



Ohjelmistorobotiikalla tehoa valmistavaan pk-teollisuuteen -hanke

Juha Yli-Hemminki

Siemens Mendix -kehitysympäristön kokeilu

Tämän kokeilun tarkoituksena oli selvittää Siemens Mendix Studio Pro -kehitysympäristön soveltuvuutta pienen verkkoselaimen kautta toimivan poikkeustenkirjausjärjestelmän kehittämiseen ja miten kehitysympäristö sopii pk-yrityksen tarpeisiin. Mendix-kehitysympäristö on low-code-tyyppinen ympäristö, jonka avulla sovelluksia voidaan kehittää käyttämättä varsinaisia ohjelmointikieliä. Tarkoituksena on helpottaa sovelluskehitystä ja mahdollistaa se myös sellaisille toimijoille, joilla ei ole ohjelmointitaitoja. Tämä madaltaa kynnystä yrityksen omalle sovelluskehitykselle.

Yritykset voivat kehitysympäristön avulla toteuttaa haluamansa sovelluksen ilman ulkopuolista apua ja toimia näin itsenäisemmin. Lisäksi yritys voi mahdollisesti saada toteutettua paremmin omaan toimintaansa sopivan ja oman määrittelynsä mukaisen sovelluksen.

Kehitysympäristö yhdistää kaikki sovelluskehityksessä tarvittavat työkalut saman käyttöliittymän alle. Näitä ovat esimerkiksi versionhallinta, testaus, tiimityöskentely, sovelluksen julkaisu jne. Tarkoituksena on siis se, että kehitystyökalun valitseva yritys saa toteutettua kaikki kehityksen vaiheet yhden ja saman työkalun avulla. Yritykseltä ei näin ollen vaadita asiantuntemusta eri työkalujen valitsemiseen ohjelmistoprosessin eri vaiheisiin, vaan suunnittelu, määrittely, toteutus, testaus ja julkaisu, hoituvat yhden työkalun avulla.

