



Arttu Perämäki, Lilli Häyhänen, Linus Willner, Teemu Olkkonen

Laitekirjanpito -sovellus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknologian tutkinto-ohjelma

Toteutusdokumentti

10.10.2021

Sisällys

1	Johdanto	1
2	Projektin visio	1
3	Toiminnallisuuksien suunnittelu ja toteutus	3
4	Sovelluksen käyttöohje	5
5	Kuvaus projektin arkkitehtuurista	5
6	Yhteenveto	9

1 Johdanto

Tämän projektin tarkoituksena oli perehtyä ketterien ohjelmistotuotannon muotojen työkaluihin ja työskentelytapoihin projektin hallinnassa ja ryhmän toiminnan ohjaamisessa. Tarkemman tarkastelun kohteena oli erityisesti scrum-menetelmä.

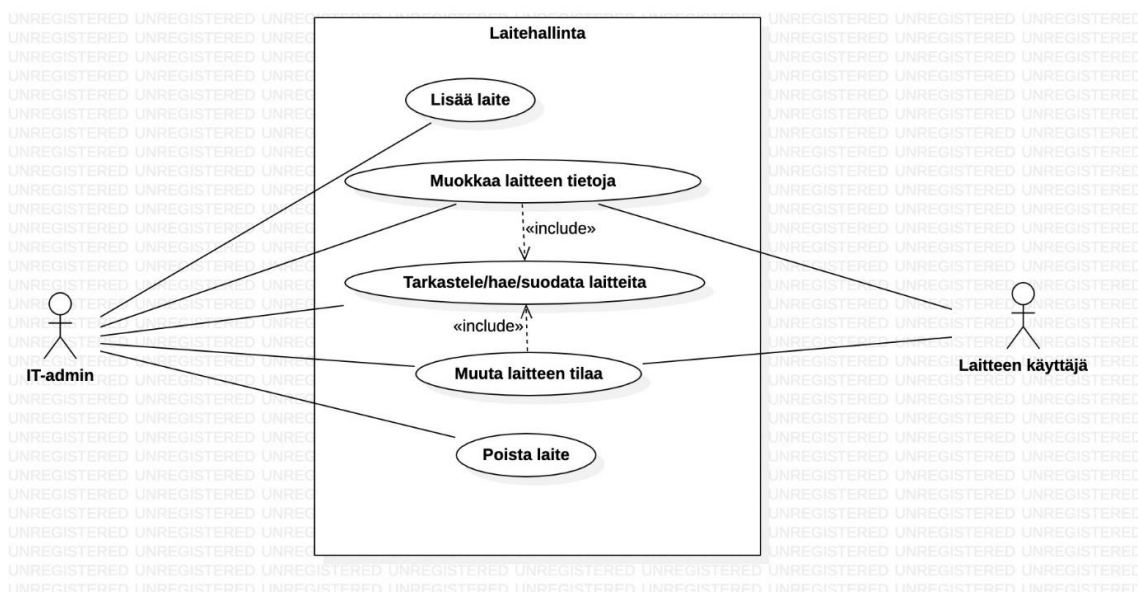
Projektityöskentely oli jaettu neljään kahden viikon mittaiseen sprinttiin, joista jokaisesta vastuussa oli yksi projektiryhmän jäsen kerrallaan, kukin vuorollaan. Ketteriin menetelmiin tutustumisen lisäksi projektin tarkoituksena on ollut tuottaa Java-ohjelmointikielellä työpöytäsovellus, jossa on JavaFX-kirjastolla toteutettu käyttöliittymä ja joka käyttää tietokantaa.

Projektin yhteydessä hyödynnettiin ja opeteltiin myös muita, osalle ryhmän jäsenistä uusia teknologioita, kuten Apache Maven, Docker, PostgreSQL, Hibernate ja Jenkins. Kehitystyön tehostamiseksi käytettiin myös Javan Lombok-kirjastoa. Projektin hallinnassa käytettiin Nektion -työkalua ja projektin mallintamiseen ja kaavioiden luomiseen käytettiin ERDPlus ja StarUML -mallinnustyökaluja.

