



Käyttöliittymä asiakaskohtaisten pankkiyhteyksien reititysasetusten hallintaan

Ilari Ijäs

OPINNÄYTETYÖ
Toukokuu 2024

Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

TIIVISTELMÄ
Tampereen ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma
Ohjelmistotekniikka

IJÄS, ILARI:
Käyttöliittymä asiakaskohtaisten pankkiyhteysreititysasetusten hallintaan

Opinnäytetyö 24 sivua
Toukokuu 2024

Opinnäytetyön tavoitteena oli kehittää Nomentia Oy:n Admin UI -web-sovellukseen käyttöliittymä, jonka avulla voidaan hallita Nomentian Cash & Treasury Management -web-sovelluksen asiakaskohtaisia pankkiyhteysreititysasetuksia. Käyttöliittymän tavoitteena on tehostaa ja helpottaa pankkiyhteysreititysasetusten hallintaa. Tarve tähän lähti liikkeelle konsulttien pyynnöstä, jotka halusivat helpottaa pankkiyhteysreititysasetusten hallintaa.

Työn toteutuksessa hyödynnettiin Visual Studio 2022- ja Visual Studio Code -kehitystyökaluja sekä C#- ja TypeScript-ohjelmointikieliä. Lisäksi käytettiin React JavaScript -kirjastoa käyttöliittymän rakentamiseen.

Käyttöliittymän avulla käyttäjät voivat tarkastella, lisätä, muokata ja poistaa pankkiyhteysreititysasetuksia. Asetusten selaamista helpottaa käyttöliittymän suodatustoiminto, joka mahdollistaa tietojen nopean ja tarkan seulonnan. Lisäksi käyttöliittymä estää virheellisten muutosten tekemisen tietokantaan. Projekti valmistui aikataulussa, saavutti kaikki tavoitteet ja Nomentia hyväksyi sen.

Asiasanat: visual studio 2022, visual studio code, c#, typescript, react

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Software Engineering

IJÄS, ILARI:
Interface for Managing Customer-Specific Bank Connection Routing Settings

Bachelor's thesis 24 pages
May 2024

The aim of the thesis was to develop a user interface for Nomentia Oy's Admin UI web application, enabling the management of customer-specific bank connection routing settings for Nomentia's Cash & Treasury Management web application. The need for the work arose from consultants' desire to facilitate the management of bank connection routing settings.

The purpose of the interface is to expedite and simplify the management of bank connection routing settings. Through the interface, users can view, add, edit, and delete bank connection routing settings. The browsing of settings is facilitated by the interface's filtering function, allowing for quick and precise data filtering. Additionally, the interface prevents erroneous changes from being made to the database.

The implementation of the project utilized Visual Studio 2022 and Visual Studio Code development environments, along with C# and TypeScript programming languages. Furthermore, the React JavaScript library was employed for building the user interface. The project was completed on schedule, met all the goals, and was approved by Nomentia.

Key words: visual studio 2022, visual studio code, c#, typescript, react

SISÄLLYS

1	JOHDANTO	6
2	YRITYSESITELY	7
3	PANKKIYHTEYSKANAVA	9
3.1	Maksuliikeaineiston lähettäminen pankkiin	9
4	KÄYTETYT TEKNOLOGIAT	11
4.1	TypeScript	11
4.2	React	11
4.3	.NET Framework	11
4.4	Azure SQL Database	12
4.5	Rest API	12
5	SUUNNITTELU	13
5.1	Ohjelmiston rakenne	18
6	TOTEUTUS	19
6.1	Palvelinpuoli	19
6.2	Käyttöliittymä	19
6.3	Testaus	21
7	POHDINTA	23
	LÄHTEET	24

LYHENTEET JA TERMIT

WS-kanava	Pankkiyhteyskanava yrityksille (Web Services)
SQL	Relaatiotietokannan käsittelyyn tarkoitettu standardoitu kyselikieli (Structured Query Language)
Transact-SQL	Microsoftin SQL Server -tietokantatuotepereheen laajalti käytetty SQL-dialekti.
SaaS	Ohjelmiston jakelumalli (Software as a Service), jossa ohjelmisto ylläpidetään palvelemilla ja käyttäjä pääsee käyttämään ohjelmistoa internetin välityksellä.
Skripti	Koodinpätkä, joka tekee tiettyjä tehtäviä, kuten tietokantakyselyn.
CRUD-pyynnöt	GET-, POST-, PUT- ja DELETE-pyynnöt
HTTP	Protokolla, jolla voidaan internetissä jakaa resursseja
TanStack Table	Ohjelmistokirjasto, joka tarjoaa käyttöliittymän tehokkaiden taulukoiden rakentamiseen.
JSON	(JavaScript Object Notation) Kevyt ja tekstipohjainen tiedostomuoto, jota käytetään tiedon siirtämiseen ja tallentamiseen.
API	(Application Programming Interface) Ohjelmistorajapinta, jonka avulla ohjelmistot voivat kommunikoida keskenään
XML	(Extensible Markup Language) Merkintäkieli, jota käytetään tietojen tallentamiseen ja välittämiseen.
SOAP-sanoma	XML-asiakirja, joiden avulla yrityksen ja pankin ohjelmisto kommunikoi
FileType	SOAP-sanoman XML-kenttä, joka kertoo sanoman sisällön tyypin

1 JOHDANTO

Tämän opinnäytetyön toimeksiantajana toimi Nomentia Oy. Opinnäytetyö tehtiin Nomentian Admin UI -web-sovellukseen. Admin UI on ainoastaan Nomentian henkilöstön, lähinnä konsulttien ja asiakaspalvelun käytössä oleva hallintatyökalu, jonka avulla voidaan tehdä ja hallita asiakaskohtaisia asetuksia tai esimerkiksi lähettää sähköpostia kaikille tai tietyille valituille asiakkaille. Cash & Treasury Management on web-sovellus, jolla sujuvoitetaan, automatisoidaan ja keskitetään yritysten maksuliikennettä, kassaennustamista ja treasury-prosesseja. Admin UI:lla hallittavat asetukset ovat Cash & Treasury Management:in yleisiä asetuksia tai asiakaskohtaisia asetuksia.