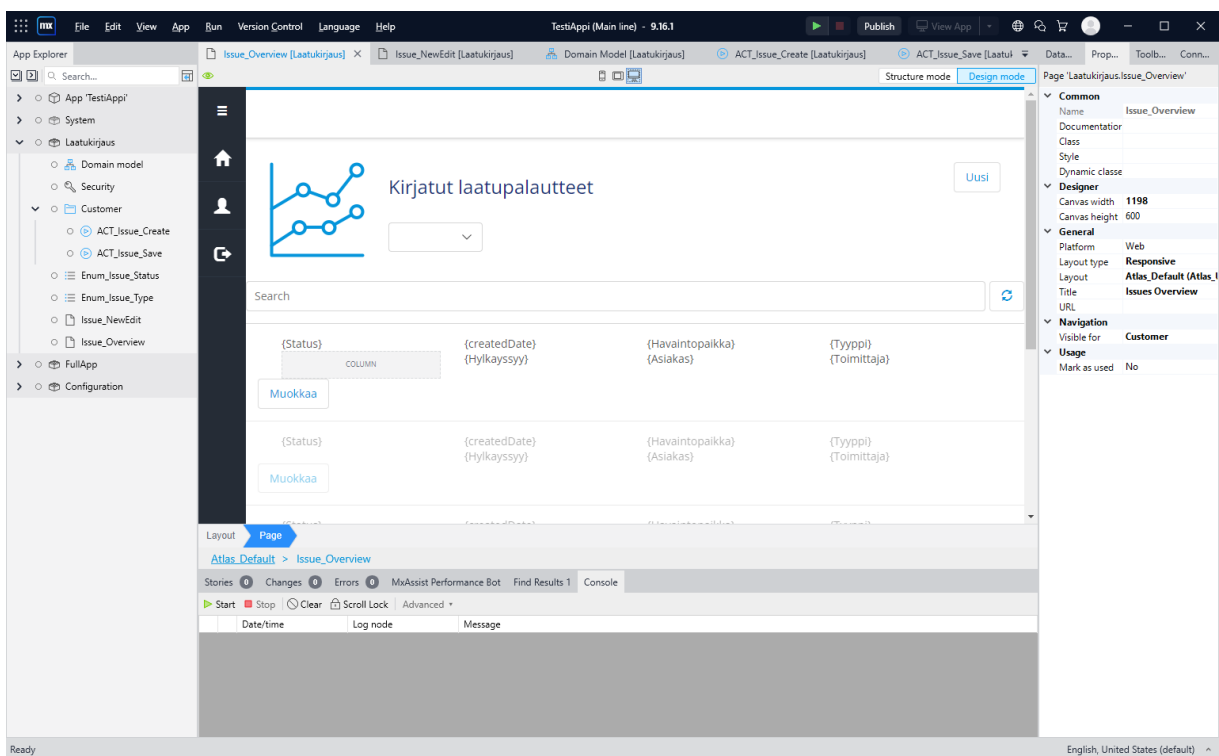
Kehitysympäristön perusrakenne

Kehitysympäristöstä löytyvät omat osiot sovelluksen käyttöliittymän näkymien (page) tekoa varten, sovelluksen tietomallin (domain model) määrittelyä varten sekä tiedon käsittelemiseksi käyttöliittymän ja tietomallin välillä (microflow).



Käyttöliittymänäkymiä voidaan suunnitella ja kehittää page-osioissa sillä periaatteella, että muokkauksia tehdessä nähdään koko ajan miltä lopullinen käyttöliittymä tulee näyttämään. Käyttöliittymän objektit näkyvät suunnittelunäkymässä sellaisena kuin ne tulevat näyttämään lopullisessa sovelluksessa ja niiden asettelua voidaan muuttaa hiirellä raahaamalla.

Lisäksi tässä osiossa voidaan tarkastella miltä kehitettävän sovelluksen käyttöliittymä näyttää mobiili- tai tablet-laiteelle mukautettuna. Käytössä sovellus mukautuu automaattisesti erityyppisille päätelaitteille. Kuvassa 1 nähdään käyttöliittymän suunnittelunäkymä työpöytäsovellukselle.



Kuva 1. Käyttöliittymän suunnittelunäkymä.

Käyttöliittymän suunnittelunäkymässä voidaan käyttöliittymän objektien asetuksissa asettaa suoraan toiminnot, jotka käyttöliittymäobjektia klikattaessa suoritetaan. Kuvassa 2 näkyvät käyttöliittymän Muokkaa-painikkeen asetukset ja painikkeelle valittu toiminto eli klikattaessa näytetään uuden poikkeaman kirjausnäkymä.



Kuva 2. Käyttöliittymän painikkeen asetukset.