Ryhmä päätti ryhtyä toteuttamaan sovellusta, jolla pystyy pitämään kirjaa tietoteknisistä päätelaitteista, niiden käyttäjistä ja tiloista, sekä lisäämään, poistamaan ja muokkaamaan laitteiden tietoja. Sovellus on tarkoitettu yritysten ja organisaatioiden IT-henkilöstön käyttöön tukemaan kenties useissa eri sijainneissa sijaitsevien päätelaitteiden tietojen tallennusta ja hallinnointia yhteen sovellukseen ja keskitettyyn tietokantaan sijainnista riippumatta.

2 Projektin visio

Projektin visiota määritellessä kohderyhmäksi valittiin yksityisen, julkisen ja kolmannen sektorin toimijat ja sovelluksen varsinaisina käyttäjinä näiden yhteisöjen tietoteknisistä ratkaisuksista ja kalustosta vastaavat henkilöt. Listattavien päätelaitteiden loppukäyttäjillä ei ole tarvetta päästä sovellukseen.



Kuva 1. Projektin käyttötapauskaavio.

Sovelluksen käyttötapaukset ovat esitettynä kuvassa 1 näkyvässä käyttötapauskaaviossa. Yrityksen IT-vastaava on sovelluksen ainoa tosiasiallinen käyttäjä. Laitteen käyttäjä ei käytä sovellusta vaan on mukana prosessissa ainoastaan vastaanottaessaan ja palauttaessaan koneen tai kun hänen tietonsa muuttuvat.

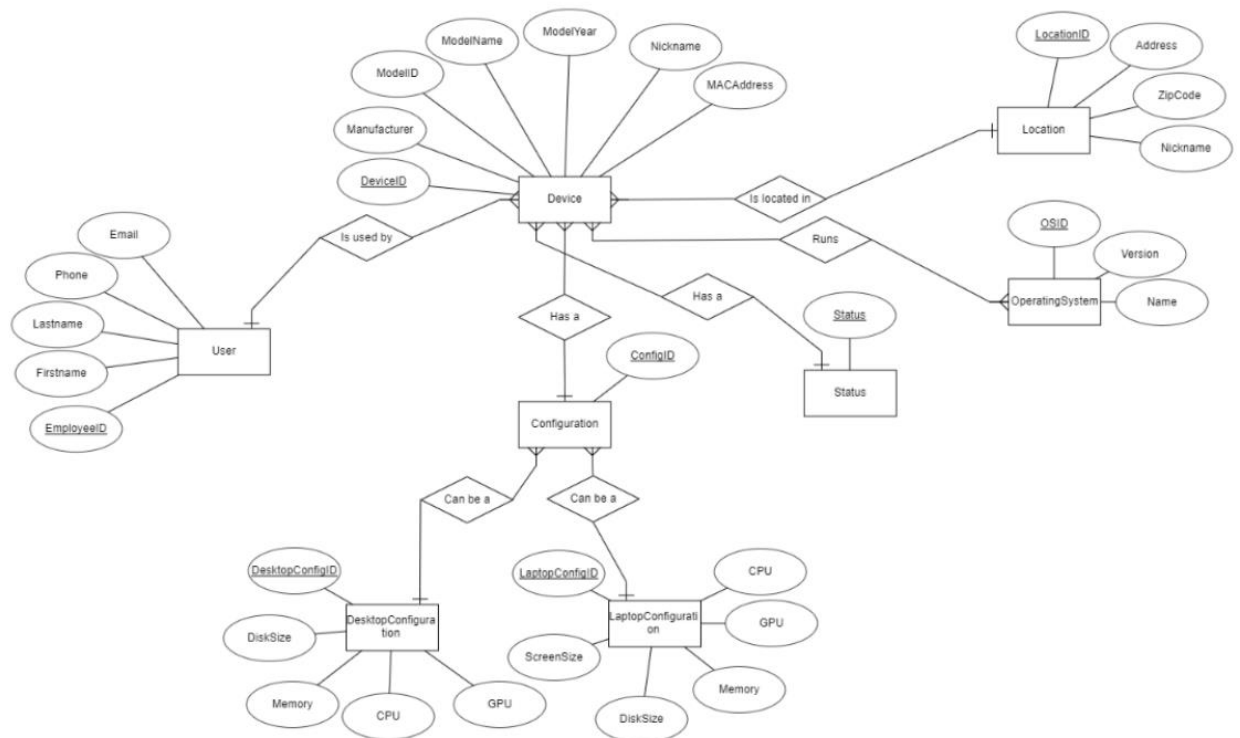
Laitteista vastaava henkilö voi lisätä laitteen, muokata sen tietoja, hakea laitteita tai tarkastella listaa laitteista, muuttaa laitteen tilaa ja poistaa laitteita. Käyttöliittymään päätettiin tehdä päänäköymä, joka listaa kaikki tietokannassa olemassa olevat laitteet tietoineen näytölle. Tietokantaan tallennetaan laitteen tekniset tiedot, tiedot käyttäjästä, toimipisteestä sekä laitteen aktiivinen statustieto, kuten onko laite jollakulla käytössä, vapaana, rikki tai huollettavana.

