



Jonne Borgman, Henri Uimonen, Lassi Bågman, Rasmus Hyyppä

## Aniwash 2.0 -ohjelmisto

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknologian tutkinto-ohjelma

Toteutusdokumentti

5.5.2023

## Sisällys

1	Johdanto	1
2	Tuotteen vaatimustavoitteet	3
3	Käyttäjäroolit ja käyttötapaukset	4
4	Ohjelmiston tietomalli	5
5	Ohjelmiston rakenne	6
6	Ohjelmiston toiminta	7
7	Kehitysprosessi ja kehitysvaiheen tekniikat	10
8	Ohjelmiston jatkokehitys	11
9	Yhteenveto	12

# 1 Johdanto

Aniwash on ohjelmisto asiakashallintaan ja lemmikkieläinten hoidon ajoitukseen, jonka avulla työnantaja sekä työntekijät voivat seurata, lisätä, poistaa ja muokata tulevia tai menneitä varauksia. Ohjelma on tarkoitettu yrityksen työntekijöiden käyttöön, mutta siitä voidaan kehittää myös kuluttajille suunnattu varausalusta.

Nykyisessä versiossa ohjelmistossa on kaksi työntekijätyyppiä: työnantaja ja työntekijä. Työntekijällä on kyky luoda asiakkaita, luoda lemmikkejä ja osoittaa niitä asiakkaille, luoda uusia tuotteita ja palveluita sekä luoda uusia tapaamisia. Työnantajalla on samat kyvyt kuin työntekijällä, mutta hänellä on myös kyky luoda uusia työntekijöitä ja muokata työntekijöiden tietoja.

Aniwash-ohjelmistossa on mahdollisuus poistaa asiakkaita ja heidän lemmikkijään tarvittaessa. Työntekijä tai työnantaja voi poistaa asiakkaan tiedot ja lemmikin tiedot tietokannasta, kun asiakas ei enää tarvitse palveluita tai heidän asiakkuutensa on päättynyt. Tämä ominaisuus auttaa pitämään tietokannan siistinä ja ylläpitämään sen tehokkuutta.

Lisäksi ohjelmiston kautta pystyy poistamaan myös tulevia tai menneitä varauksia. Työntekijät ja työnantajat voivat tarkastella kaikkia varaustietoja, mutta vain heillä on oikeus poistaa varauksia tarvittaessa. Tämä ominaisuus auttaa pitämään ajanvarausjärjestelmän ajantasaisena ja helpottaa työntekijöiden ja asiakkaiden kommunikointia.

Sovelluksessa on sisäänkirjautumissivu, johon käyttäjä voi syöttää käyttäjätunnuksensa ja salasansansa. Jos käyttäjä ei ole rekisteröitynyt, työnantaja voi luoda uudelle työntekijälle tunnukset. Sovellus käyttää ulkoista tietokantaa, johon kaikki tiedot tallennetaan käyttäjätiedoista ajanvarauksiin asti. Tietokantaan voi luoda yhteyden samaan aikaan useammasta sovelluksesta.

Kaiken kaikkiaan Aniwash-ohjelmisto on erittäin monipuolinen ja helppokäyttöinen työkalu lemmikkieläinten hoitoon ja asiakashallintaan. Ohjelmiston avulla työnantajat ja työntekijät voivat helposti seurata ja hallita varauksia ja asiakastietoja, mikä helpottaa yrityksen päivittäistä toimintaa.

Ohjelmisto on kirjoitettu Java-kielellä ja käyttää JavaFX-kirjastoa graafisen käyttöliittymän toteutuksessa. Ohjelmistoa voi käyttää vain graafisen käyttöliittymän kautta, eikä esimerkiksi komentoriviltä. Ohjelmisto vaatii toimiakseen Java ja MySQL asennuksen.

## 2 Tuotteen vaatimustavoitteet

Täyden palvelun asiakashallintajärjestelmä ajanvarauksella ja työntekijöiden kalenterin hallinnalla. Hallitse helposti asiakkaiden, sekä heidän lemmikkiensä tietoja. Analysoi työntekijöiden tehokkuutta ja karsi turhia kuluja samalla pitäen kirjaa asiakasryhmistäsi.

