

Jeremias Korhonen, Ida Kilpeläinen, Mikko Puustinen, Toni Naumanen

Remind Me

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikka

Välikirje

13.03.2019

Sisällys

1	Tavoite	1
1.1	Ominaisuudet	1
2	Ohjelmointi	2
2.1	Näkymät	2
2.2	Hibernate	3
2.3	Testaus	3
2.3.1	JUnit	3
2.3.2	Jenkins	3
2.4	Maven	4
3	Kehitettävää	4

1 Tavoite

Tavoitteena on rakentaa hyvinvointi- ja terveyssovellus muistisairaille asiakkaille.

Asiakas voi sovelluksesta nähdä tulevat ajanvaraukset, päivän sään ja suosituksen vaatuksesta, viestikentän, mihin työntekijä (lääkäri, hoitaja tai asiakaspalvelija) voi lähettää asiakkaalle viestejä. Asiakas voi myös kirjoittaa ylös veriarvonsa. Työntekijä voi nähdä asiakkaan veriarvot ja kertoa hänelle etänä ohjeita ja toimenpiteitä viestikenttään.

Sovelluksen ideana on vähentää tarpeettomia lääkäri-/hoitajakäyntejä, arvioida paremmin avun tarvetta ja etänä antaa asiakkaalle suoraan ohjeita. Lyhyesti sanottuna sovellus helpottaa asiakkaan ja omaisten elämää.

1.1 Ominaisuudet

Asiakas kirjautuessaan sisään näkee tervehdysruudun, päivän sään ja suositellun vaatetuksen sekä viestikentän, mikä näyttää mahdolliset ilmoitukset ja viestit henkilökunnalta. Asiakas voi myös käydä katsomassa kalenteria, joka näyttää ajanvaraukset ja muistutukset ja ilmoittaa veriarvokentässä mittaamansa veriarvot. Veriarvot siirtyvät automaattisesti henkilökunnan nähtäväksi.

Työntekijä voi lähettää asiakkaille etänä yksityisiä viestejä viestikenttään, lisätä aikoja kalenteriin ja katsoa sekä muuttaa terveystietoja. Ylläpitäjä voi lisätä henkilöitä tietokantaan ja muuttaa jo olemassa olevien henkilöiden tietoja tai käyttöoikeuksia.

2 Ohjelmointi

Sovellus on toteutettu Java-kielellä ja SQL-tietokannalla. Tietokanta sisältää jokaisen asiakkaan terveystiedot, reseptit, varaukset ja veriarvot. Henkilökunnalla on erillinen taulu, jossa määritellään jokaisen käyttöoikeudet. Salasanoja pidetään toistaiseksi tietokannassa.

2.1 Näkymät

Sovelluksessa on kolme näkymää: asiakas-, henkilökunta- ja ylläpitäjänäkymä. Näkymät vaihtuvat riippuen henkilön käyttöoikeuksista. Näkymien ulkoasu on rakennettu käyttäen hyödyllistä SceneBuilder-ulkoasutyökalua, mikä tuottaa *.fxml* tiedostot sovellusluokkien käytettäväksi (kts. Alla oleva koodi).

```
AnchorPane rootLayout = FXMLLoader.Load(getClass().getResource("/Scene.fxml"));
Scene scene = new Scene(rootLayout);
```

Näkymät ladataan FXMLLoader-kirjastolla. Näkymien vaihto tapahtuu sceneContent-metodilla (kts. Alla oleva koodi).

```
public void sceneContent(String fxml, MouseEvent event, String title) throws IOException {
    Pane p = FXMLLoader.load(getClass().getResource(fxml));
    Scene scene = new Scene(p);
    Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
    stage.setScene(scene);
    stage.setTitle(title);
    stage.show();
}
```

2.2 Hibernate

Hibernate-työkalua on käytetty tietokantayhteyden saamiseksi ja tietokannan tietojen hakemiseen/muuttamiseen.

```
public boolean createHenkilökunta(Henkilökunta henkilö) {
    Session istunto = istuntotehdas.openSession();
    Transaction transaktio = null;
    boolean onnistui = false;
    try {
        transaktio = istunto.beginTransaction();
        istunto.saveOrUpdate(henkilö);
        transaktio.commit();
        onnistui = true;
    } catch (Exception e) {
        if (transaktio != null) {
            transaktio.rollback();
        }
    } finally {
        istunto.close();
    }
    return onnistui;
}
```

Yllä oleva koodi vie henkilötiedot sisältävän henkilön tietokantaan.

2.3 Testaus

Testaukseen käytetään JUnit5-kirjastoa ja JavaFX-testaukseen testFX-kirjastoa

2.3.1 JUnit

JUnit-testejä on tällä hetkellä projektissa yli 50 muun muassa gettereille, settereille ja konstruktoreille.

2.3.2 Jenkins

Jenkins suorittaa kaikki tehdyt JUnit-testit joka versiohallinnan push-toiminnon yhteydessä ja tekee niistä yhteenvedon omalle palvelimelleen.

2.4 Maven

Käännös- ja koontityökalu, jolla voidaan hoitaa paremmin riippuvuuksia projektien välillä ja kolmannen osapuolen kirjastoihin. Pom.xml -tiedostoon sai tarvittavat kirjasto/lisäosariippuvuudet, mitä ollaan hyödynnetty sovelluskehityksessä. Tällä saadaan projekti loppujen lopuksi paketoitua levityskuntoon.

3 Kehitettävää

Ensi jaksossa on tehtävänä vielä henkilökunnalle näkymä ja sille metodit esimerkiksi aikojen lisäys kalenteriin ja viestin lähettäminen asiakkaalle. Asiakasnäkömään tehdään kalenteri ja sinne mahdolliset muistutukset sekä lääkäri-/hoitaja-ajat.

Koska tämä sovellus on tehty pääasiassa muistisairaille henkilöille, on pyrittävä muistuttamaan asiakkaita käyttämään sovellusta. Tarvitaan jokin muistutus tai ilmoitus. Tällä hetkellä sovellus toimii ainoastaan verkkoyhteydellä. Mobiiliversiokin on mietinnän alla, sillä realistisesti ajateltuna vanhemmat muistisairaat eivät juuri käytä tietokonetta, puhumattakaan sovelluksen käynnistämistä sieltä.

Turvallisuus on tärkeä, varsinkin sovellukselle, missä säilytetään henkilötietoja ja terveystietoja. Ensi jaksossa pyritään vaikeuttamaan ulkopuolisten mahdollisuutta päästä tietoihin käsiksi ja salasanan kryptaamiseen.