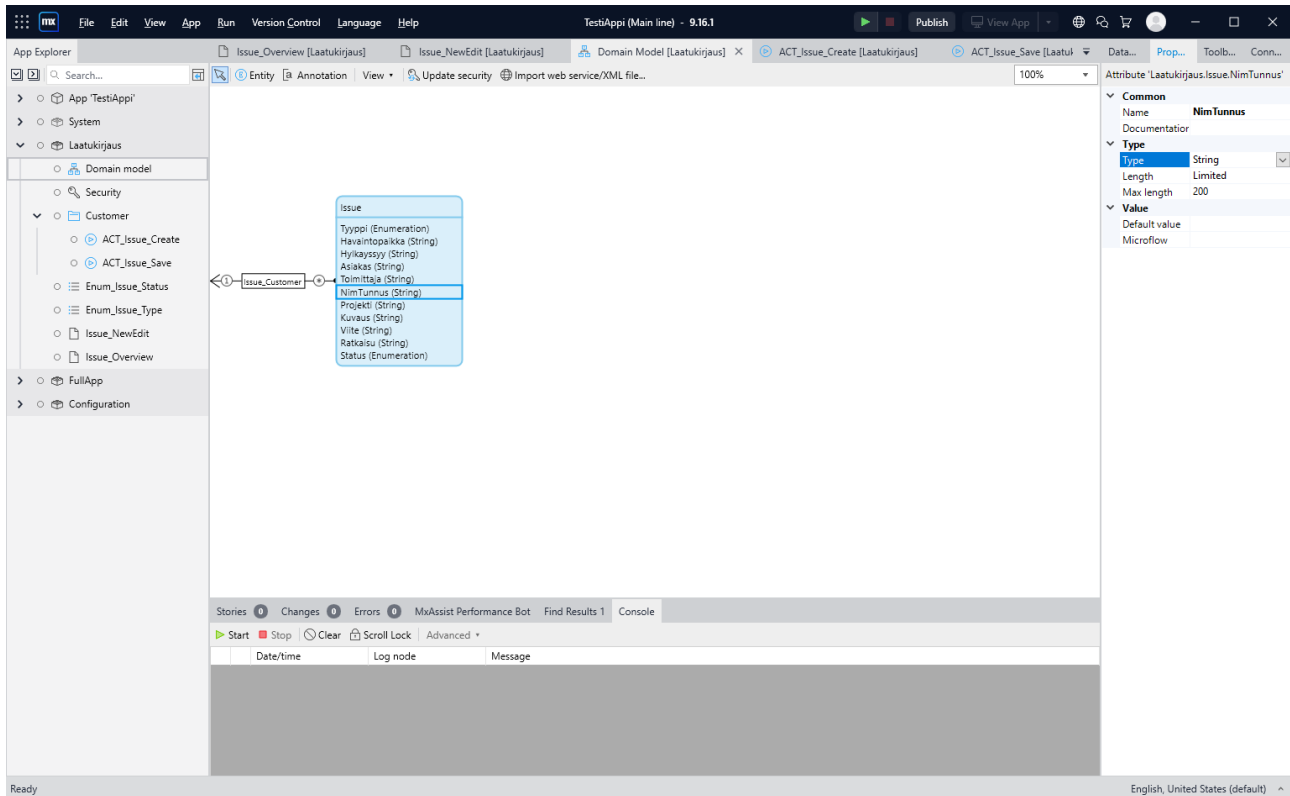
Tietomallin määrittelyosiossa määritetään mitä tietoja sovelluksessa pidetään tallessa eli määritetään mitä muuttujia sovelluksella on käytössä. Kullekin määritetylle tiedolle voidaan asetuksista määrittää, onko tieto sovelluksen suorittamisen aikana laskettu arvo (calculated) vai onko tieto tarkoitus lukea/tallentaa tietokantaan (stored). Tietokannan määrittäminen tapahtuu kehitysympäristön puolesta automaattisesti eikä sovelluskehittäjän tarvitse puuttua siihen lainkaan.

Tietomallin yhtä tietokokoelmaa kutsutaan entiteetiksi (entity) ja entiteetille voidaan määritellä yhteyksiä (association), joiden kautta voidaan rakentaa tietokokoelmaan liittyviä toimintoja tai kytkeä se toisiin entiteetteihin. Näiden toimintojen kautta päästää määrittämään sovelluksen toimintalogiikkaa.

Kullekin tietomallissa määritellylle muuttujalle voidaan normaaliin tapaan antaa tiedon yksilöivä nimi ja valita tiedon tyyppi kuten teksti (merkkijono), kokonaisluku, desimaaliluku, luettelo (enumeraatio) jne. Lisäksi muuttujille voidaan antaa muita määreitä kuten muuttujan oletusarvo ja tekstitiedon merkkien lukumäärä. Kuvassa 3 on esitetty esimerkkisovelluksen