Tuotteen avaintarkoitus: Asiakkaan luonti, eläimen lisäys asiakkaaseen, asiakas haku, ajanvaraus asiakkaalle, ostoskori tuotteille ajanvarauksessa, ostoskoriin lisäävät tuotteet, vanhojen merkintöjen ja asiakkuuksien poisto.

Asiakkaat: Eläin- pesulat tai trimmaamot

Tarve: Asiakashallinta järjestelmä ajanvarauksineen

Kilpailija: Timma.fi.

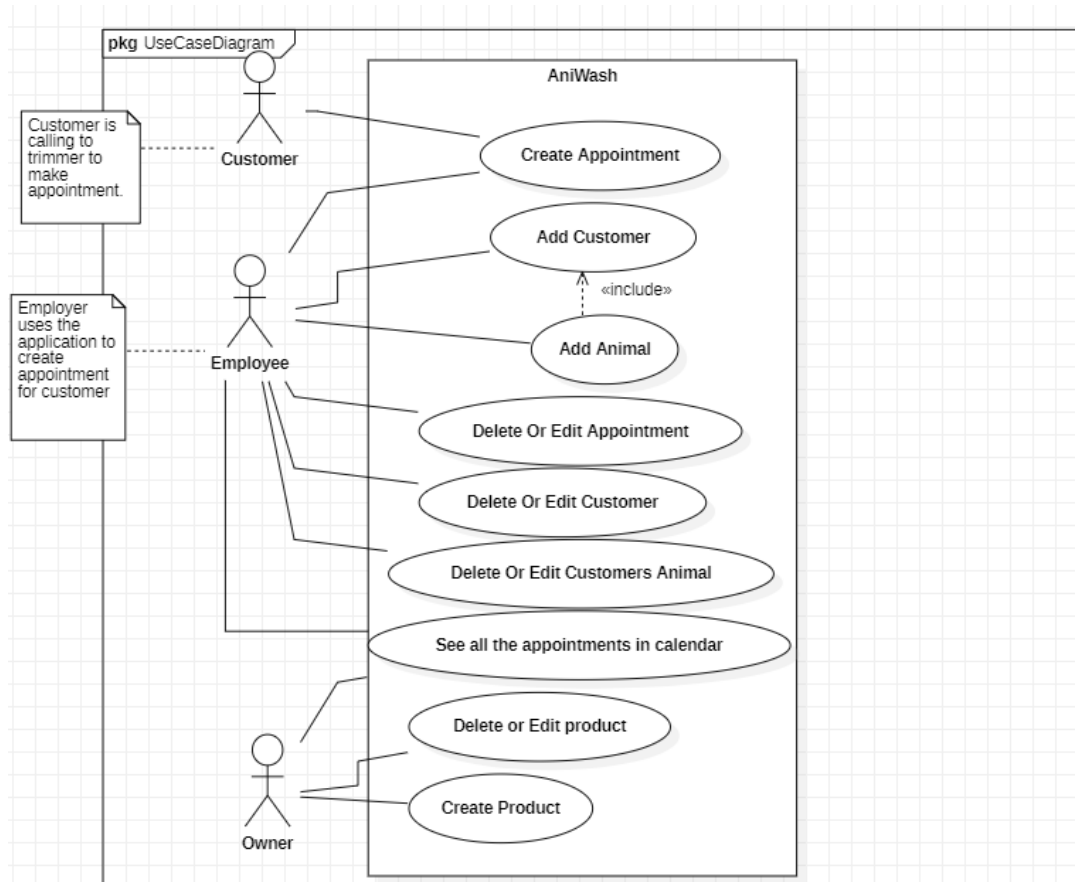
Primaari differentiaatio: Soveltuu nimenomaan eläimille palveluita tuottaville yrityksille.

