Lääkelaskenta- ja seurantasovellus - Väliraportti

# Kuvaus

Lääkelaskenta- ja seurantasovellus on tarkoitettu hoitotyön ja lääketieteen ammattilaisille. Sen tarkoituksena on yksinkertaistaa potilaskohtaisten lääkemääräysten tekemistä, lääkeannostusten arvioimista ja mahdollisten haitta- ja yhteisvaikutusten havaitsemista.

Sovellus yhdistää lääketietokannan, sairaustietokannan ja potilastietokannan yhden sovelluksen alle. Tämä mahdollistaa mm. vääristä annostuksista, potilaan allergioista ja haitallisista yhteisvaikutuksista varoittamisen lääkemääräysten tekemisen ja muuttamisen yhteydessä.

Sovellus myös mahdollistaa potilaan lääkityksen vaikutusten seurannan. Sovellus voi potilaan oireita ja potilaalle määrättyjen lääkkeiden tunnettujen haittavaikutusten perusteella löytää oireita aiheuttavan lääkkeen.

Sovelluksen ei ole tarkoitus korvata olemassa olevia potilastietojärjestelmiä, vaan toimia potilastietojärjestelmän rinnalla erillisenä lääkemääräystyökaluna.

# Toiminnallisuudet

Ylläpitäjä voi ensimmäisen käyttöönoton yhteydessä määrittää käytettävän potilastietokannan. Näin sovellusta voi käyttää eri hoitolaitokset.

Järjestelmää käytetään aina käyttöliittymän kautta ja vaatii aina kirjautumisen sisään, jottei väärinkäyttöä tapahtuisi.

Sovelluksessa ylläpitäjä voi hallinnoida ja lisätä muita käyttäjiä, sekä lukita käyttäjiltä pääsyn järjestelmään tarpeen vaatiessa.

Käyttöoikeudet määrittävät mitä järjestelmässä pystyy muokkaamaan ja näkemään. Lääkärin käyttöoikeuksilla voi nähdä järjestelmässä olevat potilaat, heidän diagnoosinsa ja lääkityksensä. Lääkärit voivat järjestelmästä tehdä uuden lääkemääräyksen, muokata jo tekemäänsä lääkemääräystä, tai katsella omia määräyksiään.

Hoitajilla on katseluoikeudet, mikä tarkoittaa, että he voivat katsella katsella potilaiden tietoja, diagnooseja ja reseptejä, sekä lääkkeiden tietoja.

Jokainen käyttäjä voi lähettää tai vastaanottaa viestejä toisilta käyttäjiltä, mikä mahdollistaa hoitajien ja lääkäreiden sulavan kommunikoimisen.

# 

# 

# Arkkitehtuuri

Sovellus noudattaa MVC-arkkitehtuuria. Sovelluksen näkymä on toteutettu siten, että uusien scenejen lataamisen sijaan ladataan uusia elementtejä sovelluksen “root”-elementtiin, jotta vältytään näkymän uudelleenlatauksilta. Kontrollerikerroksessa on pyritty huomioimaan se, että sovellus ei joudu lataamaan tietoja kahteen kertaan ja että kontrollerien ei tarvitse tuntea toisiaan. Model-kerroksessa on myös pyritty huomioimaan eri tietojen yhdisteleminen ja päällekäisten tietojen välttäminen.

## Model

Sovelluksen “Model” -osat ovat suunniteltu siten, että se mahdollistaa tarvittavien tietojen yhdistelemisen eri tietokannoista.

Kaikki asiakkaan resursseihin liittyvä tieto kootaan ClientResources-luokassa.

Ohjelman sen hetkisestä käyttäjästä ja kirjautumistiedoista pidetään kirjaa AppUser-luokassa.

PrescriptionMaker-luokka vastaa lääkemääräyksen annostuksen tarkistamisesta sekä potilaan allergioiden ja lääkeaineen allergeenien vertailusta.

DrugResources-luokan ainoana tehtävänä tällä hetkellä on hakea lääketietoja ja välittää ne kontrollerille.

## View

Sovelluksen tämänhetkinen “View” -osa on toteutettu “single-page application” -tyyliin. Näkymän eri osia ladataan .fxml-tiedostoista käyttäjän toimien ja käyttöoikeuksien perusteella.

MainGUI-luokka toteuttaa MainGUI\_IF-rajapintaa ja se sisältää perusnäkymän listojen luontimetodit ja event handlereiden luonnin perustoiminnoille. Main.fxml-tiedostoa käytetään näkymän “root”-elementtinä ja se ladataan ApplicationMain-luokassa.

LoginGUI -luokka lataa kirjautumisikkunan Login.fxml tiedostosta.

SideBarGUI ja SideBarListViewGUI -luokat sisältävät lähinnä tarvittavan logiikan sivuvalikon dynaamiseen luontiin. Sivuvalikon pohjana toimii SideBar.fxml tiedoston JavaFX-määrittely ja sen sisään ladataan TitledPanen sisään upotettuja listanäkymiä SideBarListView.fxml tiedostosta käyttäjän käyttöoikeuksien perusteella. SideBarListViewGUI on toteuttaa geneeristä SideBarListViewGUI\_IF-rajapintaa, joten sen avulla voidaan ladata eri tyyppisiä oliolistoja.

