OULUN AMMATTIKORKEAKOULU

Konsta Holm

TYÖMAA- JA URAKKATILITYSTIETOJEN RAJAPINTAHAKU FORESTPRO-TOIMINNANOHJAUSJÄRJESTELMÄSTÄ

TYÖMAA- JA URAKKATILITYSTIETOJEN RAJAPINTAHAKU FORESTPRO-TOIMINNANOHJAUSJÄRJESTELMÄSTÄ

Konsta Holm
Opinnäytetyö
Kevät 2022
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

---

Tekijä: Konsta Holm
Opinnäytetyön nimi: Työmaa- ja urakkatilitystietojen rajapintahaku ForestPro-toiminnanohjausjärjestelmästä
Työn ohjaajat: Pekka Alaluukas & Juha Valkola
Työn valmistumislukukausi ja -vuosi: Kevät 2022 Sivumäärä: 26

Opinnäytetyö tehtiin Pinja Solutions Oy:lle, joka toimii Pinja Group-konsernin alla kehittäen räätälöityä toiminnanohjausjärjestelmää metsäteollisuuteen.

Työn aiheena oli luoda mahdollisuus hakea työmaa- sekä urakkatilitystietoja pilvipalvelussa olevan rajapinnan avulla ForestPro-toiminnanohjausjärjestelmästä käyttäen REST-rajapintakutsua. Tämän ominaisuuden ansiosta ulkoinen toimija saa ForestPro-järjestelmää käyttävien motojen ja ajokoneiden työohjeet, -raportit sekä urakkatilitystiedot omaan metsäpalvelujärjestelmäänsä.

Työssä sovellettiin HTTP-protokollaan perustuvaa REST-ohjelmointirajapintaa yhteistyössä .NET-sovelluskehyksen tarjoaman Windows Communication Foundation -palvelun kanssa, joka käyttää ForestPro-järjestelmän projektitiedostoja tietojen hakemiseen sekä käsittelyyn.

Projekti saatiin toimitettua asiakkaalle muutaman korjauksen jälkeen. Lisäksi muokattiin rajapintaa niin, että ominaisuuden saa aktivoitua useammalle asiakkaalle kerralla. Johtopäätöksenä projekti oli onnistunut kokonaisuudessaan huolimatta haasteellisesta aikataulusta. Opinnäytetyö opetti havainnollistamaan käytettyjä teknologioita myös teorian tasolla entisestään.

Avainsanat: C#, olio-ohjelmointi, HTTP, REST API, .NET Framework, WCF, SQL, Pilvipalvelu, Toiminnanohjausjärjestelmät

ABSTRACT

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

---

Authors: Konsta Holm
Title of thesis: Workingsite- and contract job accounting information interface fetching from ForestPro-ERP system
Supervisors: Pekka Alaluukas & Juha Valkola
Term and year when the thesis was submitted: Spring 2022
Number of pages: 26

The thesis was done for Pinja Solutions Oy, which operates under the Pinja Group, developing a customized ERP system for the forest industry.

The topic of the work was to create an option to retrieve workingsite and contract accounting job data using the interface from the ForestPro ERP system using the REST API call. Thanks to this feature, an external operator receives work instructions, reports and contract accounting information from Harvester and Forwarder using the ForestPro system in their own forest service system.

The REST programming interface based on the HTTP protocol was applied in cooperation with the WCF (Windows Communication Foundation) service provided by the .NET application framework, which uses ForestPro system's project files to fetch and process data.

Keywords: C#, object-oriented programming, HTTP, REST API, .NET Framework, LINQ, WCF, SQL, Cloud service, ERP software

SISÄLLYS

1 JOHDANTO

Opinnäytetyö on osana tieto- ja viestintätekniikan insinöörin opintosuunnitelmaa ja opinnäytetyön projekti toteutettiin Pinja Solutions Oy:lle syksyllä 2021 ja dokumentoitiin vuoden loppuun mennessä. Suoritin opintosuunnitelmaan kuuluvat kesätyöharjoittelut sekä yrityslähtöiset tuotekehitysprojektit samaiseen yritykseen, joten opinnäytetyöhön siirtyminen ja sen toteuttaminen sujuivat sulavasti. Opinnäytetyössä tarkastellaan projektissa käytettyjä teknologioita ja niiden menetelmiä käytännössä.

Pinja Solutions Oy on vuonna 1998 perustettu osakeyhtiö, joka aikaisemmin toimi PiiMega Oy -toiminimellä, mutta siirtyi myöhemmin Pinja-konsernin alaisuuteen. Pinja Solutionsin pääasiallinen tavoite on metsäteollisuuden arvoketjun digitalisointi. Olen itse Forest by Pinja -tiimissä, joka keskittyy puunhankinnan, korjuun ja logistiikan toiminnanohjausjärjestelmään asiakaskohtaisilla räätälöidyillä muutoksilla.

Opinnäytetyön aiheena oli luoda Pinja Solutions Oy:n pilvipalvelimen palveluihin uudet palvelut, joiden kautta ulkoinen toimija pystyy hakemaan työmaatietoja sekä urakkatilityksiä ForestProsta omaan toiminnanohjausjärjestelmäänsä. Molemmat palvelut toimivat periaatteeltaan samalla tavalla eli välittävät valmista XML-tyyppistä tekstisanomaa ulkoisen toiminnanohjausjärjestelmän ja asiakkaan pilvipalvelun välillä. Asiakkaan pilvipalvelun tietokantaan lokitetaan kerätyt tiedot sekä lähetetty sanoma. Palvelussa käytetään todentamisessa käyttäjätunnusta ja salasanaa, joiden avulla tietoja pystytään hakemaan. Näin saadaan riittävä varmuus siitä, että asiattomilta hauilta pääsy palveluun on estetty. Kaikki tiedonsiirrot tapahtuvat salatun HTTPS-yhteyden avulla.