Kuka: Eläinpesulan työntekijä

Mitä: Hallinnoi asiakkaita ja heidän eläimiään

Miksi: Mahdollistaa ajanvarauksen sekä mainostamisen

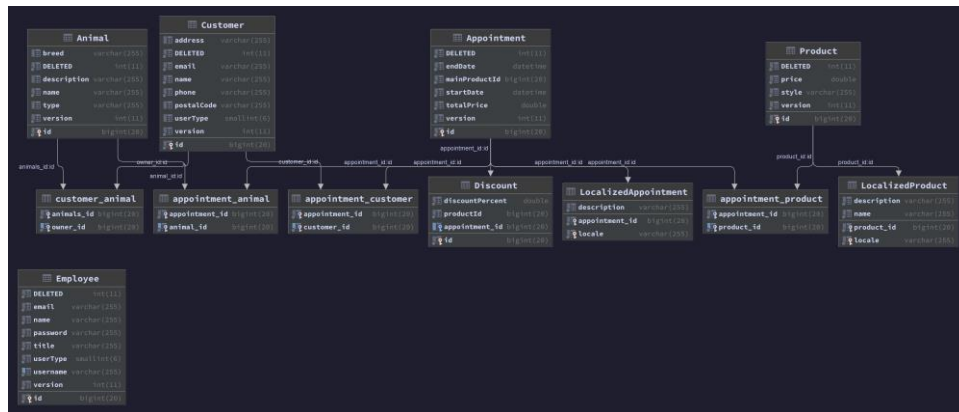
### 3 Käyttäjäroolit ja käyttötapaukset



Kuva 1. Käyttötapauskaavio

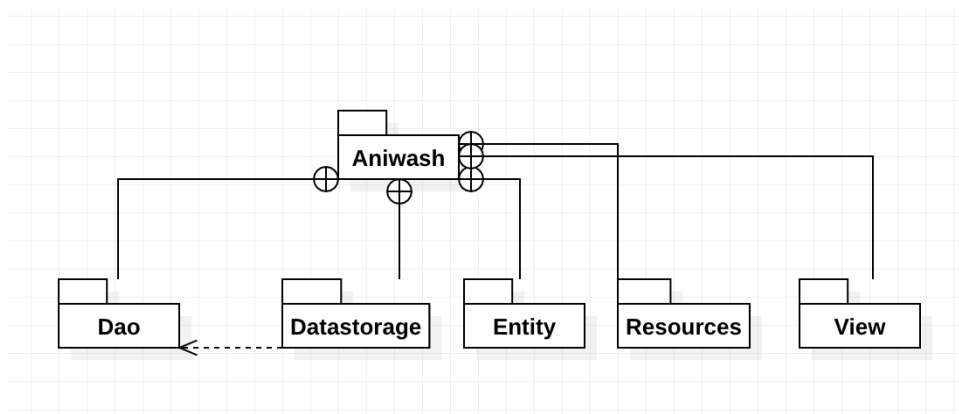
Käyttötapauskaavio kuvaa kolmea käyttäjää: Asiakasta, työntekijää ja työnantajaa. Kaaviossa sekä asiakas että työntekijä voivat tehdä varauksia, työntekijä voi tehdä ja muokata varauksia, sekä pystyy lisäämään asiakkaita ja lemmikkejä. Työnantaja sekä työntekijä näkevät varaukset. Työnantaja pystyy tekemään, muokkaamaan ja poistamaan tuotteita, joita työntekijät käyttävät varauksissa. Kuvan 1 kaavio oli ensimmäisiä prototyyppejä.