Työn tavoitteena on helpottaa ja nopeuttaa asiakaskohtaisten pankkiyhteyksien reititysasetuksien hallintaa. Pankkiyhteyksien reititysasetusten hallinnan helpottamiseksi tässä opinnäytetyössä tehdään käyttöliittymä, jonka avulla konsultit pystyvät sujuvasti muokkaamaan, luomaan tai poistamaan pankkiyhteyksien reititysasetuksia.

Tarve tälle työlle on lähtöisin konsulttien toiveesta helpottaa ja nopeuttaa pankkiyhteyksien reititysasetusten muokkaamista. Aikaisemmin asetuksia on pitänyt muokata SQL-skripteillä suoraan tietokantaan, mikä on hidasta ja kömpelöä. Konsultit itse eivät ole voineet ajaa skriptejä tietokantaan, vaan asiasta on pitänyt tehdä pyyntö SaaS-tiimille. Tämän jälkeen SaaS-tiimi on ajanut skriptin tietokantaan. Skriptin ajamis- pyynnön toteutumisessa kului aikaa ja se työllisti SaaS-tiimiä. Käyttöliittymä myös ehkäisee virheellisten muutosten tekemistä tietokantaan, koska käyttöliittymä validoi tehtyjä muutoksia tiettyjen sääntöjen mukaan. Tietokannan suorassa muokkaamisessa validointia ei tapahdu.

2 YRITYSESITTELY

Nomentia Oy on kansainvälinen ohjelmistotoimittaja, jonka kohdeasiakkaita ovat vaativien tarpeiden yritykset ja yhteisöt maksuliikenteen, kassanhallinnan ja treasury-palveluiden osa-alueilla. Useassa Euroopan maassa toimivassa monikulttuurisessa yrityksessä on yli 200 työntekijää. Yhtiön pääkonttori sijaitsee Espoossa, Suomessa, ja yli puolet yrityksen työntekijöistä työskentelee joko Espoossa tai Tampereella. Yrityksen tuotekehitys on pääosin keskitetty Tampereelle, joka on myös yrityksen suurin toimipiste työntekijämäärältään. Tuotekehitystä tehdään myös Wienissä, jossa työskentelee toiseksi eniten työntekijöitä.

Yrityksen historia alkaa 1980-luvun alussa, kun tamperelaiset ohjelmistoyritykset OpusCapita ja Analyste perustettiin. Näiden vuosien varrella Suomen markkinoilla tiukasti kilpailleiden yritysten yhdistyessä syntyi 2020-luvun alussa uusi Nomentia niminen yhtiö. Muutamaa vuotta myöhemmin yhtiö laajensi tuotevalikoimaansa liittämällä yhtiöön itävaltalaisen TIPCO-yrityksen, joka on erikoistunut treasury management -sovelluksiin.

Yrityksen strateginen kohdemarkkina-alue on jaettu useisiin maantieteellisiin alueisiin. Suomi on perinteisesti ollut ja on edelleen yrityksen suurin liikevaihdon lähde. Kuitenkin muita kasvavia markkinoita ovat muut Pohjoismaat, DACH-alue (Saksa, Itävalta ja Sveitsi), BeNeLux-maat (Belgia, Hollanti ja Luxemburg) sekä Iso-Britannia ja Irlanti.

Yli 2000 asiakasyrityksen työntekijät voivat käyttää Cash & Treasury Management -ohjelmistoa riippumatta maantieteellisestä sijainnista. Ohjelmisto on suunniteltu natiivisti pilvipohjaiseksi ja se toimii Microsoft Azuren alustalla.

Sovelluksen jakaminen tapahtuu SaaS-mallilla (Software as a Service). Nomentian tapauksessa tämä tarkoittaa sitä, että asiakkaat vuokraavat sovelluksen käyttöönsä tietyksi ajaksi sopimuksen perusteella. Nomentia puolestaan hoitaa kaiken sovelluksen ja sen alustan ylläpitoon, päivittämiseen ja tietoturvaan liittyvät asiat palveluna kuten esimerkiksi kuukausittaisten versiopäivitysten toimittamisen.

Sovellus ja sen alusta on suunniteltu ja kehitetty tietyin periaattein, joista tärkeimmät ovat moniasiakasratkaisu (multitenancy), standardisuudet ja skaalautuvuus. Tämä tarkoittaa, että tuhannet asiakkaat voivat käyttää samaa sovellusalustaa ja -sovellusta, ja niitä voidaan tarvittaessa skaalata. Jokainen asiakas käyttää täysin samanlaista standardituotetta ilman minkäänlaista koodimuokkausta, mikä helpottaa ylläpitoa ja päivityksiä.

3 PANKKIYHTEYSKANAVA

Pankkiyhteysteknologioita on maailmassa monia, mutta pohjoismaissa Web Services -teknologia (myöhemmin WS-kanava) on suosittu ratkaisu. Pankit käyttävät Web Services -teknologiaa yritysasiakkaiden ja pankin ohjelmiston väliseen maksuliikeviestintään. Web Services on liitántäratkaisu teknologia, jolla mahdollistetaan pankkien- ja yritysasiakkaiden laitteiden välinen automaattinen viestintä. Viestintä voi olla muun muassa maksuliikennettä, tili- ja tapahtumaotteita. Jokaisella Suomessa toimivalla pankilla on oma Web service -rajapinta. Teknologia perustuu kansainvälisiin standardeihin, joilla mahdollistetaan ohjelmistokehitysvälineiden käyttö ja ohjelmistotoimittajien kyky tuottaa toimivia asiakaspuolen sovittimia ja viestintäratkaisuja. (OP ryhmä 2024, 9.) Pankkien Web Service -rajapinnat mahdollistavat yritysten maksujen maksamisen suoraan Nomentian Cash & Treasury Management -ohjelmalla.

3.1 Maksuliikeaineiston lähettäminen pankkiin

WS-kanavan ja asiakkaan ohjelmiston välinen kommunikaatio tapahtuu SOAP-sanomilla (myöhemmin sanoma). Lähetettävät palvelupyynnöt ja niiden vastaukset ovat sanoman sisällä. SOAP-sanoma on standardin mukainen XML asiakirja, joka sisältää pankille olennaista tietoa. Sanoma sisältää muun muassa tietoturvaelementtejä. (OP ryhmä 2024, 7.)