2 HTTPS JA PILVIPALVELUT

HTTP on lyhenne sanoista Hypertext Transfer Protocol eli hypertekstin siirtoprotokolla. Protokolla perustuu asiakasohjelman, kuten selaimen tai WWW-palvelimen, avaamaan luotettavaan TCP-yhteyteen palvelimelle, johon pyyntö lähetetään. WWW-palvelin tarkoittaa palvelintietokonetta tai ohjelmistoa, joka jakaa verkkosivuja HTTP-protokollalla asiakasohjelmille ja koneille. TCP on lyhenne sanoista Transmission Control Protocol. Sen tarkoitus on pitää huolta, että lähetetyt paketit saapuvat perille oikeassa järjestyksessä ja tarvittaessa hävinnyt paketti voidaan lähettää uudelleen. HTTP-pyyntö voi palauttaa HTML-dokumentin, tekstitiedoston, kuvan tai minkälaisen tiedoston tai virheen tahansa. (Wikipedia 2022c.)

HTTPS on lyhenne sanoista HyperText Transfer Protocol Secure. Se on HTTP-protokollan ja TLS/SSL-protokollan yhdistelmä, jota käytetään tiedon suojattuun siirtoon verkossa. TLS eli Transport Layer Security on salausprotokolla, jolla voidaan suojata Internet-sovellusten tietoliikenne IP-verkkojen yli. Lyhyesti selitettynä TSL perustuu varmenteisiin kuten epäsymmetrinen salaus ja digitaalinen allekirjoitus, joilla sivustot todentavat olevansa luotettavia. Tämän protokollan ansiosta siirto on salattu sekä palvelimelle että palvelimelta. (Wikipedia 2022b.)

Pilvipalvelu tarkoittaa tietoteknisten palveluiden toimittamista tarvittaessa, tyypillisesti internetin välityksellä. Palveluiden pääluokat ovat Saas (Software as a Service) eli sovellusohjelmisto, jota palvelun tarjoaja ylläpitää omilla palvelimillaan. IaaS (Infrastructure as a Service), jossa palveluntuottaja tarjoaa infrastruktuuria palveluna asiakkaalle tyypillisesti niin, että asiakkaan käyttöön tarjotaan web-pohjainen hallintaliittymä, jonka kautta voi itse perustaa tarvittavia palvelimia sekä hallinnoida palvelimen kapasiteettia, verkkoyhteyksiä ja palomuurauksia. PaaS (Platform as a Service), josta puhutaan, kun palveluntuottaja tarjoaa palveluna sovellusalustoja, jotka on paketoitu helposti käyttöön otettavaan muotoon. (Kangasniemi & Lintulahti 2017.)

Käsitettä ”pilvi” käytetään kielikuvana, jolla viitataan internetiin siten, kuin se usein esitetään tietoverkkojen kaaviokuvissa, sekä abstraktiona monimutkaiselle infrastruktuurille, jonka se verhoaa. (Kuva 1.)

KUVA 1. Pilvipalvelu ja sen vuorovaikutus yksinkertaistettuna. (Homework18 2021)

Pilvipalvelun etu on, että siihen tallennettuihin tiedostoihin tai palveluihin pääsee käsiksi millä tahansa internet-yhteydellä varustetulla laitteella, kunhan niihin on oikeudet. (Liimatta 2021.)

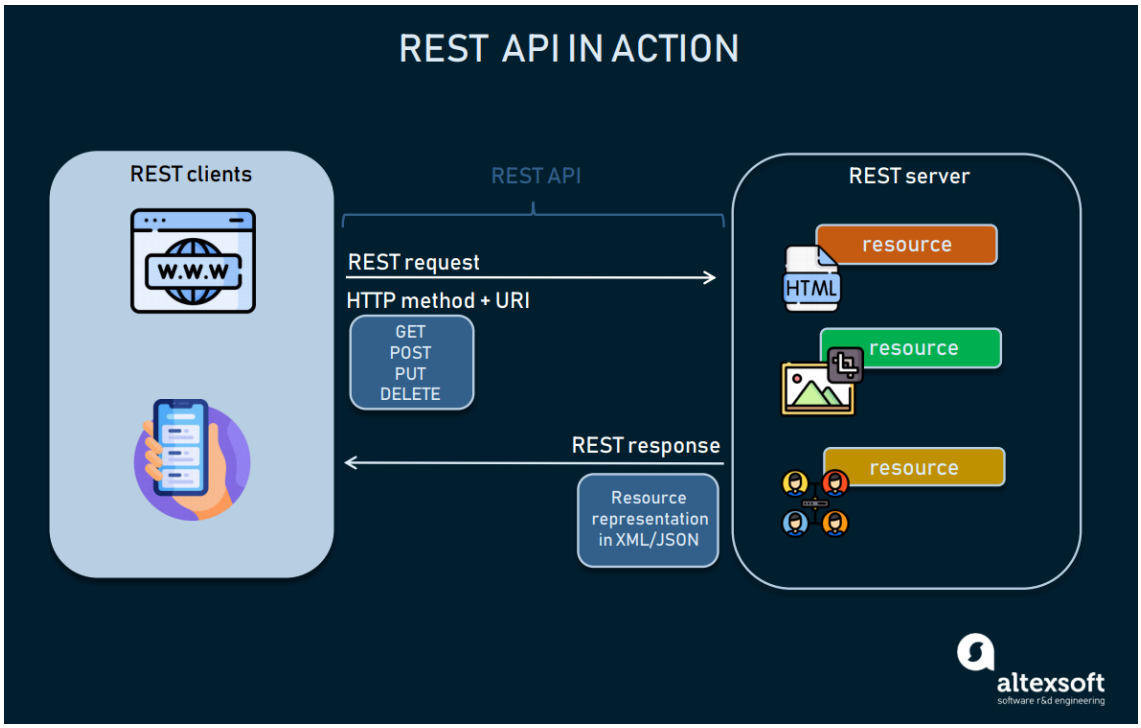
Opinnäytetyössä Pinja Solutions Oy:n oma pilvipalvelu, vastaanotettuaan tietojenhaku-kutsun, lähettää kutsun eteenpäin asiakkaalle pystytettyyn pilvipalveluun, joka hoitaa tietojen keräyksen ja lähetyksen takaisin Pinjan pilvipalvelulle, joka taas lähettää kerätyn

sanoman alkuperäisen tietojenhaku-kutsun lähettäjälle, mikäli kutsujalla ja asiakkaan palvelimella kaikki oikeudet ovat kunnossa. Kaikki tiedonsiirto tapahtuu salatun HTTPS-yhteyden avulla tiettyjen käyttäjätunnusten sekä autentikoinnin alla.