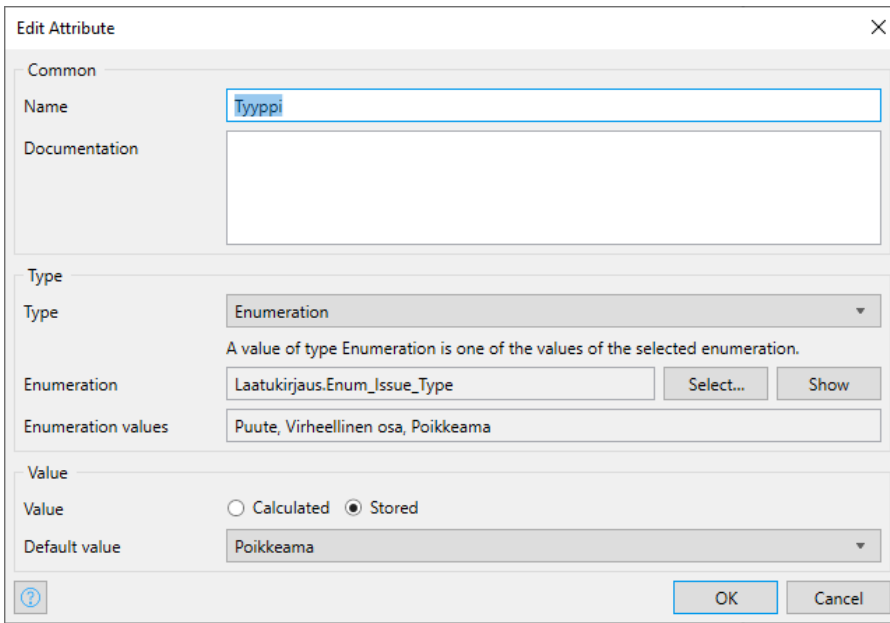


tietomalli. Tässä tapauksessa tietomalli sisältää yhden tietokokoelman, joka puolestaan sisältää yksittäiselle poikkeamalle tallennettavat tiedot. Tietokokoelma on yhdistetty assosiaatiolla sovelluksen käyttäjiin siten, että yhdellä käyttäjällä voi olla kirjattuna useita poikkeamia ja jokaisella poikkeamalla on täsmälleen yksi kirjauksen tehnyt käyttäjä.



Kuva 3. Esimerkkisovelluksen tietomalli.

Yksittäisen muuttujan tietoja päästään muokkaamaan myös oman muokkausikkunan kautta, josta esimerkki kuvassa 4. Tässä päästään valitsemaan muuttujan tyyppi, onko muuttuja laskennallinen arvo vai tietokantaan tallennettava sekä muuttujan oletusarvo alkutilanteessa.