XML muotoisien sanomien header-osassa on tietokenttä FileType, joka kertoo pankille, mitä maksutyyppiä sanoma on. Maksutyyppi voi esimerkiksi olla palkanmaksu. FileType tiedon mukaan pankin oma ohjelmisto osaa siirtää sanoman oikeanlaiseen käsittelyyn automaattisesti omassa järjestelmässään. (Nordea, OP Group, Danske Bank 2020.) Eri pankeilla on omat määritelmät FileTypeille. Esimerkiksi lähettäessä SEPA-maksun OP:lle ja Nordealle FileTypet ovat:

- Nordea: <FileType>NDCORPAYS</FileType>
- OP: <FileType>camt.001.001.03</FileType>

Nomentian Cash & Treasury Management -ohjelmistossa on valmiiksi määritelty useita erilaisia FileTypeja ja pankkiyhteyksiä. Esimerkiksi pankkiyhteys voi olla OP:n WS-kanava. Nomentian ohjelmiston on pystyttävä yhdistämään pankkiyhteys ja sille sopiva FileType maksuaineiston lähettämistä varten. Tämän yhdistämisen mahdollistavat ohjelmistossa olevat pankkiyhteysreititysasetukset, joita tässä opinnäytetyössä kehitetyllä käyttöliittymällä voi hallinnoida.

Pankkiyhteysreititysasetus on asetus, joka määrittää, missä pankkiyhteydessä käytetään mitään material type -arvoa. Material type -arvo on Nomentian ohjelmiston oma arvo, joka määrittää sanomalle tarvittavia ominaisuuksia. Suomen Web Services -pankkiyhteyksien material type -arvot määrittävät sanoman FileTypen.

4 KÄYTETYT TEKNOLOGIAT

Työssä käytetään Visual Studio 2022, Visual Studio Code ja Azure Data Studio -kehitystyökaluja. Ohjelmointikielinä käytetään TypeScriptiä ja C#:ia. Ohjelmistokehitysalustana toimii .NET Framework. JavaScript-kirjastona käytetään Reactia käyttöliittymän teossa. Relaatiotietokantojen hallintajärjestelmänä käytetään SQL Serveriä ja sen kyselykielenä toimii Transact-SQL.

4.1 TypeScript

TypeScript on Microsoftin kehittämä ohjelmointikieli, joka lisää tyyppityksen JavaScriptiin. Microsoftin mukaan TypeScript sopii paremmin suurien ohjelmistojen kehittämiseen. (Ars Technica 10.3.2012) TypeScript on kehitetty ratkaisemaan JavaScriptin ongelmia, kuten vähentämään tyyppityksen puutteesta johtuvia virheitä. Esimerkiksi tyyppityksen ansiosta kehitystyökalu pystyy varoittamaan virheistä kirjoittamisen aikana (TypeScript n.d).

4.2 React

React on Metan kehittämä avoimen lähdekoodin JavaScript-kirjasto, jota käytetään web -sovellusten käyttöliittymien rakentamiseen. Reactia voi käyttää myös TypeScriptin kanssa. Reactin avulla voi rakentaa kokonaisia ohjelmistoja käyttäen uudelleen käytettäviä komponentteja. (Stephan Miller 9.13.2021)

4.3 .NET Framework

.NET Framework on Microsoftin kehittämä ohjelmistokehitys, jota käytetään Microsoft Visual Studio -ympäristössä ohjelmistojen kehittämiseen. .Net-kehityksen avulla voidaan rakentaa muun muassa web- ja työpöytäsovelluksia. .Net-kehitys tukee useita eri ohjelmointikieliä, joista käytetyin on C#. C# on Microsoftin kehittämä avoimen lähdekoodin olio-ohjelmointikieli. (What is .NET? n.d)

4.4 Azure SQL Database

Azure SQL Database on PaaS (platform as a service) mallinen Microsoftin kehittämä pilvessä toimiva relaatiotietokantojen hallintajärjestelmä, joka pohjautuu SQL Server relaatiotietokantojen hallintajärjestelmään. SQL Database kyselykielenä toimii Transact-SQL. SQL Server on kehitetty standardoidun SQL kyselykielen pohjalta. Kyselykielellä voidaan tehdä hakuja, muutoksia ja lisäyksiä relaatiotietokantoihin. (What is SQL Server n.d; Microsoft 2024)

4.5 Rest API

REST API (Representational State Transfer Application Programming Interface) on ohjelmistorajapinta, joka perustuu REST-arkkitehtuurin periaatteisiin. REST on suunnittelumalli verkkopalveluiden toteuttamiseen, joka käyttää HTTP-protokollaa kommunikaatioon. RESTful API mahdollistaa resurssien (kuten tietokantatietojen) hallinnan ja manipuloinnin verkon kautta standardisten HTTP-metodien, kuten CRUD-metodien avulla. CRUD- metodit eli GET, POST, PUT ja DELETE, ovat yleisesti käytetyimpiä HTTP-metodeja. (Victor Yakubu 8.3.2024)

5 SUUNNITTELU

Opinnäytetyön käyttökokemuksen ja -liittymän suunnittelu tehtiin Figma-suunnittelutyökalulla. Toteutuksen hyväksymiskriteerit kirjattiin selkeästi ylös Product Backlog Itemille Azure DevOps työkalussa, jolla hallitaan ohjelmiston tuotteen kehitysjonoa. Kriteerit olivat seuraavat:

- Konsultti pystyy selaamaan olemassa olevia oletusreititysasetuksia ja suodattamaan niitä (kuva 1).
- Konsultti voi lisätä (kuva 2), muokata (kuva 4) ja poistaa (kuva 6) asiakaskohtaisia custom-asetuksia käyttöliittymän kautta. Default-asetuksia ei voi muokata, lisätä tai poistaa.
- Tietokantaan tulee lisätä uudet sarakkeet aikaleimoille, jotka näyttävät milloin asetus on luotu ja viimeksi muokattu.
- Asetusta luodessa, on mahdollista valita Material type valintaan vain ”ulospäin” meneviä tyyppejä.