2.1 REST API

REST tulee sanoista Representational State Transfer. Se on HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmointirajapintojen eli APIen toteuttamiseen. API tulee sanoista Application Programming Interface ja tarkoittaa rajaa komponenttien tai moduulien välillä ohjelmoitavassa järjestelmässä, jonka kautta ne voivat tehdä pyyntöjä ja vaihtaa tietoja keskenään. Tästä hyvänä esimerkkinä on käyttöjärjestelmän rajapinta, jolla ohjelmat voivat käyttää keskusmuistia sekä tiedostoja. REST-ohjelmointirajapintojen avulla voidaan lähettää pyyntöjä HTTP- tai HTTPS-protokollan ylitse toiselle palvelimelle, jotka palauttavat dataa esimerkiksi XML- tai JSON-muodossa. (Wikipedia 2022d.)

REST-rajapinnan kanssa käytetään usein HTTP-metodeja, joiden avulla tietoja pystytään hakemaan GET-metodilla, päivittämään PUT-metodilla, luomaan POST-metodilla ja poistamaan DELETE-metodilla. PUT-metodin lisäksi on päivittämistä varten olemassa myös PATCH, jonka avulla pystytään muokkaamaan pienempiä osia datasta koskematta jokaiseen tietokenttään. (Kuva 2.)



Kuva 2. REST API:n toiminta havainnollistettuna. (Altexsoft 2021.)

Opinnäytetyössä Pinja Solutionsin API ottaa vastaan GET-metodin kutsun URL:n kautta, johon sisällytetään parametrit tietojenhakua varten. Nämä parametrit välitetään eteenpäin asiakkaan rajapintaan, joka hoitaa datahaun ja käsittelyn sekä lokittaa käsitellyt tiedot ja palauttaa XML-sanoman takaisin Pinja Solutionsin rajapintaan, joka palauttaa sen alkuperäiselle kutsujalle. Projektin kehityksessä käytettiin rajapinnan testaamiseen Fiddler 4 -työkalua, joka osoittautui erittäin päteväksi.

2.2 Autorisointi

Lähes jokainen REST API käyttää jonkinlaista todennusta. Todennus on prosessi, jolla varmennetaan rekisteröidyn käyttäjän tai prosessin henkilöllisyys ennen suojattujen verkkojen ja järjestelmien käytön mahdollistamista. Valtuutus on yksityiskohtaisempi prosessi, joka vahvistaa, että todennetulle käyttäjälle tai prosessille on myönnetty lupa päästä käyttämään tiettyä pyydettyä resurssia. Todennusprosessi tulee aina ennen valtuutusprosessia. Vaikka nämä toteutetaan usein yhdessä, ne ovat kaksi erillistä toimintoa. Prosessia, jolla pääsy näihin resursseihin rajoitetaan tiettyyn määrään käyttäjiä, kutsutaan kulunvalvonnaksi. (Levin 2021.)

Todennuksen aikana käyttäjän antamia tunnistetietoja verrataan valtuutettujen käyttäjien tietokannassa oleviin tietoihin joko paikallisessa käyttöjärjestelmäpalvelimessa tai todennuspalvelimen kautta. Jos annetut tunnistetiedot vastaavat tiedostossa olevia ja todennetulla entiteetillä on oikeus käyttää resurssia, käyttäjälle myönnetään käyttöoikeus. (Levin 2021.)

Projektissa käytetään basic-autentikaatiota, joka on yksinkertainen autentikaatio-skeema rakennettuna HTTP-protokollaan, joka sisältää base64-enkoodatun merkkijonon muodossa “käyttäjätunnus:salasana”. Mikäli autentikaatio on kunnossa, käytetään base64-enkoodattua käyttäjätunnusta ja salasanaa, kun otetaan yhteyttä Pinjan rajapintaan.

3 C#-OLIO-OHJELMOINTI

C# on yhdysvaltalaisen Microsoft Corporationin kehittämä moderni olio-orientoitunut korkeamman tason ohjelmointikieli, jota käytetään ensisijaisesti Windows-pohjaisiin sovelluksiin. C#:n suunnitteluun ja kehittämiseen vaikuttivat monet muut olio-orientoituneet kielet kuten C++ ja Java. Kielenä C# on symbioottinen .NET-alustan kanssa. C#:lla ei itsellään ole esimerkiksi omia luokkakirjastoja, mutta toisaalta suurin osa .NET-luokkakirjastoista on toteutettu C#-kieltä käyttäen. Kielenä C# tukee jokaista kolmea Bjarne Stroustrupin mainitsemaa olio-ohjelmoinnin periaatetta: objekteja, luokkia ja periytymistä. Lisäksi se tukee vielä kapselointia ja monimuotoisuutta, joten se toteuttaa kaikki modernin olio-ohjelmoinnin tunnusmerkit. Kielen tarkoituksena oli luoda yksinkertainen, moderni ja yleiskäyttöinen olio-ohjelmointikieli, joka soveltuu sulautettujen järjestelmien sekä myös ylläpidettyjen sovellusten kehitykseen. (Wikipedia 2022a.)

Opinnäytetyön projekti on kirjoitettu käyttäen C#-ohjelmointikieltä, sillä ForestPro-toiminnanohjausjärjestelmän uusimmat ominaisuudet on toteutettu pääosin C#:lla ja vanhemmat ominaisuudet Visual Basicilla. Toiminnanohjausjärjestelmässä olioiden luokkarakenne ja varsinainen tietojenkäsittely on eriytetty eri projektien alle selkeyttämään ison kokonaisuuden rakennetta.