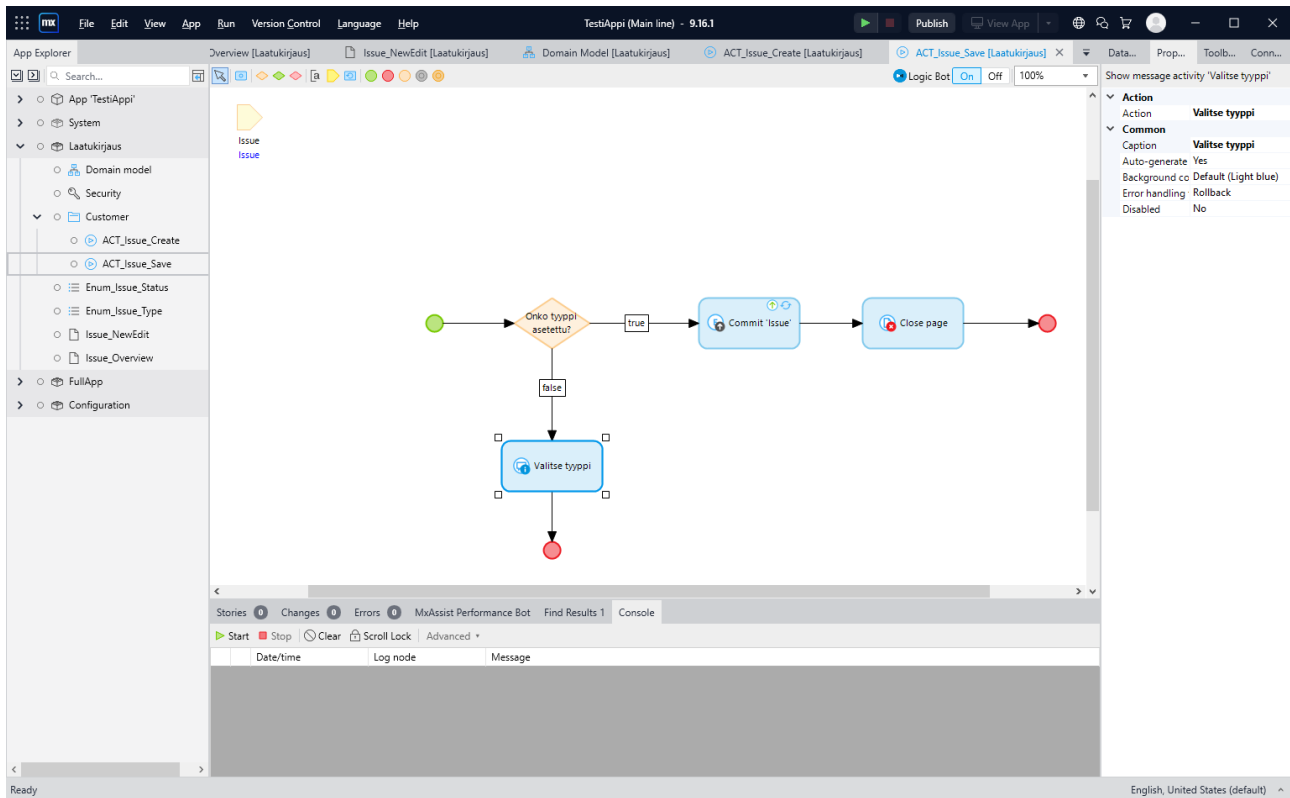


Kuva 4. Muuttujan tietojen muokkaus.

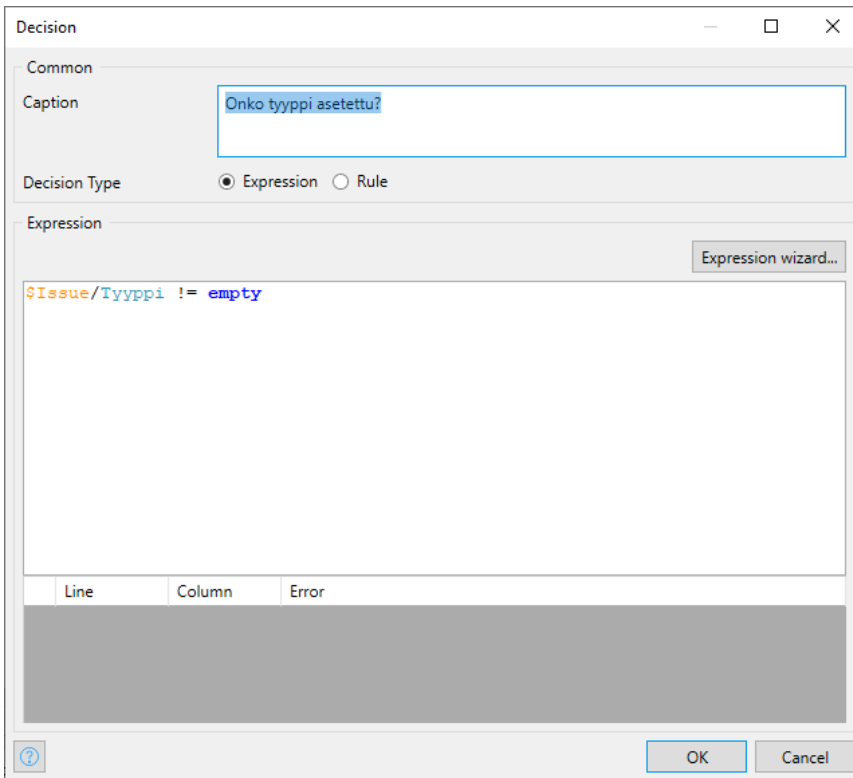
Kolmannessa osiossa kehitettävän sovelluksen toimintojen määrittämiselle, määritellään sovellukseen liittyvät MicroFlow:t. Näiden tarkoituksena on tarkentaa sovelluksen toimintalogiikkaa. MicroFlow:t mahdollistavat muun muassa käyttöliittymän kautta syötettyjen tietojen tarkistamisen ja sovelluksen toiminnan muuttamisen tarkistuksen tuloksen mukaisesti.

MicroFlow:ien avulla saadaan toteutettua ohjelmoinnin perinteiset ehtorakenteet ja haarautettua toimintaa muuttujan arvon tai tyyppin perusteella. Rakenteet tarjoavat myös mahdollisuuden suorittaa toimintoja ja esimerkiksi avata ja sulkea näkymiä ehtolauseissa suoritettujen vertailujen perusteella.