Teknisiä tietoja ovat muun muassa valmistaja- ja mallitiedot, tiedot laitteen muistista, prosessorista, näytönohjaimesta, käyttöjärjestelmästä ja se, onko kyseessä kannettava vai pöytätietokone. Päänäkymässä voi myös poistaa laitteita. Päänäkymän valikosta avautuu ponnahdusikkunoita, joissa voi lisätä uusia laitteita ja käyttäjiä tai muokata olemassa olevan laitteen tietoja.

Projektin suunnitteluun käytettiin aikaa ja opintojakson lähestyessä loppuaan on havaittavissa, että visiosta ei olla työskentelyn aikana juurikaan poikettu vaan visio on toiminut koko ajan työskentelyä ohjaavana suunnannäyttäjänä.

3 Toiminnallisuuden suunnittelu ja toteutus

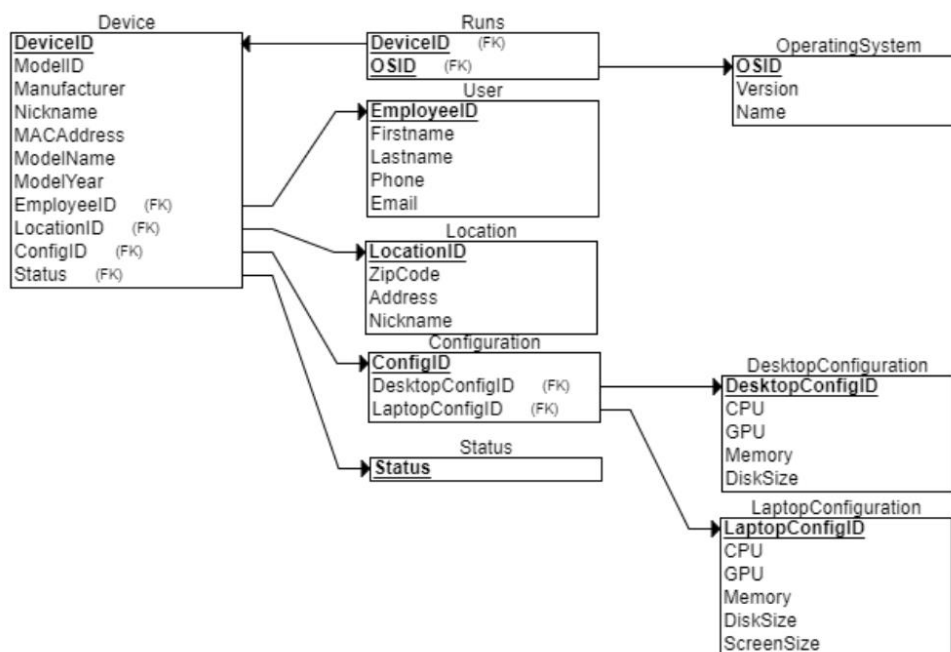
Toiminnallisuuden ja käyttöliittymän suunnittelu tehtiin yhteistyössä ja jaettiin projektinhallintajärjestelmä Nektioniin tarinoiksi ja tehtäviksi ja niiden tärkeyttä ja toteutusjärjestystä arvioitiin ja suunniteltiin pitkälti jo ennalta. Ennen sprinttien aloittamista suunniteltiin, mitä toiminnallisuuden kussakin sprintissä työskentetään.