3.1 Tyypitys

C# on vahvasti tyypitetty kieli. Jokaisella muuttujalla on tyyppi kuten myös jokaisella lausekkeella, joka määrittää arvon. Jokainen metodin määrittely määrittää nimen, tyypin ja lajin kullekin parametrille ja palautusarvolle. Tyypillinen C#-ohjelma käyttää luokkakirjaston tyyppejä ja käyttäjän määrittämiä tyyppejä, jotka mallintavat ohjelman ongelma-alueelle ominaisia käsitteitä. Kääntäjä käyttää tyyppitietoja varmistaakseen, että kaikki koodissa suoritettavat toiminnot ovat tyyppiturvallisia. Jos esimerkiksi määritetään muuttuja, jonka tyyppi on Integer, kääntäjä antaa käyttää muuttujaa

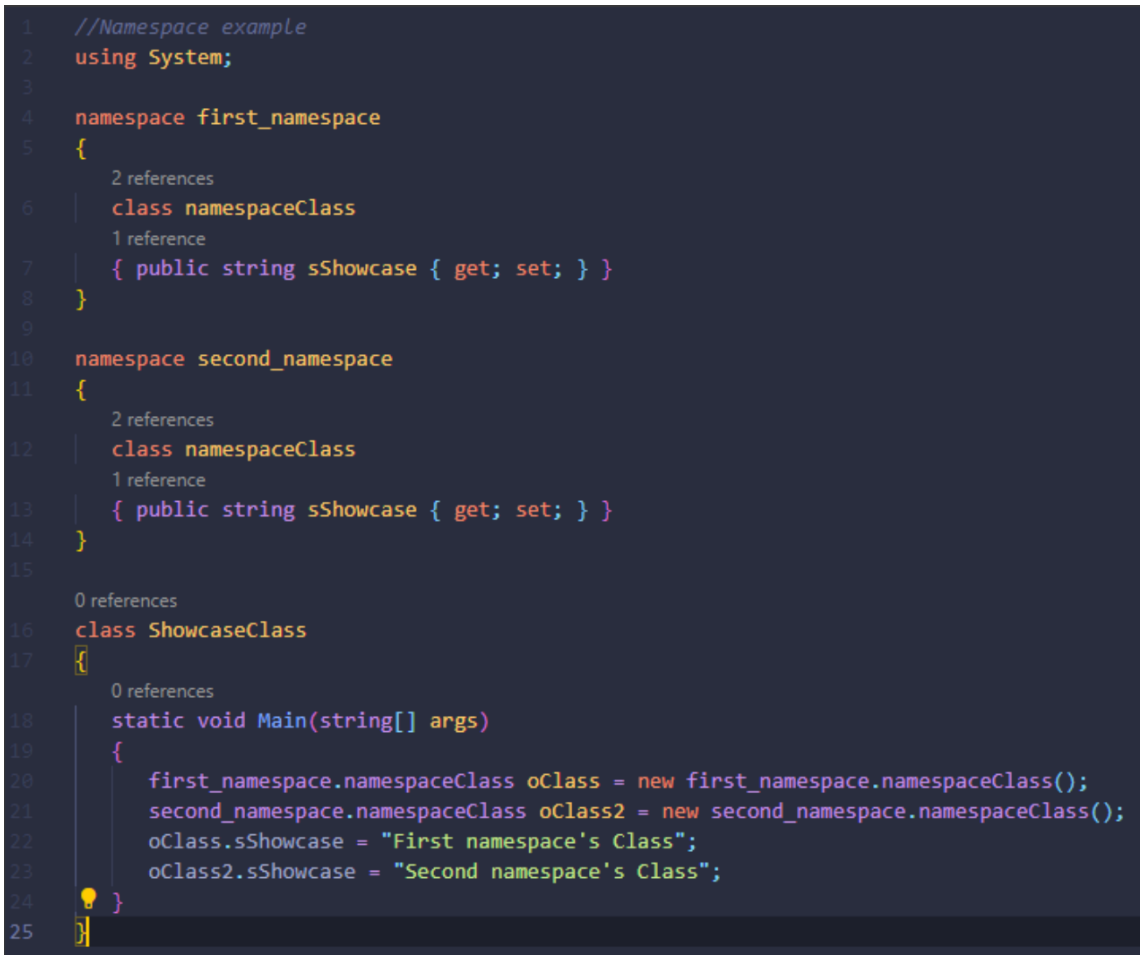
yhteen- ja vähennyslaskuoperaatioissa. Jos yritetään suorittaa samat toiminnot muuttujalle, jonka tyyppi on Boolean, kääntäjä luo virheen. (Wikipedia 2022a.)

3.2 Nimiavaruus

Nimiavaruuksia käytetään runsaasti C#-kielessä. Nimiavaruuksia käyttämällä voidaan hallinnoida luokkien, rajapintojen, enum-muuttujien ja delegaattien näkyvyyttä ja metodien nimiä isommissa projekteissa.

Kuvan 3 esimerkissä kahden **eri nimiavaruuden ansiosta niiden sisällä pystytään luomaan samanniminen luokka ilman, että ne sekottuvat keskenään,** sekä käyttämään niitä referoimalla oikeaa nimiavaruutta. (Kuva 3.)

ForestProssa nimiavaruudet ovat keskeisessä roolissa, sillä ohjelma on jaettu useampaan eri projektiin, joissa usein käytetään yhden ominaisuuden kohdalla samaa nimitystä eri luokalle monessa eri nimiavaruudessa, jotka kuitenkin liittyvät toisiinsa tavalla tai toisella.



```
//Namespace example
using System;

namespace first_namespace
{
    2 references
    class namespaceClass
    1 reference
    { public string sShowcase { get; set; } }
}

namespace second_namespace
{
    2 references
    class namespaceClass
    1 reference
    { public string sShowcase { get; set; } }
}

0 references
class ShowcaseClass
{
    0 references
    static void Main(string[] args)
    {
        first_namespace.namespaceClass oClass = new first_namespace.namespaceClass();
        second_namespace.namespaceClass oClass2 = new second_namespace.namespaceClass();
        oClass.sShowcase = "First namespace's Class";
        oClass2.sShowcase = "Second namespace's Class";
    }
}
```

Kuva 3. Esimerkki nimiavaruuksista.

3.3 Periytyminen

C# ei tue useampaa periytystä kerralla. Luokka pystyy siis perimään ainoastaan yhden luokan kerrallaan, mutta voi kuitenkin toteuttaa lukuisia rajapintoja eli täysin abstrakteja luokkia. Tämä oli johtavan arkkitehdin suunnittelupäätös komplikaatioiden välttämiseksi. Toteutettaessa useita käyttöliittymiä, jotka sisältävät samannimisen menetelmän ja jotka ottavat vastaan saman tyyppiset parametrit samassa järjestyksessä, C# sallii yhden menetelmän kattamaan kaikki liitännät. (Wikipedia 2022a.)