Admin UI

Instance

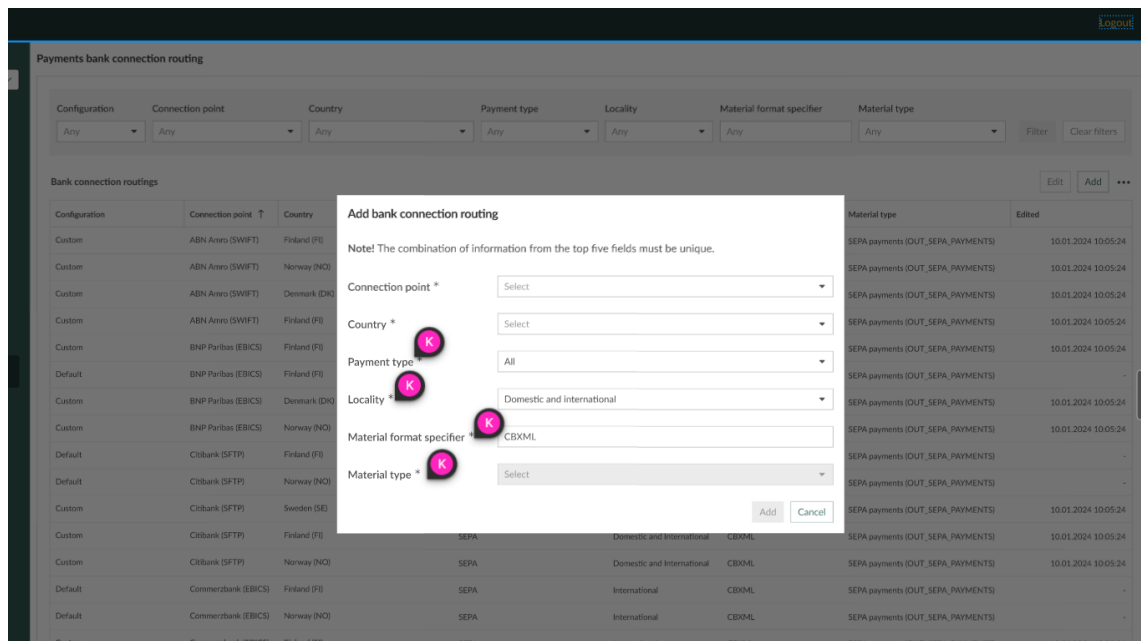
Operations

- Conversions
- Email notification
- Login notification
- Product licence
- SFTP Clients
- Storage
- Payments

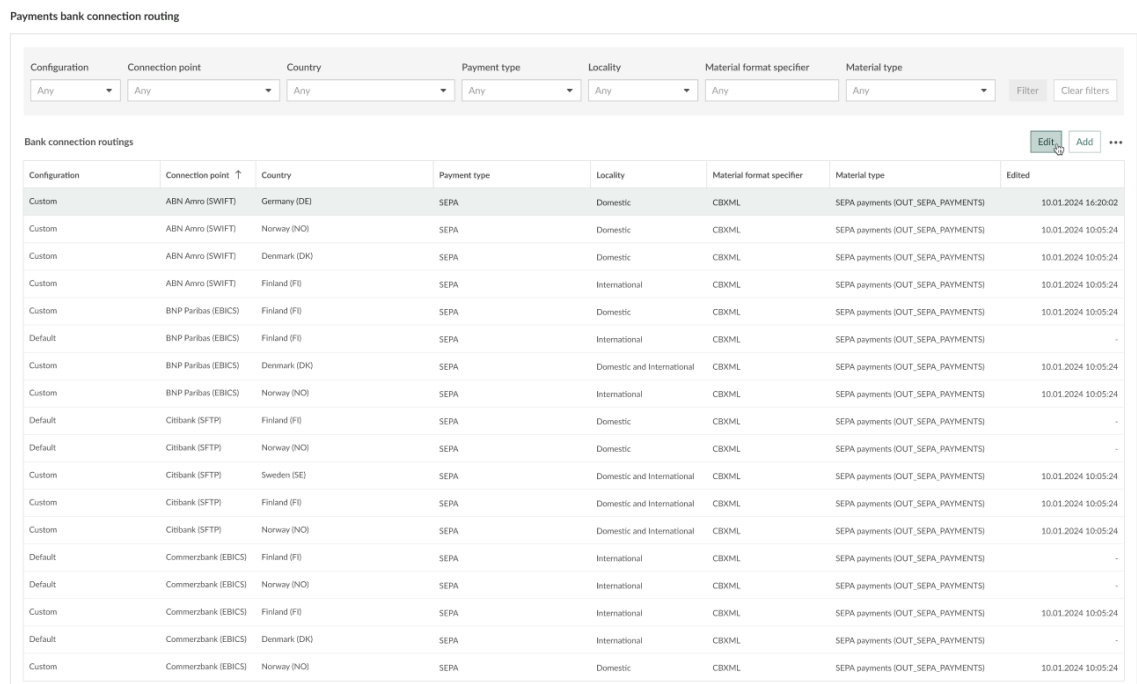
Payments bank connection routing

Configuration	Connection point	Country	Payment type	Locality	Material format specifier	Material type	Edited
Any	Any	Any	Any	Any	Any	Any	Filter Clear filters
Custom	ABN Anro (SWIFT)	Finland (FI)	SEPA	Domestic	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	ABN Anro (SWIFT)	Norway (NO)	SEPA	Domestic	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	ABN Anro (SWIFT)	Denmark (DK)	SEPA	Domestic	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	ABN Anro (SWIFT)	Finland (FI)	SEPA	International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	BNP Paribas (EBICS)	Finland (FI)	SEPA	Domestic	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Default	BNP Paribas (EBICS)	Finland (FI)	SEPA	International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	BNP Paribas (EBICS)	Denmark (DK)	SEPA	Domestic and International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	BNP Paribas (EBICS)	Norway (NO)	SEPA	International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Default	Citibank (SFTP)	Finland (FI)	SEPA	Domestic	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	-
Default	Citibank (SFTP)	Norway (NO)	SEPA	Domestic	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	Citibank (SFTP)	Sweden (SE)	SEPA	Domestic and International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	Citibank (SFTP)	Finland (FI)	SEPA	Domestic and International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	Citibank (SFTP)	Norway (NO)	SEPA	Domestic and International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Default	Commerzbank (EBICS)	Finland (FI)	SEPA	International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	-
Default	Commerzbank (EBICS)	Norway (NO)	SEPA	International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	Commerzbank (EBICS)	Finland (FI)	SEPA	International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Default	Commerzbank (EBICS)	Denmark (DK)	SEPA	International	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	Commerzbank (EBICS)	Norway (NO)	SEPA	Domestic	CBXXL	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24