Kuva 3. Relaationtietokanta-mallinnus tietokannasta

## 5 Ohjelmiston rakenne

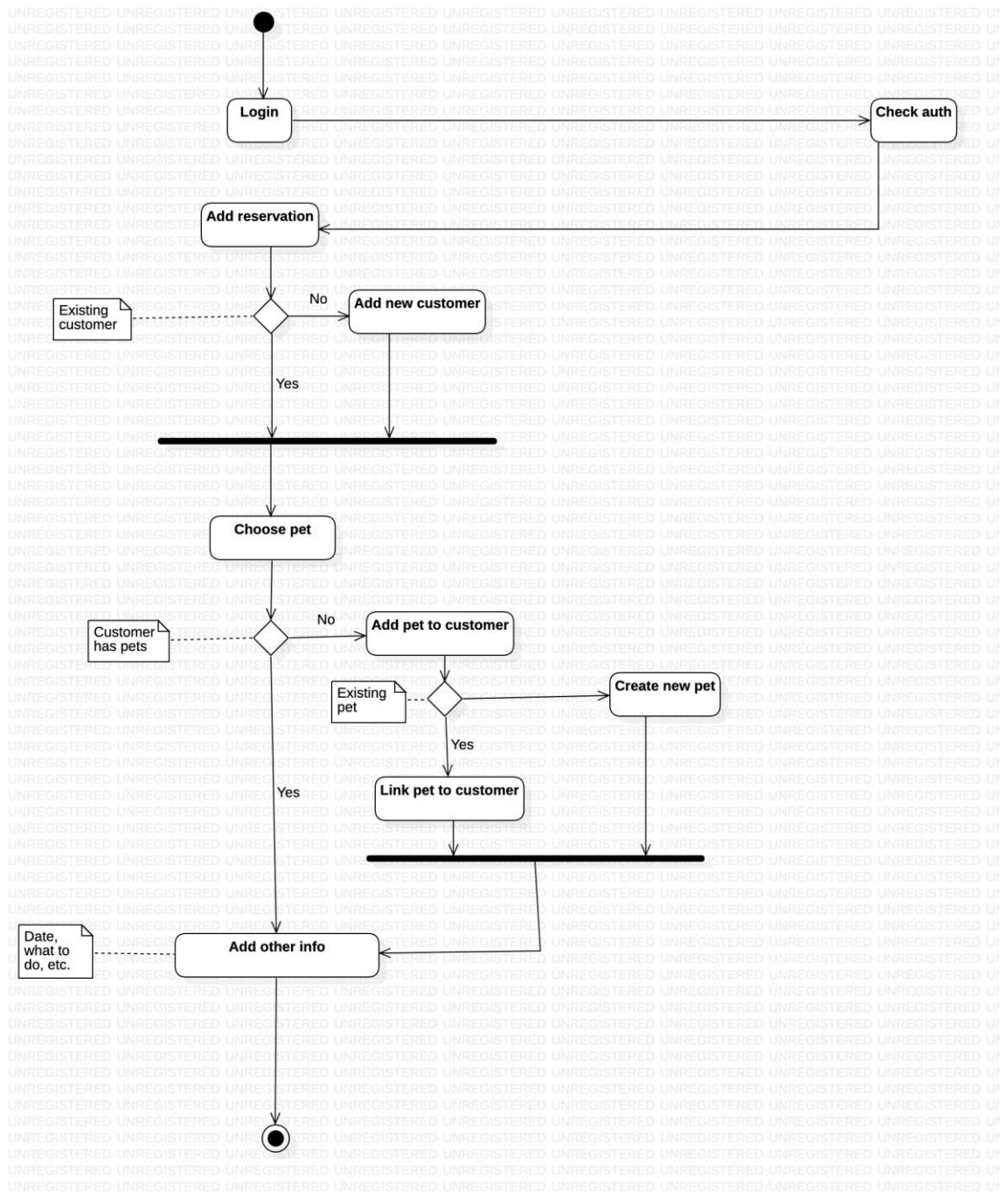


Kuva 4. Pakkauskaavio

Pakkauksista ”aniwash” sisältää MainApp:in josta ohjelman käynnistäminen tapahtuu, se tuo view-pakkauksesta eri näkymien kontrollereita. View-paketti hakee Resources-paketista apuvälineitä käyttäjänäkymään, sekä tuo tietokantaan tarvittavia käyttöliittymiä Dao-paketista ja luo järjestelmän entiteettejä niille suunnatuilla kontrolleri luokilla. Resources paketti sisältää myös lokalisaatiossa käytettävät properties tiedostot. Kuvassa 4 kuvattuna projektin pakkaus rakenne.