4 LINQ-KOMPONENTTI

LINQusta puhuttaessa tarkoitetaan lyhennettä “Language-Integrated Query” eli kieli-integroitua kyselyominaisuutta, jonka avulla pystytään käsittelemään esimerkiksi luokkia tai tietotauluja. Se tarjoaa yleisen syntaksin minkä tahansa tyyppisen tietolähteen kyselyyn. LINQ helpottaa myös kehitystyötä, sillä se sisällyttää älykkään koodin täydennys- ja virheenkorjaustuen jo koodin käännöshetkellä kokoonpanoonsa. LINQ on vahvasti tyypitetty. Käännöshetkellä varmistetaan arvojen käsittely oikean tyyppisinä, jotta vältytään virheiltä ajon aikana. (Microsoft 2021b.)

4.1 LINQ-metodit

LINQussa eniten käytettyjä metodeja ovat First, FirstOrDefault, Any, Select, Where, OrderBy, ToList ja Join, joista lisätietoa seuraavissa luvuissa.

4.1.1 First ja FirstOrDefault

First-metodilla voidaan valita ja palauttaa listan ensimmäinen objekti suoraan tai sen predikaattiin voidaan sisältää ehtolauseke, jonka avulla voidaan valita listan ensimmäinen objekti, joka täyttää annetut ehdot.

Kuvan 4 esimerkissä on kuvitteellinen lista, joka sisältää kokonaislukuja. Listasta valitaan ensimmäinen objekti, jonka Arvo1 vastaa annetun objektin Arvo1:tä. (Kuva 4.)

FirstOrDefault-metodi on toiminnaltaan lähes samanlainen, mutta tilanteessa, jossa tämä kuvitteellinen lista ei sisällä arvoja, jotka vastaavat annettua ehtoa, se osaa käsitellä ”virhetilanteen” ja palauttaa tällöin oletusarvon.

```
1   var objekti = new objekti();
2   lista.First(x => x.Arvo1 == objekti.Arvo1);
```

Kuva 4. Esimerkki First-metodin käytöstä.

4.1.2 Any

Metodilla Any tarkistetaan tästä kuvitteellisesta listasta, täsmääkö yksikään listan arvoista sille syötettyä vertailuarvoa. Mikäli lista sisältää luvun 3, palautetaan Boolean-arvo True, mikäli ei sisällä niin palautetaan Boolean arvo False.

```
3    lista.Any(x => x == 3);
```

Kuva 5. Esimerkki Any-metodista.

4.1.3 Select

Select-metodia käytetään usein, kun halutaan valita arvoja jostain tietolähteestä. Esimerkiksi jos halutaan valita tietoja datatablesta ja luoda niistä tiedoista uusia objekteja. Ehtolausekkeita voi olla yksi tai useampi.

4.1.4 Where

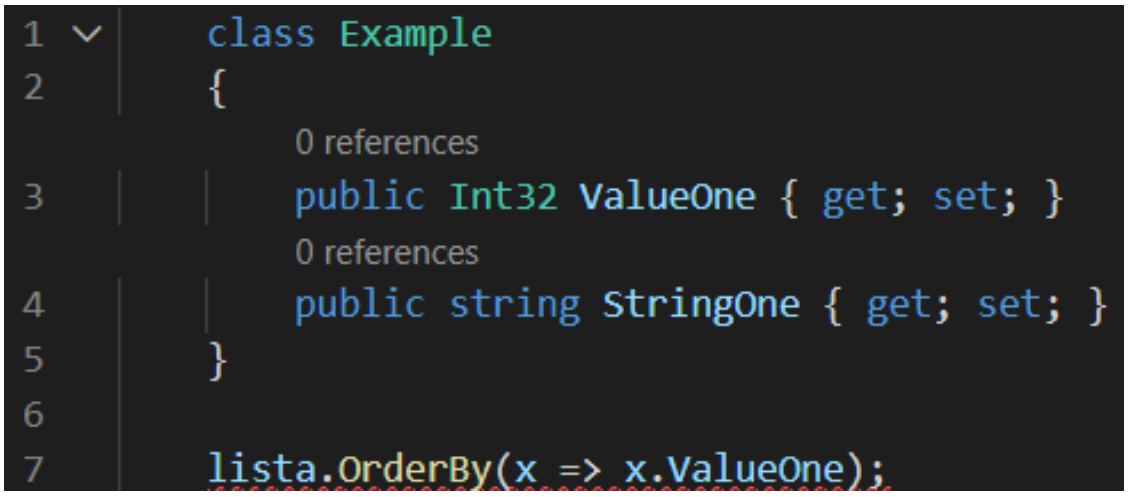
Where-metodin avulla pystytään palauttamaan listan kaikki objektit, jotka vastaavat annettuja ehtoja.

```
lista.Where(x => x.Arvo1 == objekti.Arvo1);
```

Kuva 6. Esimerkki Where-metodista.

4.1.5 OrderBy

OrderBy-metodin avulla voidaan järjestellä lista objekteja uuteen järjestykseen. Jos lista sisältäisi objekteja, joiden propertyissä olisi kokonaisluku, listan objektit voidaan järjestää kokonaisluvun perusteella pienimmästä suurempaan.



Kuva 7. Esimerkki OrderBy-metodista.

4.1.6 ToList

ToList-menetelmä ottaa IEnumerable-tyypin ja muuntaa sen List-tyypiksi. Tämä muunnos on tärkeä tapauksissa, joissa haluamme käyttää konkreettista tyyppiä "Lista", abstraktin "IEnumerable" sijaan. ToList-metodi on LINQ:n yksi käytetyimmistä metodeista. Kun halutaan valita Select-metodilla listasta objekteja ja niiden objektien arvoja ja luoda niistä lista uusia objekteja käyttäen vanhojen objektien arvoja, joudutaan muuttamaan nämä abstraktit IEnumerablet lista-muotoon.

4.1.7 Join

Join-metodia käytetään usein, kun halutaan käyttää kahden eri listan objekteja, jotka sisältävät yhden tai useamman saman arvon, joiden avulla kyseiset listat pystytään linkittämään toisiinsa. Näitä linkitysarvoja kutsutaan nimellä “Keys”, eli avaimiksi.