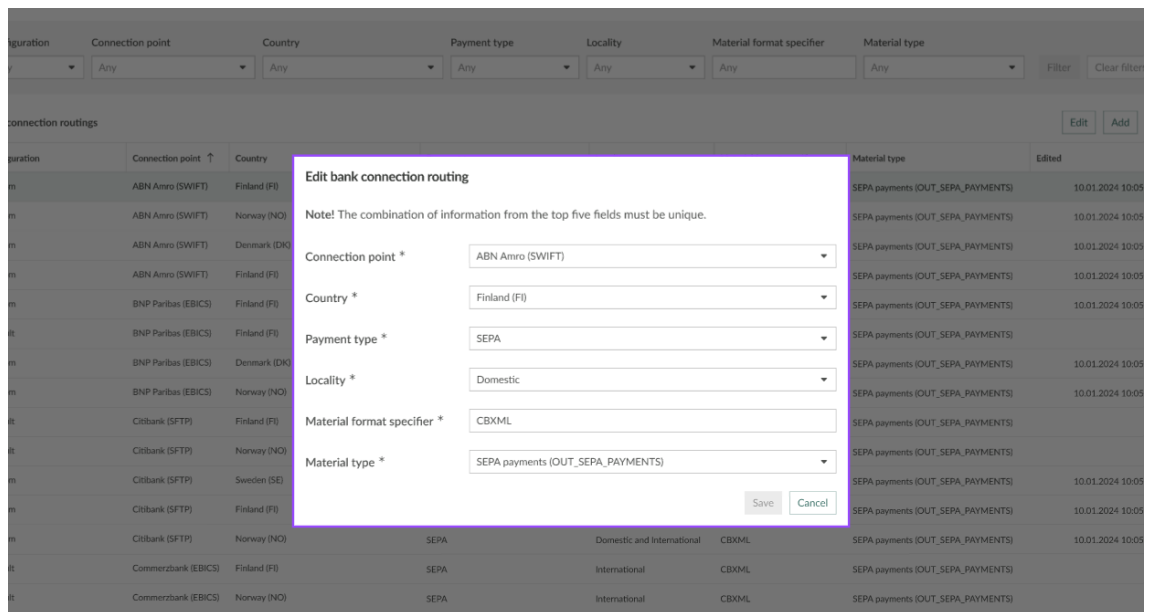
KUVA 1. Figma-suunnitelma aloitusnäkyästä



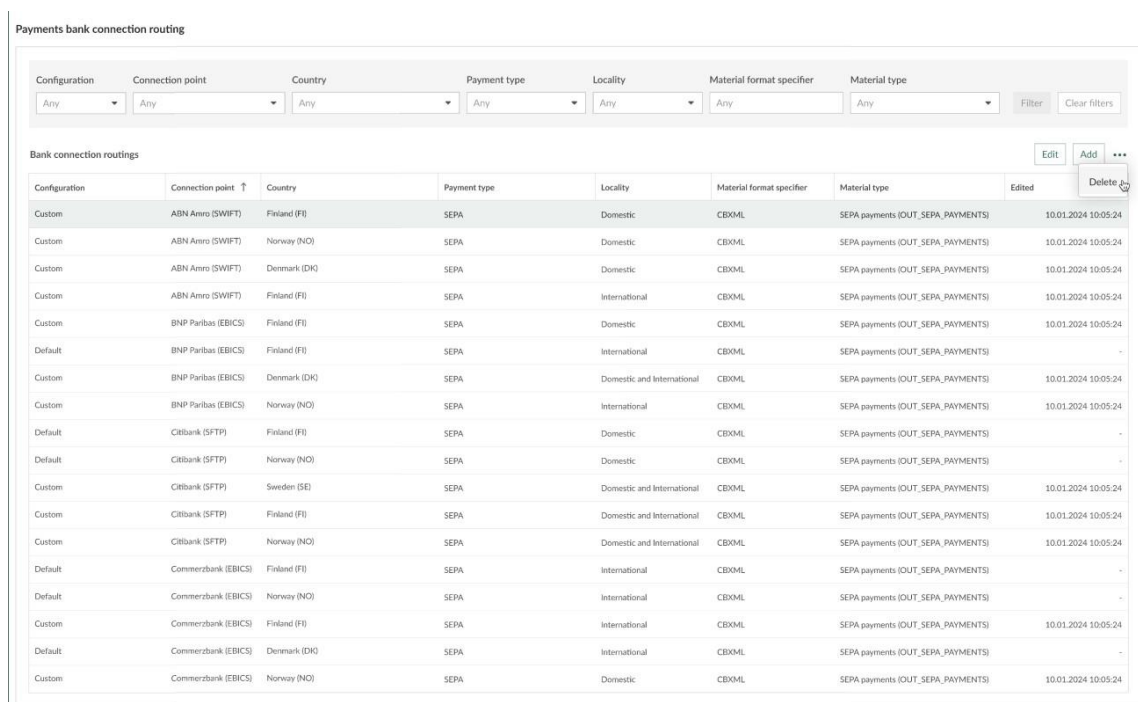
KUVA 2. Figma-suunnitelma uuden asetuksen lisäyksestä Add-painikkeen painamisen jälkeen



KUVA 3. Figma-suunnitelma rivin valinnasta ja Edit-painikkeen painamisesta



KUVA 4. Figma-suunnitelma asetuksen muokkaamisesta asetuksen valitsemisen ja Edit-painikkeen painamisen jälkeen



KUVA 5. Figma-suunnitelma rivin valitsemisesta ja Delete-painikkeen painamisesta

Connection point	Country	Payment type	Locality	Material format specifier	Material type	
ABN Amro (SWIFT)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
ABN Amro (SWIFT)	Norway (NO)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
ABN Amro (SWIFT)	Denmark (DK)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
ABN Amro (SWIFT)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
BNP Paribas (EBICS)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
BNP Paribas (EBICS)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
BNP Paribas (EBICS)	Denmark (DK)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
BNP Paribas (EBICS)	Norway (NO)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
Citibank (SFTP)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
Citibank (SFTP)	Norway (NO)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
Citibank (SFTP)	Sweden (SE)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
Citibank (SFTP)	Finland (FI)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
Citibank (SFTP)	Norway (NO)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2
Commerzbank (EBICS)	Finland (FI)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:2

KUVA 6. Figma-suunnitelma asetuksen poistamisesta

Käyttöliittymällä voi vain lisätä asetuksia, joilla arvojen connection point, country, payment type, locality ja material format specifier yhdistelmä on yksilöllinen verrattuna muihin olemassa oleviin custom-asetuksiin (kuva 7).