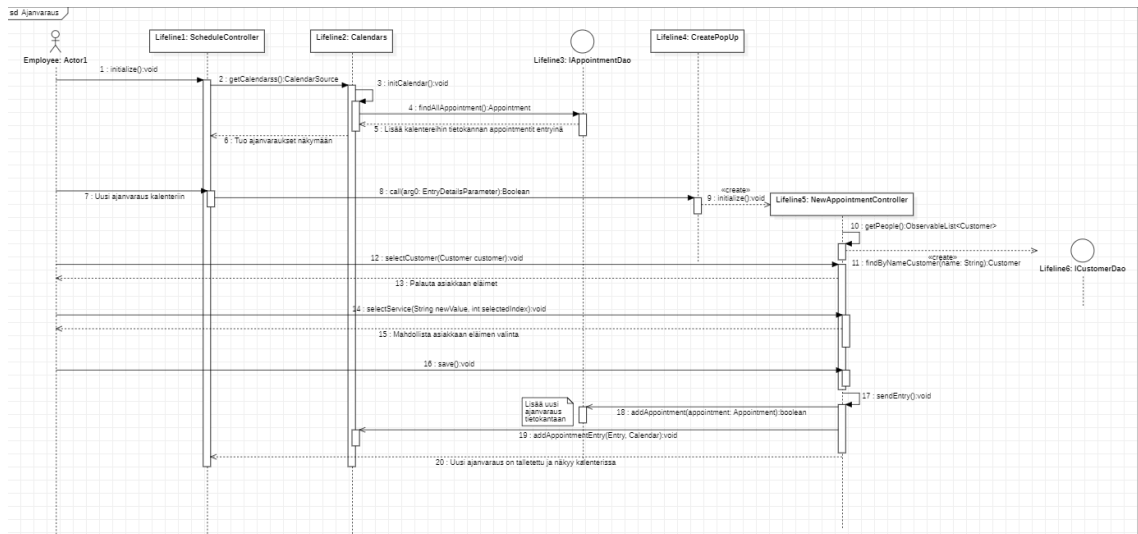
## 6 Ohjelmiston toiminta



Kuva 5. Aktiviteettikaavio

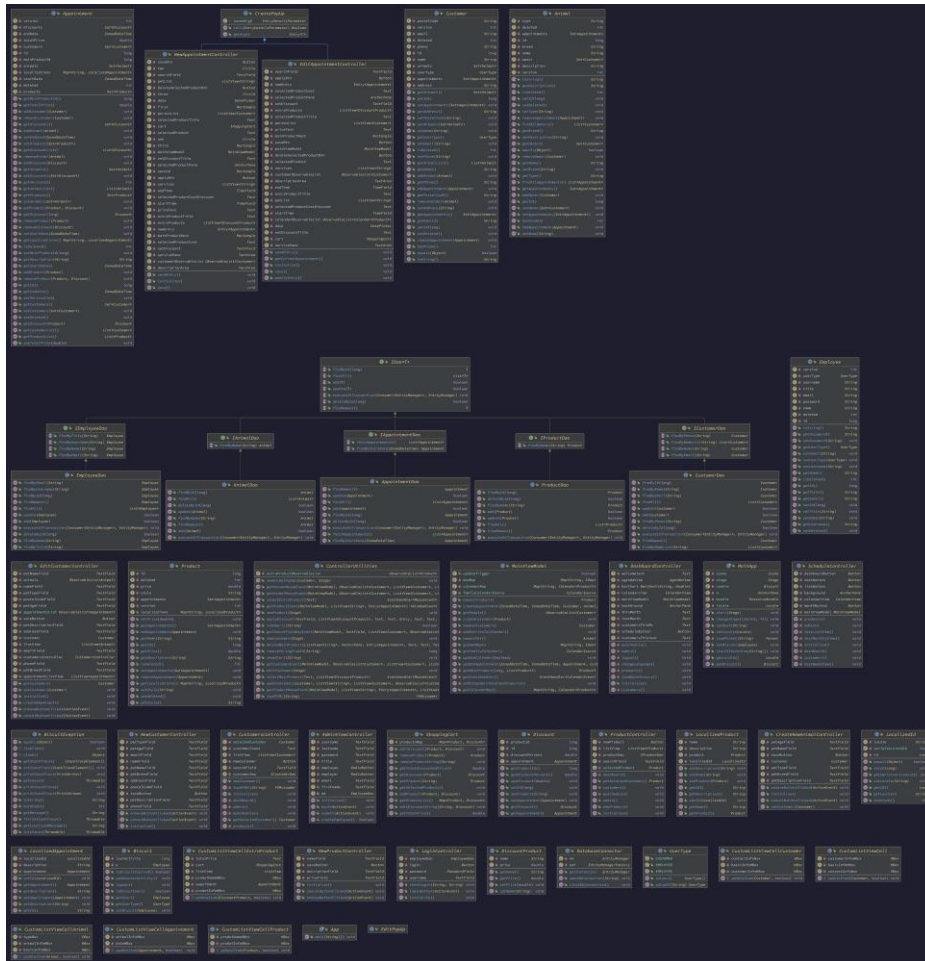
Kuvan 5 malli kuvaa työntekijää, joka kirjautuu sisään järjestelmään ja luo varauksen. Varausta luodessa työntekijän oikeus luoda varaus tarkistetaan.