4.2 LINQun käyttö projektissa

Tässä esimerkissä on haettu työmaan tiedot kahdella eri SQL-kyselyllä tietokannasta. Kuvassa 8 ryhmitellään ensimmäisen tietotaulun tiedot KuvioID:n ja työmaaohjeen ID:n perusteella. Tämän jälkeen liitetään toisen tietotaulun tiedot KuvioID:n mukaan ensimmäiseen tietotauluun. Sen jälkeen luodaan muuttuja joka käyttää tietoja tästä

lopullisesta ryhmittely liitoksesta ja tekee niistä uuden Shape-objektin listaan Selectin avulla, ottamalla liitoksen ensimmäisen objektin arvon First-metodilla määritellyn kentän nimen perusteella. Tämä shape-objekti on pieni osa lähetettyä kokonaisuutta, mutta esittää hyvin lähes jokaisen käytetyimmistä LINQ-metodeista käytännössä. (Kuva 8.)

```
lstWorkingSites = (from DataRow dr in dt.Rows
                   group dr by new
                   {
                       StandID = dr.Field<Int32>("KuvioID"),
                       WorkInstructionID = dr.Field<Int32>("ID")
                   } into gWorkInstruction
                   join DataRow drCompartment in dtCompartment.Rows
                   on gWorkInstruction.FirstOrDefault().Field<Int32>("KuvioID")
                   equals drCompartment.Field<Int32>("KuvioID")
                   into gWorkInstructionCompartment
                   let Shapes = (from drShape in gWorkInstructionCompartment
                   group drShape by new { CompartmentID = drShape.Field<Int32>("CompartmentID") }
                   into gCompartment
                   select new Planner_Core.Classes.Schemas.FinnHarvest.FinnHarvest.Shape
                   {
                       ShapeNumber = gCompartment.First().Field<string>("CompartmentNumber"),
                       Area = gCompartment.First().Field<decimal>("Area"),
                       Points = GetShapePoints(gCompartment.First().Field<string>("MapShapeDataPointsXML"))
                   }).ToList()
```

Kuva 8. LINQ-ryhmittely projektissa.

5 .NET FRAMEWORK

.NET Framework on Microsoft Corporationin kehittämä ohjelmistokomponenttikirjasto, joka tukee noin 20 eri ohjelmointikieltä, joista käytetyimpiä ovat Microsoftin kehittämät C# ja VB.Net. NET Framework koostuu kahdesta osasta: ajonaikaisesta ympäristöstä ja luokkakirjastoista.

.NET Frameworkille kirjoitetut ohjelmat käännetään Common Intermediate Language-koodiksi sen sijaan, että ne käännetään suoraan konekoodiksi. Suorituksen aikana arkkitehtuurikohtainen just-in-time-kääntäjä muuttaa Common Intermediate Language -koodin konekoodiksi. Tämä yhteinen kieli-infrastruktuuri tarjoaa kielineutraalin alustan sovellusten kehittämiselle ja suorittamiseen. Kun .NET Frameworkin ydinominaisuudet otetaan käyttöön Common Intermediate Languagen puitteissa, näitä toimintoja ei sidota yhteen kieleen, vaan ne ovat saatavilla useilla puitteen tukemilla kielillä. (Wikipedia 2022e.)

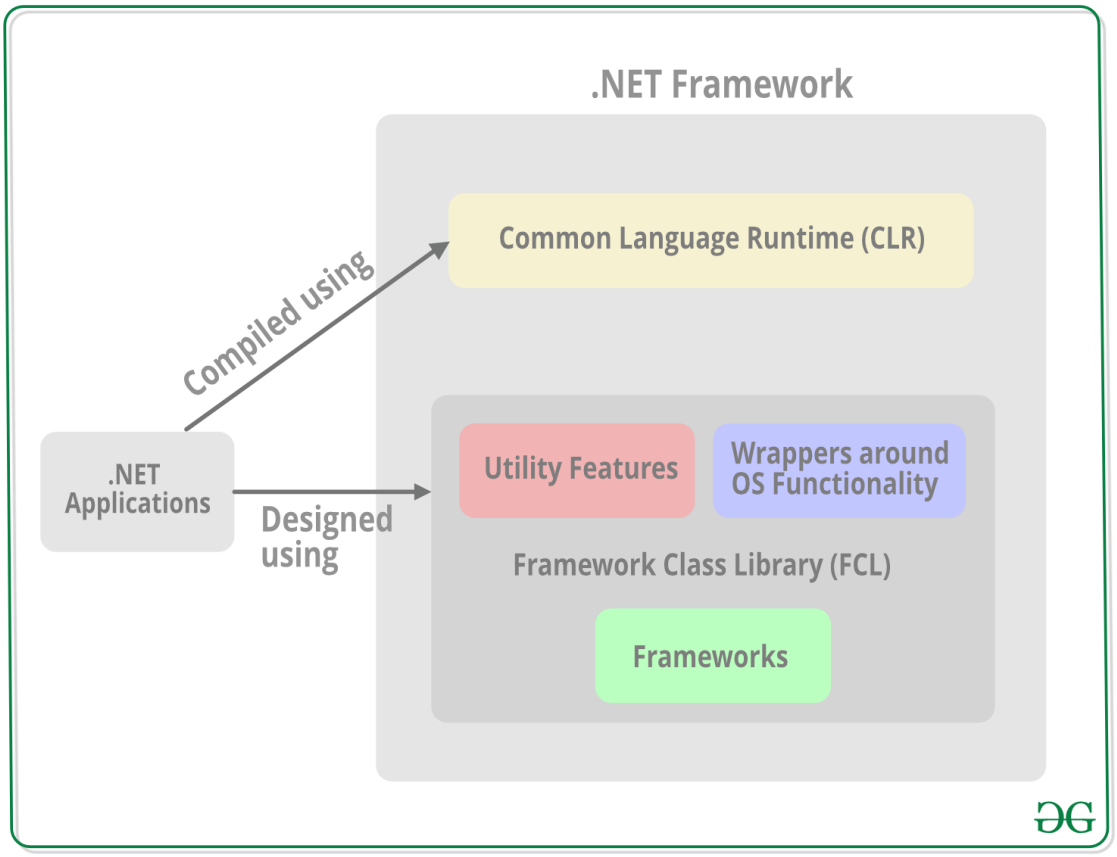
5.1 Common Language Runtime

Common Language Runtime toimii .NET Frameworkin suoritusmoottorina ja tarjoaa monia palveluita, kuten muistinhallinnan, poikkeusten käsittelyn, roskienkeruun, suojauksen ja säikeiden hallinnan. Kaikki .NET Frameworkille kirjoitetut ohjelmat suoritetaan Common Language Runtimen toimesta ohjelmointikielestä riippumatta. (Microsoft 2021a.)