Connection point	Country	Payment type	Locality	Material format specifier	Material type	Edited
ABN Amro (SWIFT)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
ABN Amro (SWIFT)	Norway (NO)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
ABN Amro (SWIFT)	Denmark (DK)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
ABN Amro (SWIFT)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
Paribas (EBICS)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
Paribas (EBICS)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
Paribas (EBICS)	Denmark (DK)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
Paribas (EBICS)	Norway (NO)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
ank (SFTP)	Finland (FI)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
ank (SFTP)	Norway (NO)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
ank (SFTP)	Sweden (SE)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
ank (SFTP)	Finland (FI)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
ank (SFTP)	Norway (NO)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0
merzbank (EBICS)	Finland (FI)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.0

KUVA 7. Figma-suunnitelma virhe palautteesta, jos yrittää lisätä asetusta epäuniikilla arvojen kombinaatiolla

Payments bank connection routing

Configuration	Connection point	Country	Payment type	Locality	Material format specifier	Material type	
Any	Citibank (SFTP) x	2 selected x	Any	Any	Any	Any	Filter Clear filters

Bank connection routings Edit Add ...

Configuration	Connection point ↑	Country	Payment type	Locality	Material format specifier	Material type	Edited
Custom	ABN Amro (SWIFT)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	ABN Amro (SWIFT)	Norway (NO)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	ABN Amro (SWIFT)	Denmark (DK)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	ABN Amro (SWIFT)	Finland (FI)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	BNP Paribas (EBICS)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Default	BNP Paribas (EBICS)	Finland (FI)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	BNP Paribas (EBICS)	Denmark (DK)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	BNP Paribas (EBICS)	Norway (NO)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Default	Citibank (SFTP)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	-
Default	Citibank (SFTP)	Norway (NO)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	Citibank (SFTP)	Sweden (SE)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	Citibank (SFTP)	Finland (FI)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	Citibank (SFTP)	Norway (NO)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Default	Commerzbank (EBICS)	Finland (FI)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	-
Default	Commerzbank (EBICS)	Norway (NO)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	Commerzbank (EBICS)	Finland (FI)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Default	Commerzbank (EBICS)	Denmark (DK)	SEPA	International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	Commerzbank (EBICS)	Norway (NO)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24

KUVA 8. Figma-suunnitelma suodattimien valinnasta ja Filter-painikkeen painamisesta

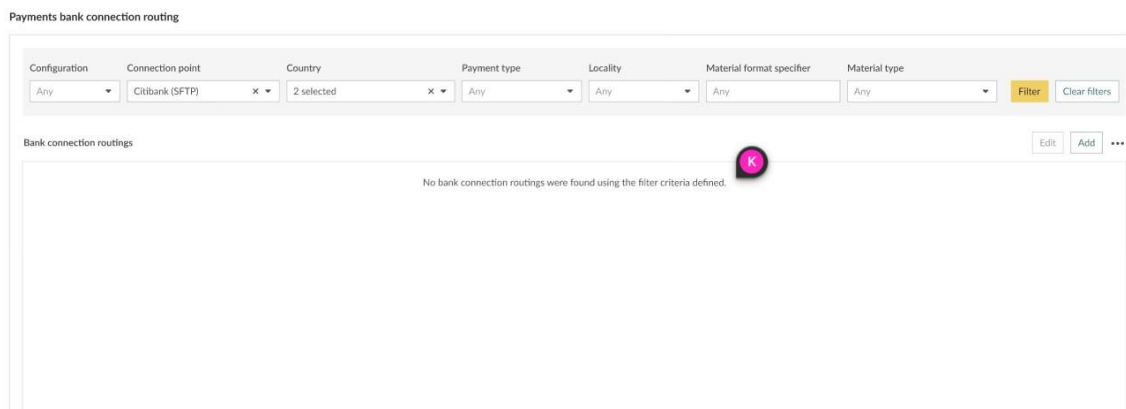
Payments bank connection routing

Configuration	Connection point	Country	Payment type	Locality	Material format specifier	Material type	
Any	Citibank (SFTP) x	2 selected x	Any	Any	Any	Any	Filter Clear filters

Bank connection routings Edit Add ...

Configuration	Connection point ↑	Country	Payment type	Locality	Material format specifier	Material type	Edited
Default	Citibank (SFTP)	Finland (FI)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	-
Default	Citibank (SFTP)	Norway (NO)	SEPA	Domestic	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	-
Custom	Citibank (SFTP)	Finland (FI)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24
Custom	Citibank (SFTP)	Norway (NO)	SEPA	Domestic and International	CBXML	SEPA payments (OUT_SEPA_PAYMENTS)	10.01.2024 10:05:24

KUVA 9. Figma-suunnitelma Filter-painikkeen painamisen jälkeen



KUVA 10. Figma-suunnitelma näkymästä, kun valituilla suodattimilla ei löydy yhtäkään asetusta

5.1 Ohjelmiston rakenne

Ohjelmiston rakenne suunniteltiin toimivan REST API -rajapintaa käyttäen. Asetusten hallitseminen toimisi CRUD-pyyntöillä. Käyttöliittymä lähettäisi erilaisia API-pyyntöjä ohjelmiston palvelinpuoleen käyttäjän halutessa tehdä seuraavia toimintoja:

- Hakea olemassa olevat asetukset = GET-pyyntö
- Poistaa valittu asetukset = DELETE-pyyntö
- Lisätä uusi asetukset = POST-pyyntö
- Muokata valittu asetukset = PUT-pyyntö

Kutsun muodon mukaan palvelinpuoli osaisi tehdä halutun toiminnon tietokantaan, jonka jälkeen käyttöliittymä hakisi uudella API-kutsulla päivittyneen tietokannan sisällön ja näyttäisi sen käyttöliittymän taulukossa.