Kuvassa 5 näkyvä Microflow tarkistaa uudelle poikkeamalle syötetyistä tiedoista, onko uudelle poikkeamalle asetettu tyyppi. Tämän vertailun perusteella jatketaan joko poikkeaman tallentamiseen ja poikkeaman syöttönäkymän sulkemiseen tai avataan käyttöliittymään näkymä, jossa pyydetään käyttäjää valitsemaan puuttuva tyyppi. Kuvassa näkyvä vinoneliö edustaa ehtoa, jonka perusteella toiminta haarautuu. Haluttu ehto kirjoitetaan kyseisen päätösobjektiin asetuksiin skiptinä, josta esimerkki kuvassa 6.



Kuva 5. Microflow käyttöliittymäsoyöteen tarkistamiseksi.



Decision

Common

Caption: Onko tyyppi asetettu?

Decision Type: ☒ Expression ☐ Rule

Expression: `$Issue/Tyyppi != empty`

Expression wizard...

Line	Column	Error
------	--------	-------

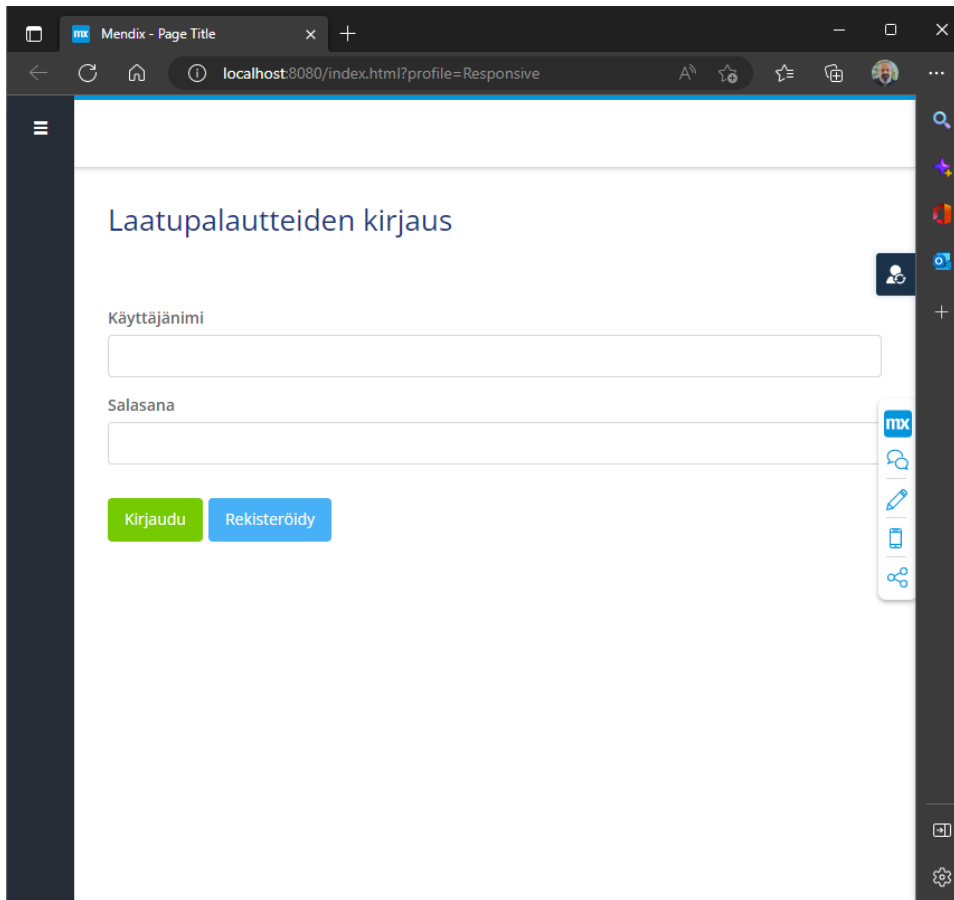
OK Cancel

Kuva 6. Haarautumisen ehtolause.

Mendix-sovelluksiin voidaan itse määritettävien näkymien lisäksi liittää valmiita osia, joissa on toteutettuna useimmille sovelluksille yhteisiä toiminnallisuuksia, kuten käyttäjien hallinta ja sisäänkirjautuminen. Komponentteja voidaan ottaa käyttöön kehitysympäristöön sisältyvän Marketplace-osion kautta. Käyttämällä omassa sovelluksessa valmiita komponentteja, saadaan hyödynnettyä valmiiksi testattuja ja toimivaksi todettuja osia. Valmiita toiminnallisuuksia käyttämällä oman sovelluksen laatu paranee ja kehitystyö nopeutuu.