Kuva 2. ER-kaavio ohjelmiston datasta.

Melko varhaisessa vaiheessa ohjelman tietokannan rakennetta ja ohjelmiston tarvitsemia tietoja lähdettiin suunnittelemaan ER-kaavion (Entity Relationship Model) ja relaatiotietokantakaavion luomisen avulla. Kuvan 2 ER-kaaviossa on kuvattuna, mitä kaikkia tietoja yksilötyypit Device, User, Configuration,

DesktopConfiguration, LaptopConfiguration, Status, Location ja OperatingSystem tarvitsevat ja millaisia ovat niiden väliset yhteydet. Alla olevassa kuvassa 3 näkyy ER-kaavion pohjalta laadittu relaatiotietokantakaavio. Kuvissa näkyvät käsitteet myös vakiintuivat kuvaamaan ohjelmiston eri osia projektin aikana.



Kuva 3. Projektin relaatiotietokantakaavio.

Tärkeimmiksi toiminnoiksi valikoituivat laitteiden listaaminen ja niiden lisääminen, poistaminen sekä muokkaaminen. Tietokannan rakenteen vuoksi myös esimerkiksi käyttäjien ja toimipisteiden tietoja pitää pystyä käsittelemään, jotta kokonaisuus olisi järkevä sovelluksen käyttötarkoituksen ja luonteen puolesta.

Joidenkin haasteiden vuoksi käyttöliittymän kehitystyössä ei päästy aina etene-mään sillä tahdilla, kuin oli alun perin suunniteltu ja toisinaan tehtäviä uudelleen järjesteltiin sen turvaamiseksi, että projektissa päästään siihen pisteeseen, että opintojakson lopuksi tärkeimpiä ominaisuuksia päästään esittelemään muille ryhmille.

Muita mahdollisia toiminnallisuuksia ovat muun muassa erilaiset laitteiden haku-, järjestely- ja suodatustoiminnot, joissa tietokantaa voidaan hyödyntää entistä

tehokkaammin. Näiden toimintojen toteuttaminen jäi vielä toistaiseksi tekemättä aikataulun toimiessa ryhmän kehitystyötä vastaan.