5.2 Framework Class Library

.NET Framework sisältää CLI-perustaisten standardikirjastojen toteutuksen. Framework Class Library on järjestetty hierarkkiseen puurakenteeseen ja se on jaettu nimiavaruuksiin. Suurin osa sisäänrakennetuista sovellusliittymistä on joko System- tai Microsoft-nimiavaruutta. Nämä luokkakirjastot toteuttavat monia yleisiä toimintoja,

kuten tiedostojen lukemisen ja kirjoittamisen, graafisen renderöinnin, tietokanta-vuorovaikutuksen ja XML-dokumenttien käsittelyn. Luokkakirjastot ovat saatavilla kaikille CLI-yhteensopiville kielille. (Microsoft 2015d.)

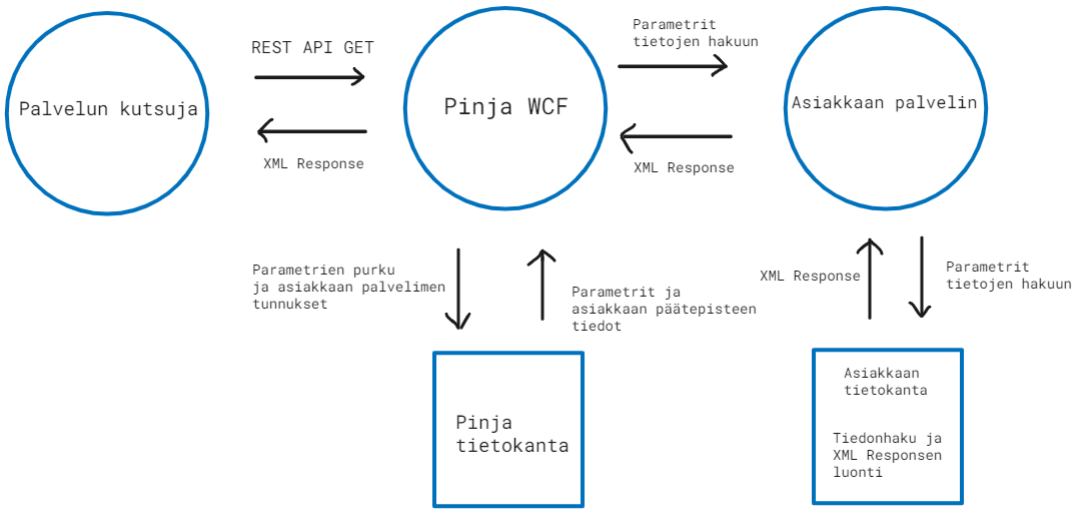


Kuva 9. .Net Frameworkin toiminta havainnollistettuna. (Geeksforgeeks 2021.)

6 WINDOWS COMMUNICATION FOUNDATION

WCF eli Windows Communication Foundation on avoimessa lähdekoodissa oleva ajonaikainen järjestelmä .NET Frameworkissa, joka soveltuu palvelukeskeisten yhdistettyjen sovellusten rakentamiseen. WCF avulla voidaan lähettää tietoa asynkronisesti yhdestä päätepisteestä toiseen. Palvelun päätepiste voi olla osana jatkuvasti saatavilla olevaa palvelua, jota ylläpidetään Internet Information Servicesin toimesta tai päätepiste voi olla myös palvelun asiakas, joka pyytää tietoja toisen palvelun päätepisteestä. WCF on suunniteltu tarjoamaan hallittavissa oleva lähestymistapa verkkopalvelujen luomiseen. (Microsoft 2021c.)

Pinjan pääasiallinen tietokanta pyörittää useampaa palvelua jatkuvasti käyttämällä Internet Information Serviceä, jonne myös tämän opinnäytetyön aiheena oleva työmaa- ja urakkatilitystietojen rajapintapalvelu asennettiin. Projektin testaaminen kehitysvaiheessa tapahtui pyörittämällä kahta eri palvelun päätepistettä lokaalisti työkoneella, jotka kuvastivat Pinjan omaa päätepistettä ja asiakkaan päätepistettä.



Kuva 10. WCF-palvelujen toiminta projektissa havainnollistettuna.

7 SQL

SQL:stä puhuttaessa tarkoitetaan “Structured Query Language” eli strukturoitua kyselykieltä, jota käytetään relaatiotietokantojen datan käsittelyyn. SQL:ää käyttävät sekä tietokannan ylläpitäjät että ohjelmistokehittäjät, jotka haluavat määrittää ja suorittaa kyselyitä. Kyselyillä voidaan esimerkiksi muokata tietokannan taulua ja indeksirakenteita tai lisätä, päivittää ja poistaa datarivejä. (Sirkin 2020.)

Relaatiotietokannat, jotka tunnetaan myös nimellä SQL-tietokannat, sisältävät joukon taulukoita, jotka sisältävät tietoja riveissä ja sarakkeissa. Jokainen taulukon sarake vastaa tietoluokkaa, esimerkiksi asiakkaan nimeä tai osoitetta, kun taas jokainen rivi sisältää data-arvon risteävälle sarakkeelle.

Projektin kaikki haettavat työmaatiedot ja urakkatilitysten tiedot löytyvät asiakkaan tietokannasta, jota käsitellään pääsääntöisesti Microsoft SQL Server-hallintajärjestelmällä. SQL-kyselyitä käytetään tietojen hakemiseen kannasta. Kyselyt sisältävät samoja metodeja kuin LINQ-kappaleen tietojen käsittely, kuten select, where, join ja orderby.