Käyttöliittymän taulukkona suunniteltiin käytettävän TanStack Table -taulukkoa, joka on taulukon rakentamista helpottava kirjasto. TanStack Table helpottaa taulukon datan käsittelyä esimerkiksi valmiiden rivien suodattamis- ja järjestämistoiminnoilla.

6 TOTEUTUS

6.1 Palvelinpuoli

Ensimmäiseksi tehtiin Admin UI:n palvelinpuolelle toimivat API-rajapinnat CRUD-pyyntöjä varten. Rajapinnan ohjelmointi tehtiin Microsoft Visual Studio 2022 -kehitystyökalulla, ja ohjelmointikielenä käytettiin C#:ia. Palvelimella suoritettavat tietokantakyselyt Azure SQL Database tietokantaan toteutettiin Transact-SQL kielellä.

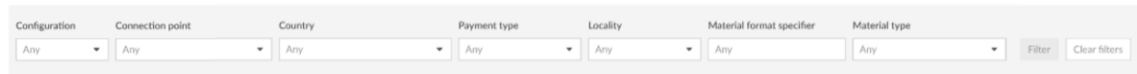
Kaikkien asetusten hakemiseen käytettävän GET-pyyntön saadessaan palvelin hakee oikeasta taulusta kaikki asetukset ja palauttaa ne JSON-muodossa käyttöliittymälle. Uuden asetuksen lisäykseen käytettävässä POST-pyyntössä on mukana uuden asetuksen data JSON-muodossa, jonka pohjalta palvelin lisää uuden asetuksen tietokannan tauluun. DELETE-pyyntön mukana on asetuksen yksilöivä id, jonka avulla palvelin tietää, mikä asetusta pitää poistaa tietokannasta. Olemassa olevan asetuksen muokkaamista varten PUT-pyyntössä on mukana muokattavan asetuksen yksilöivä id ja uusi muokattu data.

6.2 Käyttöliittymä

Rajapinnan valmistumisen jälkeen aloitettiin käyttöliittymän ohjelmointi. Taulukon rakentaminen aloitettiin hakemalla GET-pyyntöllä palvelimen rajapinnasta asetusten data käyttöliittymälle. Palvelimelta saatu data syötetään TanStack Table -kirjaston taulukkoon, mikä näytetään käyttöliittymällä.

Seuraavaksi lisättiin taulukkoon rivien suodatus toiminto. Suodattimella pystytään suodattamaan vain halutun muotoiset asetukset näkyviin, mikä helpottaa haluttujen arvojen löytämistä. Suodattimien valitseminen toimii pudotusvalikoilla, joista löytyy mahdolliset vaihtoehdot (kuva 11). Suodattimessa on Filter-painike, jota voi painaa, kun vähintään yksi suodatin on valittuna (kuva 8). Taulukko suodattuu vasta sitten kun Filter-painiketta on painettu (kuva 9). Lisäksi suodatin komponentissa on Clear filters -painike, jota painaessa kaikki suodattimet poistuvat ja taulukossa näkyy taas kaikki asetukset. Painiketta ei voi painaa, jos suodatusta ei ole laitettu päälle Filter-painikkeella. Lisäksi taulukkoon lisättiin

toiminto, joka lajittelee asetuksen Connection point -arvon mukaan aakkosjärjestykseen.



KUVA 11. Suodatus komponentti

Uuden asetuksen lisäystoimintoa varten tehtiin erillinen valintaikkuna, joka aukeaa ikään kuin sivuston päälle (kuva 2). Valintaikkunan avautuessa kenttiin payment type, locality ja material format specifier määritettiin oletusarvot (kuva 2). Material type -kenttä avautuu arvon syöttöä varten vasta, kun connection point -arvo on valittu. Tämä johtuu siitä, että kullekin connection point -arvolle on käytettävissä vain tietyt material type -vaihtoehdot. Näkymän saa auki painamalla Add-painiketta. Valintaikkunan viiden ylimmän arvon pitää olla yksilöllinen lisättäessä asetusta. Yksilöllisen yhdistelmän tarkistus toimii etsimällä, löytyykö kyseistä yhdistelmää vastaavaa asetusta taulukosta. Jos asetusta vastaavalla yhdistelmällä löytyy, näytetään virhe valintaikkunassa (kuva 7). Add-painiketta voi painaa, kun kaikissa kentissä on jokin arvo ja määriteltyjen arvojen yhdistelmä on yksilöllinen. Asetuksen lisäyksessä lähetetään POST-pyyntö palvelimelle uusilla asetuksen arvoilla ilman id-arvoa. Tämän jälkeen asetusta lisätään tietokantaan ja käyttöliittymän taulukko päivitetään.

Asetuksen muokkaamisessa käytettiin samaa valintaikkuna komponenttia, kuin uuden asetuksen lisäyksessä. Komponentille syötettävien arvojen perusteella komponentti osaa muuttaa toiminnallisuuttaan lisäys tai muokaus tilanteeseen sopivaksi. Edit-painiketta voi painaa vain, kun custom-asetus on valittuna (kuva 3). Valintaikkunan avautuessa valitun asetuksen arvot tulevat näkyviin valintaikkunan valinta kenttiin (kuva 4). Save-painiketta voi painaa, kun arvojen yhdistelmä on yksilöllinen, vähintään yhtä arvoa on muokattu ja kaikissa kentissä on jokin arvo. Muokkauksessa lähetetään PUT-pyyntö palvelimelle muokattujen arvojen ja asetuksen id-arvon kanssa. Asetuksen id-arvon perusteella palvelin tietää, mitä asetusta muokata.

Asetuksen poistoa varten tehtiin uusi valintaikkuna komponentti (kuva 6). Valintaikkuna avautuu painamalla Delete-painiketta, joka löytyy kolmen pisteen valikon alta (kuva 5). Delete-painiketta voi painaa vain, kun custom-asetus on valittuna. Valintaikkunaan laitettiin varmistuskysymys, jossa näytetään valitun asetuksen connection point, country ja material type -arvot. Poistamisessa lähetetään palvelimelle DELETE-pyyntö, jonka mukana on valitun asetuksen id-arvo. Id-arvon perusteella palvelin poistaa asetuksen tietokannasta.