Esimerkkisovelluksen toiminta

Esimerkkisovelluksena toteutettiin yksinkertainen verkkosivun kautta käytettävä laatupalautteiden kirjausjärjestelmä. Sovelluksessa on käytetty kehitysympäristön tarjoamaa valmista komponenttia, joka sisältää tietoturvallisen käyttäjäprofiilien hallinnan ja kirjautumistoiminnon sovelluksen käytön suojaamiseksi. Kuvassa 7 sovelluksen kirjautumis- ja rekisteröitymisnäky.



Kuva 7. Sovelluksen kirjautumisenäkymä.

Kirjautumisenäkymästä päästään tarkastelemaan jo kirjattujen laatupoikkeamien perustietoja, lisäämään uusia ja muokkaamaan jo olemassa olevien palautteiden tietoja. Uudet kirjatut palautteet lisätään listan jatkoksi ja kulloinkin listassa näkyviä palautteita voidaan suodattaa palautteen tilan perusteella (Avoinna, käsittelyssä, jne.). Kuvassa 8 esimerkinäkymä sovelluksen toiminnasta, joka vastaa kuvan 1 suunnittelunäkymää kehitystyökalun puolella.

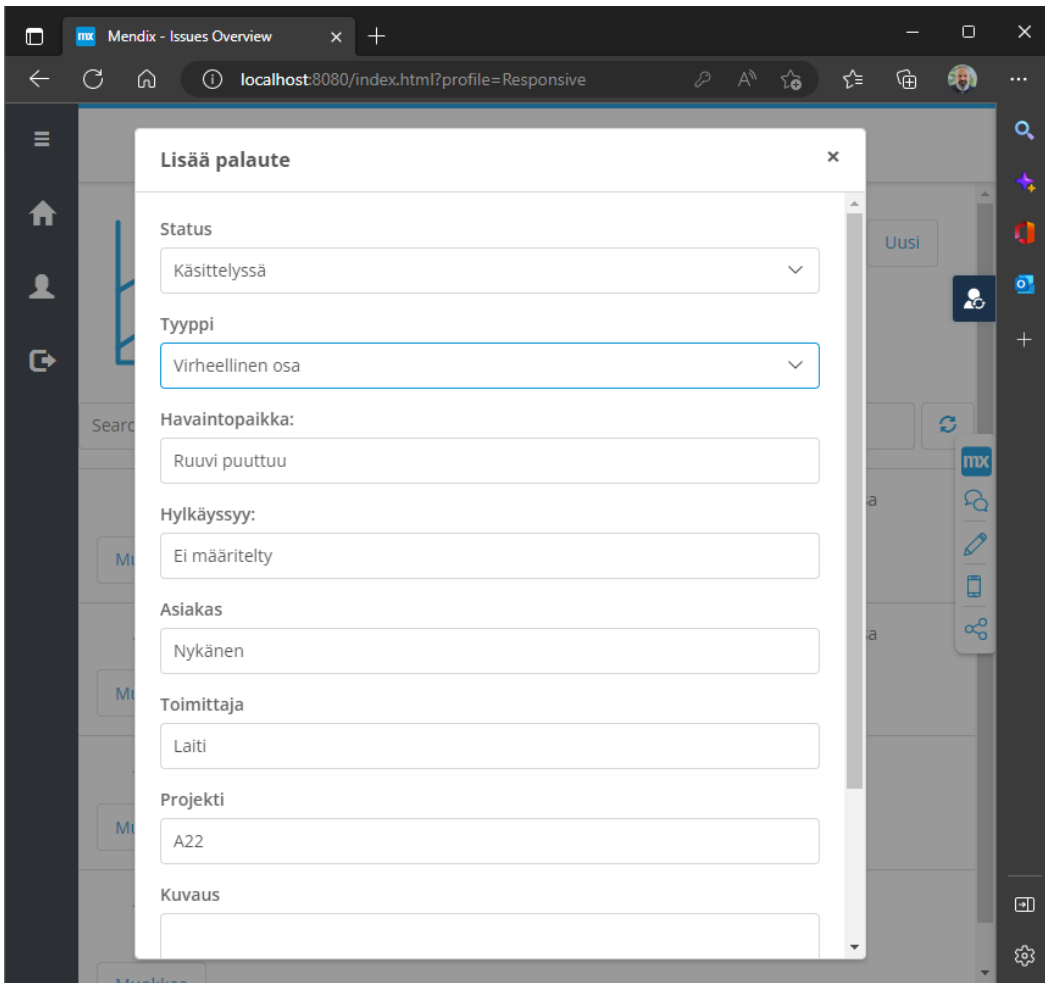


The screenshot shows a web application titled 'Mendix - Issues Overview' running on a local host. The main heading is 'Kirjatut laatu-palautteet' (Recorded quality feedback). There is a 'Uusi' (New) button in the top right and a 'Avoinna' (Open) dropdown menu. Below the heading is a search bar. The main content is a table with four columns: Status, Date/Time, Location, and Description. Each row has a 'Muokkaa' (Edit) button. The table contains four entries of quality feedback.

Status	Date/Time	Location	Description
Avoinna	1/11/2023, 9:29 AM	Halli Pasmation	Poikkeama Paraskone
Avoinna	1/11/2023, 9:32 AM	Varasto Pasmation	Puute Pulttikauppa
Avoinna	1/11/2023, 9:36 AM	Tehdas HyttiCabin	Poikkeama Pertsan paja
Avoinna	1/11/2023, 9:37 AM	Tehdas HyttiCabin	Virheellinen osa Pertsan paja

Kuva 8. Laatu-palautteiden selausnäköymä.