8 POHDINTA

Työmaa- ja urakkatilitystietojen rajapintahaun toteutus onnistui pääosin hyvin, pieniä korjauksia lukuunottamatta. Vuoden 2022 helmikuun loppuun mennessä ominaisuus saatiin vietyä tuotantoon testattuna ja viimeisteltynä.

Opinnäytetyössä suurin osa käytetyistä teknologioista oli tuttuja jo ennestään, lukuunottamatta REST API-ominaisuutta. Aikaisempien hyvin samantapaisten ominaisuuksien kehittämisestä oli paljon apua kokonaisuuden hahmottamisessa, joka mahdollisti uuden toteutustavan opettelun hyvin nopeasti. Vaikka teknologiat olivatkin tuttuja, oli toteutustavoissa edelleen parannettavaa, kun kyseessä oli valtava määrä tietoa, joka tuli hakea tietokannasta. Parhaimpana esimerkkinä oli LINQ-metodien käyttäminen for each loopin sijaan, joka teki datan käsittelystä paljon tehokkaampaa sekä myös helpompaa.

Toimialan tunteminen on hyvin iso osa ohjelmoijan tietotaitoa ja se näkyi myös tässä projektissa. Tarvitsin apua työkaveriltani työmaatietojen hakuun tietokantataulujen linkityksissä, niiden runsaan määrän takia sekä varmistaakseni, että tiedot tulevat oikein ja oikeasta taulusta. Varsinkin toimialan ollessa hyvin monimutkainen on rohkeus uskaltaa kysyä erittäin tärkeää. Olen itse ollut Pinjalla töissä tällä hetkellä hieman yli vuoden, ja vaikka ohjelmointitaidot olisivat kehittyneet jo omasta mielestä ammattilaisen tasolle, olen silti vielä kuvainnollisesti lasten kengissä toimialan tietämyksen osalta ja siksi tiimityöskentely on hyvin tärkeää sekä suuressa roolissa kehitystyötäni.

Opin itse erityisesti käytännön kautta, jolloin abstraktit käsitteet saattavat monesti jäädä ymmärtämättä. Tämä näkyy usein keskustelua käydessä, kun puhutaan tietystä toteutustavasta tai teknologiasta, jolloin en osaa suoranaisesti vastata, osaanko asiaa

vai en, vaikka käytännössä osaisinkin asian luoda. On tietysti sanottu, että jos et osaa selittää asiaa yksinkertaisesti, et ymmärrä sitä tarpeeksi hyvin. Opinnäytetyötä kirjoittaessani on tullut useita valaistumisia lukiessa käyttämieni teknologioiden kuvausta ja varsinaista käyttötarkoitusta sekä tapoja, miten niitä voisi käyttää. Tämän ansiosta nykyään osaisin jopa kertoa ulkopuoliselle henkilölle, mitä olen tekemässä. Olen ennen tämän opinnäytetyön kirjoittamista ajatellut sen olevan vain taakka, eikä työn dokumentoinnista olisi hyötyä, mutta olisi naiivia väittää, etteikö kertaus olisi opintojen äiti.

Olen kiitollinen tästä projektista, sillä se auttoi minua hahmottamaan oman osaamiseni rajoja sekä kehittymään ohjelmoijana parempaan suuntaan. Stressinhallintakyvyt pääsivät myös koitokselle, sillä projektin aikataulu oli kokonaisuuteen nähden tiukka, ottaen huomioon toisen projektin aikataulun sekä muut satunnaiset tehtävät. Projektin maaliin saattaminen aikataulun puitteissa nosti itseluottamustani selviytyä hankalistakin tilanteista myös tulevaisuudessa.

LÄHTEET

Altexsoft 2021. REST API: Key Concepts, Best Practices, and Benefits. Hakupäivä 16.12.2021 https://www.altexsoft.com/blog/rest-api-design/

Geeksforgeeks 2021. .NET Framework Class Library (FCL). Hakupäivä 26.12.2021 https://www.geeksforgeeks.org/net-framework-class-library-fcl/

Homework18 2021. Cloud Services. Hakupäivä 14.12.2021 https://home-work18.com/cloud-services.html

Kangasniemi, Hanna & Lintulahti, Matti 2017. Mikä on pilvipalvelu? Elisa. Hakupäivä 14.12.2021 https://elisa.fi/ideat/mika-on-pilvipalvelu/

Liimatta Aki 2021. Pilvipalvelut. Tietoevry. Hakupäivä 14.12.2021 https://www.tietoevry.com/fi/blogi/2021/05/pilvipalvelut-tieda-tarkeimmat-termit/

Levin, Guy 2016. RESTful API Authentication Basics. RestCase. Hakupäivä 16.12.2021 https://blog.restcase.com/restful-api-authentication-basics/

Microsoft 2021a. Common Language Runtime (CLR) overview. Hakupäivä 26.12.2021 https://docs.microsoft.com/en-us/dotnet/standard/clr

Microsoft 2021b. Language Integrated Query (LINQ) (C#). Hakupäivä 13.12.2021 https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/

Microsoft 2021c. What is Windows Communication Foundation. Hakupäivä 29.12.2021 https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf

Microsoft 2015d. .NET Framework Class Library. Hakupäivä 26.12.2021 https://docs.microsoft.com/en-us/previous-versions/gg145045(v=vs.110)?redirectedfrom=MSDN

Sirkin, Jessica 2020. SQL (Structured Query Language). TechTarget. Hakupäivä 29.12.2021 https://searchdatamanagement.techtarget.com/definition/SQL

Wikipedia 2022a. C Sharp (programming language). Hakupäivä 21.12.2021 https://en.wikipedia.org/wiki/C_Sharp_(programming_language)

Wikipedia 2022b. HTTPS.Hakupäivä 13.12.2021. https://en.wikipedia.org/wiki/HTTPS

Wikipedia 2022c. Hypertext Transfer Protocol. Hakupäivä 13.12.2021. https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Wikipedia 2022d. Representational state transfer. Hakupäivä 16.12.2021 https://en.wikipedia.org/wiki/Representational_state_transfer

Wikipedia 2022e. .NET Framework. Hakupäivä 26.12.2021 https://en.wikipedia.org/wiki/.NET_Framework