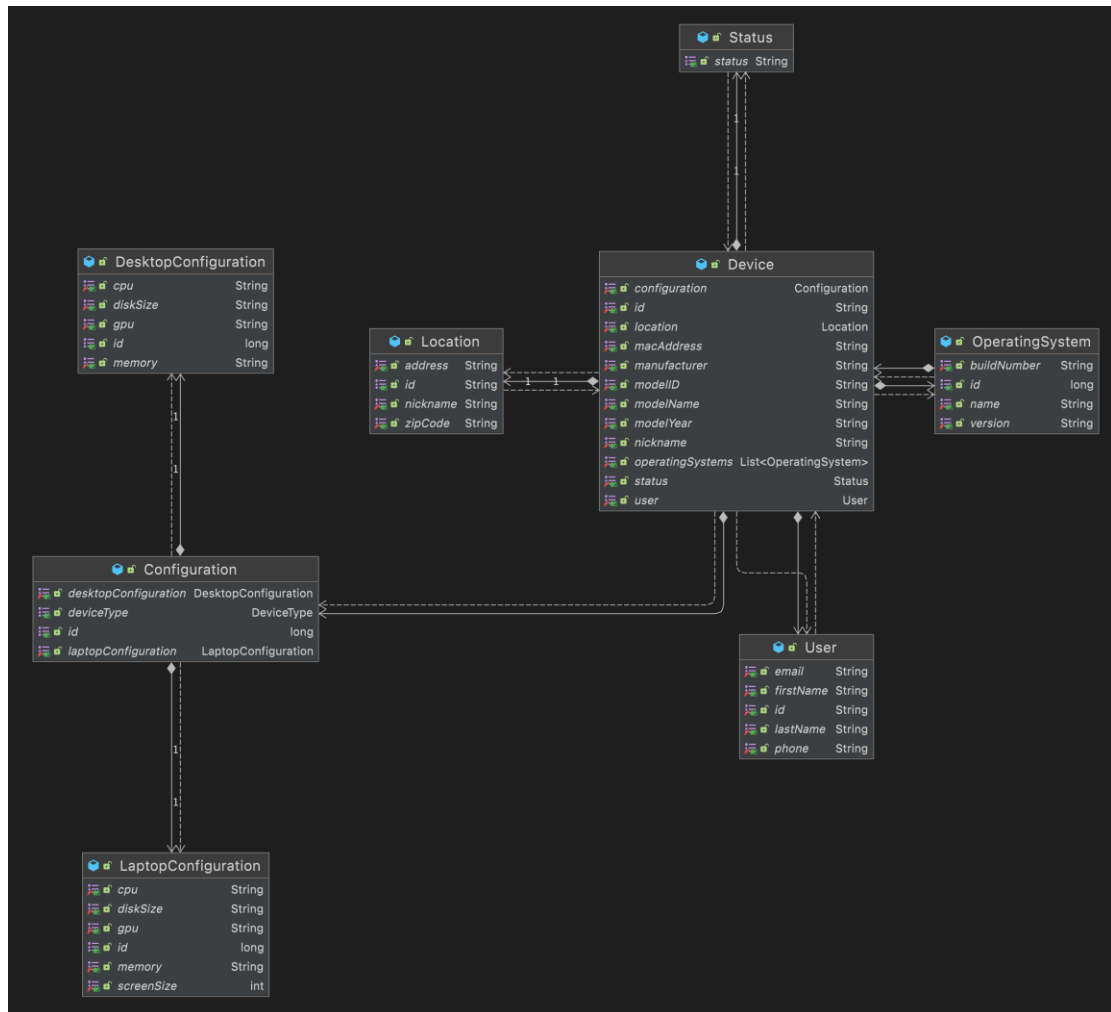
4 Sovelluksen käyttöohje

Kun käyttäjä avaa sovelluksen, aukeaa näkyviin suoraan listanäkymä, joka näyttää tietokannassa olemassa olevat laitteet tietoineen. Kullakin laitteen tietoja sisältävällä rivillä on nappi, jota painamalla käyttäjä voi valita, haluaako tarkastella yksittäisen laitteen tietoja tarkemmin, jolloin tiedot aukeavat uuteen näkymään. Vaihtoehtoisesti käyttäjä voi valita poistavansa laitteen.

Sovelluksen ylälaudassa on myös painike ”File”, josta aukeasta vetovalikosta käyttäjä voi valita haluaako lisätä uuden laitteen, konfiguraation, käyttäjän, toimipisteen, tai käyttöjärjestelmän tietokantaan. Näitä painamalla aukeaa valinnan mukainen ponnahdusikkuna, jossa tiedot voi syöttää niille varattuihin kenttiin ja tallentaa ne tietokantaan tai peruuttaa muutokset. Myös olemassa olevien laitteiden ja käyttäjien muokkaaminen onnistuu samanlaisista ikkunoista.

