka haluavat testata uusia järjestelmiä ennen laajamittaista käyttöönottoa. Mendix voisi tarjota kustannustehokkaan ja ketterän ratkaisun (What is Low-Code Development? 2022).

Integraatio olemassa oleviin järjestelmiin, Mendixin REST- ja SOAP-rajapinnat tekevät siitä hyvän lisäosan esimerkiksi MES-, SCADA- ja ERP-järjestelmien rinnalle. Esimerkiksi säiliöiden täyttöasteen seuranta voidaan toteuttaa Mendixillä ja yhdistää olemassa olevaan tuotannonhallintajärjestelmään.

Pienet ja keskiuuret sovellukset, joissa tarvitaan visuaalista käyttöliittymää. Mendix on erityisen hyödyllinen nimenomaan pienissä ja keskiuurissa teollisuussovelluksissa, joissa perinteinen ohjelmistokehitys olisi liian kallista tai aikaa vievää. Eritoten käyttäjäystävällisten käyttöliittymien luominen ilman erillistä front-end-kehitystä on Mendixin vahvuus.

Pilvipohjaiset käyttöönotot ja suositeltavia, koska Mendix on optimoitu pilviympäristöihin (Mendix Cloud, AWS, Azure), pilvipohjainen käyttöönotto voi tarjota paremman skaalautuvuuden ja suorituskyvyn. Yritysten, joilla on korkeat tietoturva-vaatimukset, kannattaa arvioida Mendixin soveltuvuutta tapauskohtaiseksi.

Lähteet

A practical take on low-code vs. traditional development 2025 Owen, G & Tozzi, C Viitattu 14.2.2025 Saatavilla: <https://www.techtarget.com/searchsoftwarequality/tip/A-practical-take-on-low-code-vs-traditional-development>

Digitalisaation vaikutukset teollisuudessa 2022 Viitanen, A Diplomityö, Tampereen yliopisto Viitattu 14.2.2025 Saatavilla: <https://trepo.tuni.fi/bitstream/10024/142294/2/ViitanenAnnika.pdf>

Forrester 2025 Low-Code Unlocks Agile and Integrated Industrial Manufacturing Viitattu 14.2.2025 Saatavilla: [https://www.mendix.com/wp-content/uploads/Low-Code Unlocks Agile and Integrated Industrial Manufacturing.pdf](https://www.mendix.com/wp-content/uploads/Low-Code_Unlocks_Agile_and_Integrated_Industrial_Manufacturing.pdf)

Gartner forecast on low-code development technologies 2023 Tooljet Viitattu 14.2.2025 Saatavilla: <https://blog.tooljet.ai/gartner-forecast-on-low-code-development-technologies/>

iFIX HMI / SCADA 2023 Novotek Finland Viitattu 14.2.2025 Saatavilla: <https://www.novotek.com/fi/ratkaisut-ja-tuotteet/hmi-scada/ifix-hmi-scada>

Low-code-pohjaisen tehdasapplikaatiokonseptin kehitysprosessi 2023 Korkko, J Diplomityö, Tampereen yliopisto Viitattu 17.3.2025 Saatavilla: <https://trepo.tuni.fi/handle/10024/147161>

Security Overview 2023 Mendix Viitattu 14.2.2025 Saatavilla: <https://docs.mendix.com/refguide/security-overview/>

TEPA-termipankki Viitattu 14.4.2025 Saatavilla: <https://termipankki.fi/tepa/fi/>

Top 4 Ways to Integrate SCADA and MES to ERP Systems 2020 Vertech Viitattu 14.2.2025 Saatavilla: <https://www.verttech.com/blog/4-ways-to-integrate-scada-and-mes-to-erp-systems>

What is Low-Code Development? 2022 Mendix Viitattu 14.2.2025 Saatavilla: <https://www.mendix.com/low-code-guide>