Oikean yläkulman Uusi-painikkeesta aukeaa näköymä 'Lisää palaute', jossa on kentät tarvittavien tietojen syöttämiseksi ja uusien laatu-palauttetietojen tallentamiseksi. Samaa näköymää käytetään jo kirjattujen palautteiden muokkaamiseen. Tällöin sama näköymä avataan ja kentät täytetään aiemmin kirjattun palautteen tiedoilla. Tallennettaessa tiedot päivittyvät olemassa olevalle palautteelle. Kuvassa 9. on esimerkki palautteen lisäysnäköymästä.



Kuva 9. Palautteiden lisäysnäkömä.

Yhteenveto

Mendix-kehitysympäristön sujuva käyttöönotto edellyttää huolellista perehtymistä Siemensin tarjoamiin ohjemateriaaleihin ja tutoriaaleihin. Kehitysympäristössä on kohtalaisen suuri määrä eri osioita ja valikoita ja sujuva käyttö ei onnistu, jos aika kuluu tarvittavien ominaisuuksien etsimiseen. Perustoiminnallisuuksien selvittyä, sovellusten rakentaminen on kohtalaisen helppoa ja pienen web-sovelluksen rakentaminen sujuu nopeasti. Myös sovelluksen päivittäminen ja ylläpito on helppoa.

Web-sovelluksen toiminta mukautuu automaattisesti eri päätelaitteille (mobiili, tabletti, työpöytä) ja sovelluksen toimintaan voidaan tehdä myös omia mukautuksia eri laitetyppejä varten.



Kehitysympäristössä käytetään pääasiassa ohjelmoinnista tuttuja käsitteitä, kuten muuttuja, olio, yhteys jne. (attribute, entity, object, association). Aihepiiriä tuntevalle tämä on luontevaa ja selkeää, mutta niiden jotka eivät ohjelmointiin ole aiemmin tutustuneet, pitää sisäistää käsitteet sujuvan kehitystyön mahdollistamiseksi.

Kehitysympäristön kokeilu on ilmaista ja kaikki toiminnot ovat tällöin kokeiltavissa ja käytössä. Oman sovelluksen saa myös otettua käyttöön Mendixin palvelimelle ja jakamalla tiedon oman sovelluksen verkko-osoitteesta voidaan sovellusta kokeilla helposti esimerkiksi yrityksen työntekijöiden kesken. Ilman maksullista tilausta sovelluksen käyttöaika kuitenkin yhdellä käyttökerralla on rajattu.