5 Kuvaus projektin arkkitehtuurista

Projektia mallinnettiin projektikurssin ohessa käynnissä olleella kurssilla Kuvaus- ja mallintamismenetelmät, jossa projektista, sen rakenteesta ja arkkitehtuurista tehtiin harjoitustöinä useita erilaisia kaaviokuvia ERDPlus ja StarUML työkaluilla. Alla on esiteltynä osa kurssilla aikaansaaduista tuotoksista.



Kuva 4. Projektin DTO-kokonaisuutta kuvaava luokkakaavio.

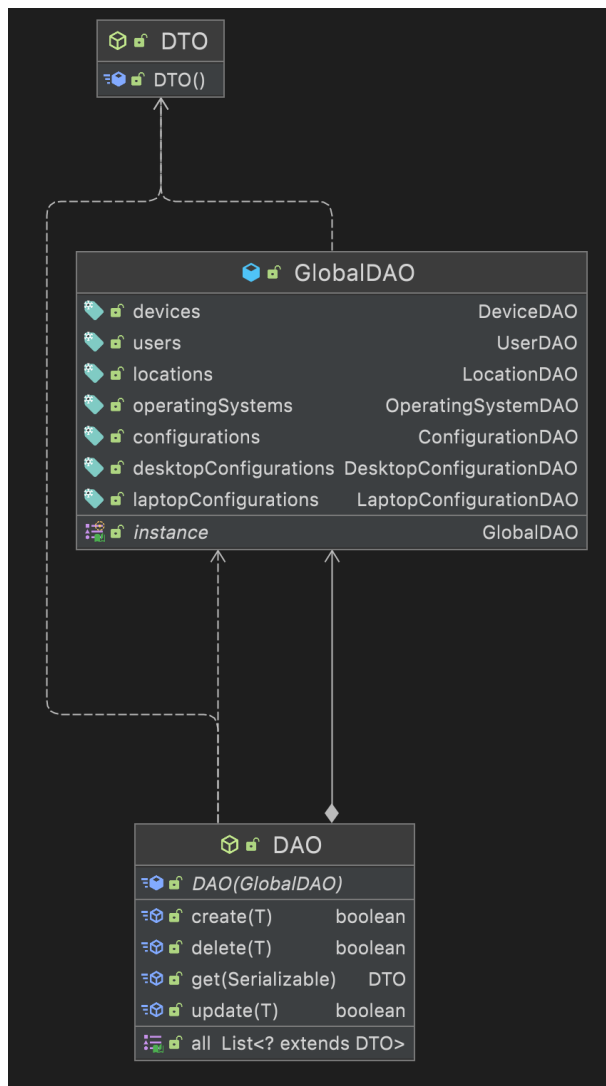
Luokkakaavio koostuu kolmesta kokonaisuudesta, jotka ovat DTO (Data Transfer Object) – joka näkyy kuvassa 4, kontrollerit ja DAO (Data Access Object).

DTO:n alla on kuvattuna luokka Device, jossa on laitteen perustiedot.

Luokan Configuration alla on aliluokat LaptopConfiguration ja DesktopConfiguration, jotka määrittelevät, mitä kaikkia teknisiä tietoja kuuluu kullekin laitetyyppille. Luokka Status kertoo laitteen tilan, onko se käytössä, vapaana, rikki tai huollossa.