MessageFormGUI-luokka toteuttaa MessageFormGUI\_IF-rajapintaa ja sen tehtävänä on ladata viestin luontinäkymä ja kerätä käyttäjän lomakkeelle syöttämät tiedot.

PrescriptionFormGUI toteuttaa PrescriptionFormGUI\_IF-rajapintaa ja sen tehtävänä on ladata lääkemääräyksen luonti- tai päivityslomake ja kerätä käyttäjän lomakkeelle syöttämät tiedot.

## Kontrolleri

MainGUI:n ja ClientResources:n välillä tietoja välittää Controller\_IF-rajapintaa toteuttava Controller-luokka.

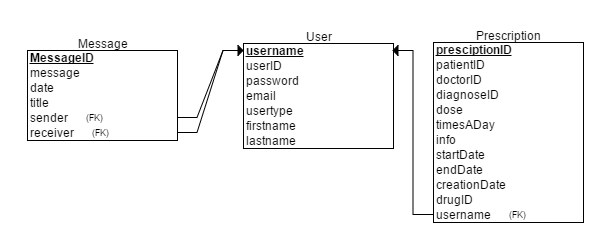
PrescriptionFormGUI ja PrescriptionMaker yhdistyy PrescriptionMakerController\_IF-rajapintaa toteuttavan PrescriptionMakerController-luokan avulla.

## Ulkoiset riippuvuudet

Sovelluksessa tietokantarakenne on toteutettu siten, että vain käyttäjät, viestit ja lääkemääräykset talletetaan sovelluksen omaan tietokantaan. Kaikki muu sovelluksen käyttämä tieto on “Read-only”. Tiedon turhan duplikoinnin välttämiseksi ja sovelluksen antaman tiedon ajantasaisuuden takaamiseksi sovellus on suunniteltu siten, että potilastiedot haetaan asiakaskonsernin tietokannasta, lääketiedot haetaan ulkopuolisesta lääketietokannasta ja tiedot sairauksista haetaan ulkopuolisesta sairaustietokannasta.

## Tietokanta

Käytämme sovelluksessa MySQL tietokantaa johon tallennetaan käyttäjät, reseptit ja viestit. Ohessa sovelluksen tietokannan relaatiomalli



**Lääkelasku- ja seurantasovellus - kohdealueen sanasto**

**Diagnoosi** tarkoittaa lääkemääräyksen perusteena käytettävää lääkärin tekemää potilaan sairauden toteamista ja sen syiden määrittelyä.

**Haittavaikutus** tarkoittaa mitä tahansa lääkityksestä aiheutuvaa potilaan terveyttä tai hyvinvointia alentavaa vaikutusta.

**Hoitaja** on henkilö, jolla on valtuus tarkastella kaikkien potilaiden tietoja ja tehdä hakuja lääketietokannasta, mutta ei valtuuksia lisätä tai muuttaa lääkemääräyksiä.

**Ilmoitus** on huomautus käyttäjän tilillä, jos järjestelmässä tapahtuu jotain mihin on asetettu ilmoituksen lähetys ominaisuus.

**Järjestelmä** on ohjelma, jossa kaikki toiminnot tapahtuvat.

**Kirjaus** on tapahtuma, jonka lääkäri tai hoitaja voi tehdä potilaan tietoihin.

**Käyttöoikeus** on oikeus, jonka ylläpitäjä voi antaa käyttäjälle. Käyttöoikeuksia voi lisätä tai poistaa.

**Käyttäjä** kuvaa kaikkia henkilöitä, joilla on käyttöoikeudet järjestelmään.

**Lääkemääräys** on määräys, jonka lääkäri voi tehdä potilaalle. Lääkemääräys kirjataan potilaan tietoihin.

**Lääketietokanta** tarkoittaa järjestelmän ulkopuolista tietokantaa, josta sovellus hakee lääkkeiden tiedot.

**Lääkäri** on henkilö, jolla on valtuudet tarkastella kaikkien potilaiden tietoja, tehdä hakuja lääketietokannasta ja lisätä sekä muuttaa lääkemääräyksiä.

**Potilas** on henkilö, jonka yleiset tiedot löytyvät potilastietokannasta ja jonka tietoja lääkärit ja hoitajat analysoivat ja kirjaavat.

**Potilastietokanta** tarkoittaa sitä järjestelmän ulkopuolista tietokantaa, josta sovellus hakee potilaiden tiedot.

**Tuotteen omistaja** on henkilö/yritys, joka omistaa tehtävän tuotteen.

**Yhteensopiva** tarkoittaa sitä, että kahden lääkkeen välillä ei ole etukäteen tiedossa olevia haitallisia yhteisvaikutuksia.

**Ylläpitäjä** on henkilö, joka vastaa sovelluksen käyttöönotosta ja/tai sen käyttäjien hallinnasta.

## 