Työntekijä valitsee asiakkaan tai luo uuden, valitsee lemmikin tai luo uuden asiakkaalle ja lisää muut varauksen tiedot (esim. Tuotteen ja asiakkaan toiveet).



Kuva 6. Sekvenssikaavio

Kuvan 6 sekvenssikaaviossa on johdettu (ei uuden) asiakkaan ajanvaraus. Ajanvarauksen tekijänä on työntekijä, joka menee ajanvaraus näkymään (ScheduleController). Täältä hän valitsee sopivan ajankohdan ja luo uuden ajanvarauksen järjestelmällä. Tallennettu ajanvaraus näkyy heti palatussa kalenteri näkymässä.



Kuva 7. Luokkakaavio

Kuvan 7. luokkakaaviossa MainViewModel pitää huolen yhteyksistä front- ja backendin välillä ja toteuttaa MVVM-arkkitehtuuria. Yleisesti ottaen tämä luokka tarjoaa toiminnallisuuden kalenterin hallintaan ja sen sisältämien tapahtumien, asiakkaiden, eläinten ja tuotteiden käsittelyyn. MainViewModelilla on pääsy kaikkiin IDao rajapinnan toteuttaviin DAO-luokkiin.

Ohjelmiston "view"-paketista löytyvät -Controller-päätteiset luokat vastaavat eri näkymien toiminnallisuuksien hallinnasta. Koska monet näkymät hyödyntävät samoja toimintoja, olemme luoneet yhteisen luokan nimeltä ControllerUtilities kontrollereiden toimintojen käsittelyyn.

Kirjautumisvälineinä käytettävät evästeet sijaitsevat Datastorage-paketissa, joka sisältää myös DatabaseConnector-luokan. Tämä luokka luo

EntityManagerFactory-olion DAO-luokkien käyttöön. Entities-paketin sisältä löytyy lisäksi Localization-alipaketti, josta löytyvät Product- ja Appointment-luokkien lokalisoinnit. Valittu lokalisointi tallennetaan tiedostoon luokan Language-Save avulla, jotta sovellus muistaa käyttäjän valinnan seuraavalla käynnistyksellä.

## **7 Kehitysprosessi ja kehitysvaiheen tekniikat**

Sovellusta kehitettiin alustavasti kahdessa osassa. UI toteutettiin ensin koodatulla datalla ja tietokanta erillisenä osiona testejä käyttämällä. Molempien osien saavutettua riittävän taso yhdistettiin osat toimivaksi tuotteeksi.

Koodaus on toteutettu vaihtelevasti Visual Studio Code:a tai IntelliJ IDEA käyttämällä koodaajan preferensseistä riippuen. Käyttöliittymä on suunniteltu, sekä toteutettu SceneBuilder:in avulla

UI on kehitetty JavaFX ohjelmistoalustaa hyödyntäen. Tietokanta on taas rakennettu Hibernate oliorelaatiokartoitustyökalu frameworkiä käyttäen. Kehitysprosessissa on hyödynnetty manuaalisia JUnit testejä, sekä automaattista Jenkins pipeline testausmenetelmää toimivuuden varmistamiseksi. Jenkins pipeline tuottaa ja esittää myös JaCoCo kirjaston kattavuus raportin testatun koodin osuudesta.

MariaDB:tä käyttävä tietokanta, sekä Jenkins pipeline pyörii Metropolian Ammattikorkeakoulun Linux pohjaisella palvelimella. Tietokantaan yhdistetään JDBC connectorin avulla, sekä yhteys vaatii yhteyden Metropolian verkkoon. Tämä voidaan toteuttaa VPN yhteyden avulla tai olemalla yhteydessä kampusen verkkoon.

Käytimme hyväksi kalenteri näkymässä valmista kirjastoa nimeltä CalendarFX, joka tarjosi valmiin komponentin kalenteri tapahtumien näyttämiseen. Riippuvuuksien hallintaan ja lataamiseen käytetään Maven työkalua.