OperatingSystem on oma luokkansa, sillä kahdella muuten keskenään samankaltaisella laitteella voi olla eri käyttöjärjestelmä tai yhdellä yksittäisellä laitteella voi olla useita eri käyttöjärjestelmiä. Location -luokka kertoo toimipisteen, jonka

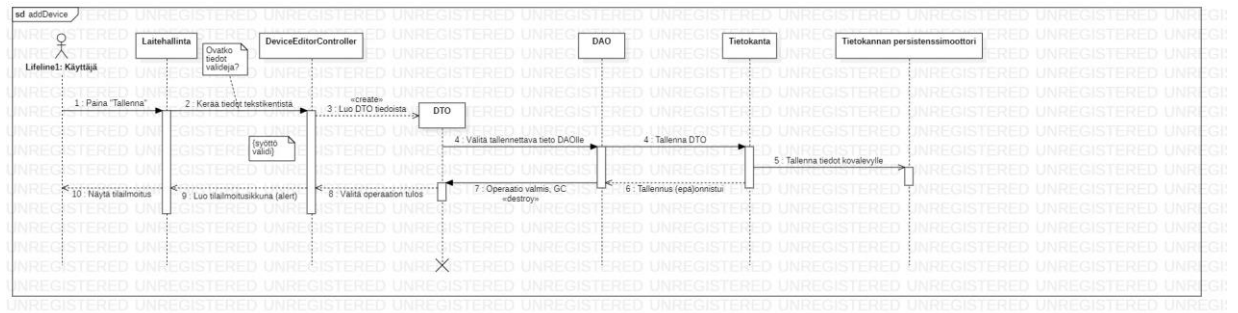
alle laite on rekisteröity. Luokka User pitää sisällään tarvittavat tiedot laitteen käyttäjästä.



Kuva 5. Projektin DAO-kokonaisuutta kuvaava luokkakaavio.

Kuvassa 5 näkyy, miten DAO:n alla on kuvattuna tietokannan rakenne ja sen eri taulut ja näiden sisällä olevat metodit ja operaatiot. Tietoja on pystyttävä hakemaan, lisäämään, päivittämään ja poistamaan.

Kontrolleriluokkia ovat DeviceListController, jonka tehtävänä mahdollistaa laitteiden listaus ja poistaminen sovelluksen käyttöliittymänäkymässä, sekä DeviceEditorController puolestaan huolehtii laitteiden lisäämisestä ja tietojen päivittämisestä.



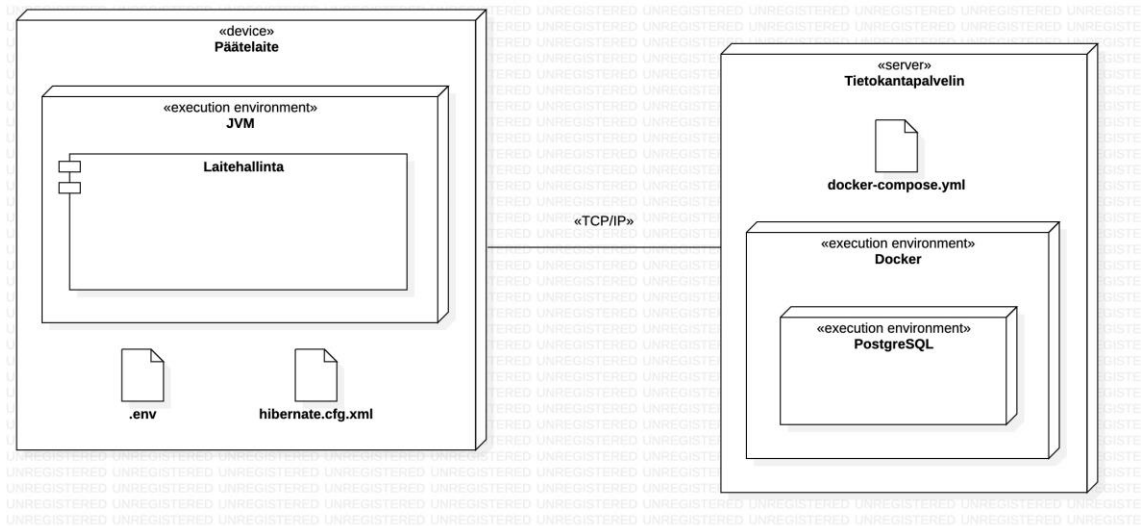
Kuva 6. Tallennusprosessia kuvaava sekvenssikaavio.