Lopuksi lisättiin vielä Edited-sarake tietokannan tauluun, sekä käyttöliittymään. Uuden sarakkeen lisäystä varten tehtiin SQL-skripti toiseen Nomentian Git tietosäiliöön, jolla ajetaan tietokantamuutoksia ohjelmistoalustan päivityksien yhteydessä. Rakentaessa tietosäiliön skriptit ajetaan ja niillä muun muassa määritellään tietokantojen taulujen rakenne. Palvelinpuolelle lisättiin asetuksen luomisen ja muokkaamisen SQL-kyselyihin rivi, jossa lisätään kyselyn ajon aikainen aikaleima edited -sarakeeseen. Käyttöliittymän taulukkoon lisättiin uusi Edited-sarake sekä funktio, joka muokkaa aikaleimadatan alueelliseen aikaformaattiin. Esimerkiksi Suomessa muoto olisi pp.kk.vvvv tt:mm:ss.

6.3 Testaus

Ominaisuuden toteutuksen jälkeen kokonaisuus siirrettiin testattavaksi manuaalitestajaalle. Manuaalitestaja testasi käyttöliittymän virheiden löytämiseksi. Manuaalitestaja käytti käyttöliittymää simuloidakseen oikean käyttäjän roolia ja tarkastaakseen, toimiiko se oikealla tavalla. Virheiden löytyessä käyttöliittymä siirrettiin takaisin toteutukseen korjattavaksi. Testauksessa huomattiin, että valintaikkunoissa material format specifier -kenttään pystyi syöttämään loputtoman määrän merkkejä, mikä muokkauksen tai lisäyksen yhteydessä aiheutti palvelinpyynnön epäonnistumisen, koska tietokannan kentän maksimipituus on 100 merkkiä. Tämä korjattiin laittamalla tekstikenttään 100 merkin pituus rajoitin.

Manuaalitestajan hyväksynnän jälkeen tuote siirrettiin automaatiotestaajalle. Automaatiotestaaja lisäsi ohjelmistoon automaatiotestejä, joilla testataan automaattisesti käyttöliittymän toimiminen. Jos ohjelmistoon tehtäisi

tulevaisuudessa muutoksia, jotka rikkoisivat käyttöliittymä toiminnallisuuden, niin automaatiotestit huomaisivat sen. Manuaali- ja automaatiotestauksen hyväksynnän jälkeen käyttöliittymä yhdistettiin Admin UI -web-sovelluksen päähaaraan, josta se kuukausittaisen versiopäivityksen yhteydessä otetaan käyttöön.

7 POHDINTA

Työtavoite saavutettiin erittäin hyvin. Tavoitteena oli mahdollistaa asiakaskohtaisten pankkiyhteysreititysasetusten nopeampi ja helpompi muokkaaminen, luominen ja poistaminen samalla vähentäen virheellisten muutosten tekemistä tietokantaan. Työ mahdollistaa juuri tämän.

Työn tekeminen sujui hyvin, eikä suurempia ongelmia ilmennyt. Työn toteuttaminen vaati perehtymistä Tanstack Table -kirjastoon, TypeScript-ohjelmointikieleen sekä C#-ohjelmointikielellä tehtävään palvelinpuolen toteutukseen, jotka eivät olleet aiemmin tuttuja. Uusien asioiden oppiminen lisäsi työn tekemiseen mielenkiintoa. Lisäksi sain lisää kokemusta Reactin käytössä.

Käyttöliittymän odotetaan nopeuttavan ja helpottavan asiakaspalvelijoiden ja konsulttien työtä. Lisäksi SaaS-tiimin työmäärän odotetaan vähenevän, koska enää heidän ei tarvitse suorittaa SQL-skriptejä tietokantaan.

Tulevaisuudessa käyttöliittymää tullaan parantamaan saadun palautteen perusteella. Palautteen pohjalta mietitään, miten käyttöliittymää voidaan kehittää entistä paremmaksi. Lisäksi käyttöliittymää voitaisiin optimoida nopeammaksi. Yksi tapa tähän olisi taulukon rivien virtualisointi, jolloin käyttöliittymä renderöisi vain näytöllä näkyvät rivit ja suorituskyky paranisi.

LÄHTEET

Ars Technica. 10.3.2012. Microsoft TypeScript: the JavaScript we need, or a solution looking for a problem? Verkkosivu. Viitattu 11.5.2024.

<https://arstechnica.com/information-technology/2012/10/microsoft-typescript-the-javascript-we-need-or-a-solution-looking-for-a-problem/>

Microsoft. n.d. What is .NET? Verkkosivu. Viitattu 11.5.2024.

<https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>

Microsoft. 2024. What is Azure SQL Database? Viitattu 23.5.2024.

<https://learn.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql>

Nordea, OP Group, Danske Bank. 2020. Security and Message Specification for Financial Messages using Web Services. Pdf- dokumentti. Viitattu 18.5.2024.

https://www.finanssiala.fi/wp-content/uploads/2021/03/WebServices_Messages_v110_20200504.pdf

OP Ryhmä. 2024. Web Services -kanavan asiakasohje. Pdf-dokumentti. Viitattu 22.4.2024.

<https://www.op.fi/documents/20556/35498061/Yrityksen+pankkiyhteyskanavan+sovellusohje/dcc66747-f139-3efc-de52-8a57ee4d9a06>

SQLServertutorial.net. n.d. What is SQL Server. Verkkosivu. Viitattu 11.5.2024.

<https://www.sqlservertutorial.net/getting-started/what-is-sql-server/>

Stephan Miller. 9.13.2021. What Is React? Verkkosivu. Viitattu 11.5.2024.

<https://www.codecademy.com/resources/blog/what-is-react/>

TypeScript. n.d. What is TypeScript? Verkkosivu. Viitattu 11.5.2024.

<https://www.typescriptlang.org>

Victor Yakubu. 8.3.2024. How to Create a CRUD API – NodeJS and Express Project for Beginners. Verkkosivu. Viitattu 11.5.2024.

<https://www.freecodecamp.org/news/create-crud-api-project/#what-is-a-crud-api>