Lokalisatio toteutettiin käyttäen Javan I10n ja Resource Bundle ominaisuuksia. Käyttöliittymä on käytettävissä täysin joko Englannin tai Ranskan kielellä. Käyttöliittymän kielen vaihtaminen pohjautuu Resource Bundlen Localen vaihtamiseen. Resource Bundle osaa valitun Localen avulla ottaa käyttöönsä käyttäjän valintaa vastaavan resource tiedoston, joka sisältää käyttöliittymän tarvitsemat String arvot. Kalenterissa käytettävä kirjasto CalendarFX osaa vaihtaa käytettävän kielen asetetun oletus Localen perusteella, joten tämä päivitetään myös käyttäjän vaihtaessa kieltä. Myös tietokanta on lokalisoitu tukemaan eri kielille määritettäviä tallennuksia, jos tiedot haluttaisiin tallentaa kieli kohtaisesti. Ohjelman tämänhetkisessä tilassa kaikki tiedot tosin tallennetaan kopiona molemmille kielille.

## 8 Ohjelmiston jatkokehitys

Tämän Java-ohjelman kehitystyön jatkokehityksessä on suunniteltu useita uusia ominaisuuksia, joiden avulla voidaan parantaa käyttäjäkokemusta ja ohjelman toiminnallisuutta. Suunnitelmissa on muun muassa push-notifikaatioiden käyttöönotto ajanvarauksista, työntekijöiden haun lisääminen, tilastotietojen parantaminen, kuten perutut ajanvaraukset ja työntekijöiden hoidetut ajanvaraukset.

Ohjelmiston jatkeeksi voisi myös kehittää verkkopohjaisen käyttöliittymän nykyisen kohdekäyttäjän asiakkaille, jonka kautta he voisivat tehdä omia ajanvarauksiaan, sekä tarkistaa ja päivittää omia tietojaan. Tästä oli varmasti hommaa omaksi pidemmäksi projektiksi asti.

Push-notifikaatioiden käyttöönotto mahdollistaa ilmoitusten lähettämisen käyttäjille tärkeistä tapahtumista, kuten ajanvarauksen vahvistamisesta tai perumisesta. Tämän ominaisuuden avulla voidaan parantaa asiakaskokemusta ja vähentää mahdollisia väärinkäsityksiä ajanvarauksiin liittyen.

Työntekijöiden haun lisääminen mahdollistaa työnantajalle helpon tavan etsiä ja löytää työntekijöitä. Hakutoimintojen avulla voidaan etsiä työntekijöitä nimen tai

tehtävän avulla. Tämä ominaisuus helpottaa työnantajaa tekemään mahdollisia muutoksia työntekijän tietoihin.

Tilastotietojen parantamisella, kuten perutut ajanvaraukset ja työntekijöiden hoidetut ajanvaraukset, voidaan tarjota työnantajille parempi käsitys siitä, kuinka hyvin ajanvaraukset sujuvat. Tämä ominaisuus voi auttaa työnantajaa näkemään kuinka hyvin työntekijöillä menee, ja etteivät työntekijät esimerkiksi väärinkäytä alennuksien antamisia.

Lisäksi on suositeltavaa harkita ohjelman käyttöliittymän parantamista ja responsiivisuuden lisäämistä. Tällä tavoin käyttäjät voivat käyttää ohjelmaa helpommin erikokoisilla laitteilla. Tämän ominaisuuden avulla voitaisiin tarjota parempi käyttökokemus ja lisätä ohjelman käytettävyyttä.

## **9 Yhteenveto**

Projektin skaalaan nähden ohjelmiston toteuttaminen eteni hyvin. Osa ominaisuuksista jouduttiin siirtämään tulevaan ja nykyisissä ominaisuuksissa on varmasti vielä viilattavaa, jotta virheiden mahdollisuus saadaan ajettua mahdollisimman alas.

Ajankäytöllisesti projekti pysyi tyydyttävästi hallussa. Käytimme enemmän tunteja kuin kursseihin olisi vaadittu, ja näin pääsimme henkilökohtaisesti asetettuihin tavoitteisiin, jotka ylittivät kurssien vaatimukset. Uusien asioiden opetteleminen vie aina aikansa ja laajemman projektin toteutus ei koskaan suju ilman aikaa vieviä yllätyksiä.

Ohjelmiston jatkokehitysideoiksi on suunniteltu verkkopohjaista käyttöliittymää jonka avulla asiakas voisi selata omia tietojaan ja mahdollisesti jopa tehdä varauksia. Toiseksi kehitysideaksi on ehdotettu tekstiviesti ja sähköposti muistutusten lähettämistä asiakkaalle, kun tämän ajanvaraus lähestyy.