Kuvan 6 sekvenssikaaviossa kuvataan prosessia, kun käyttäjä klikkaa laitteen lisäysnäkylässä nappulaa "Tallenna". Tämän seurauksena tiedot kerätään syöttökentistä, ja niiden kelvollisuus tarkistetaan vielä kerran, vaikka ne on tarkistettu jo syöttövaiheessa.

Jos syöttö ei ole kelvollinen, prosessi katkeaa ja näyttää käyttäjälle virheilmoituksen. Jos syöttö on kelvollinen, kontrolleriluokka luo DTO:n keräämistään tiedoista, joka välitetään DAO:lle tallennusta varten.

DAO parsii DTO:n ja antaa tarvittavat SQL-käskyt tietokannalle. Tietokanta lisää tietojen muutokset muutosjonoon, ja ne tallennetaan kovalevylle asynkronisesti. Kun tietokanta ilmoittaa, että joko kaikki operaatiot onnistuivat, tai että jokin niistä epäonnistui, lähtee asiasta paluuviesti kontrollerille metodien paluuarvojen kautta.

DTO päättyy samassa prosessissa roskankeräykseen sen luovan funktion instanssin olemassaolon päättyessä. Kontrolleri ilmoittaa lopuksi tallennusoperaation onnistumisen tilan käyttäjälle.



Kuva 7. Projektin sijoittelukaavio.

Sijoittelukaavio kuvassa 7 kuvastaa ohjelmiston osien fyysisen sijainnin ja rakenteen valmiissa ohjelmistotuotteessa. Ohjelmistotuote pyörii käyttäjän päätelaitteella, jolla sijaitsevat myös ohjelmiston konfigurointitiedostot.

Suoritusympäristönä on JVM, jossa ohjelmistotuote sijaitsee. Lisäksi ohjelmisto tarvitsee pääsyn tietokantapalvelimelle, jossa pyörii PostgreSQL tietokanta Docker-kontin sisällä.

6 Yhteenveto

Projektin edetessä vastaan on tullut haasteita ja toisinaan aikatauluttaminen on tuottanut ryhmälle päänvaivaa. Visiossa ja suunnitelmissa pysyttiin hyvin ja ohjelmisto, käyttöliittymä tai rakenne eivät juurikaan muuttuneet matkan varrella.

Yhteistyö arvalla valitun ryhmän kanssa voi toisinaan luoda työskentelyyn omia haasteitaan itse valittuun ryhmään nähden. Haasteiden yli kuitenkin on päästy kohtuudella pitkin projektityöskentelyä, vaikka erimielisyyksiltä ei ollakaan täysin välttytty.

Varsinaisen sovelluksen kehitystyön lisäksi oli opintojakson tarkoituksena opetella projektinhallintaa scrum-menetelmällä ja uusien työkalujen hyödyntämistä.

Nektion projektinhallintatyökalun kankea käyttöliittymä ja sen päivittämiseen kulutettu aika aiheutti ryhmälle toisinaan harmaita hiuksia, samoin se, että kaikki sprintin yksityiskohdat eivät ole aina mitenkään ennakoitavissa ja on vaikea arvioida tiettyihin tehtäviin käytettävää aikaa.

Dokumentaatio ja sovellus kuitenkin saatiin hyvään vaiheeseen kurssin puitteissa ja myös Kuvaus- ja mallintamismenetelmien kurssilla tehdyt ryhmätyöt saatiin lopulta kunnialla tehtyä ja mallintamista päästiin harjoittelemaan osana projektityöskentelyä. Jatkokehitysideoiksi tulevaisuuteen jäävät ainakin erilaiset haku- ja suodatustoiminnot.