

Ohjelmointi 3: Harjoitustyö

Tämä asiakirja on loppudokumentaatio työryhmän AJ&Puranen harjoitustyölle kurssilla COMP.CS.140. Harjoitustyönä toteutettiin ohjelma, jolla voidaan tarkastella Tampereen yliopiston tutkinto-ohjelmien rakenteita. Harjoitustyössä tutustuimme Sisun API:in, JavaFX:ään, ohjelman rakenteen suunnitteluun, Gitin käyttöön ryhmätyöskentelyssä, sekä tehtävien jakamiseen ja ohjelman toteuttamiseen yhdessä.

Ohjelman toiminta ja käyttöohjeet

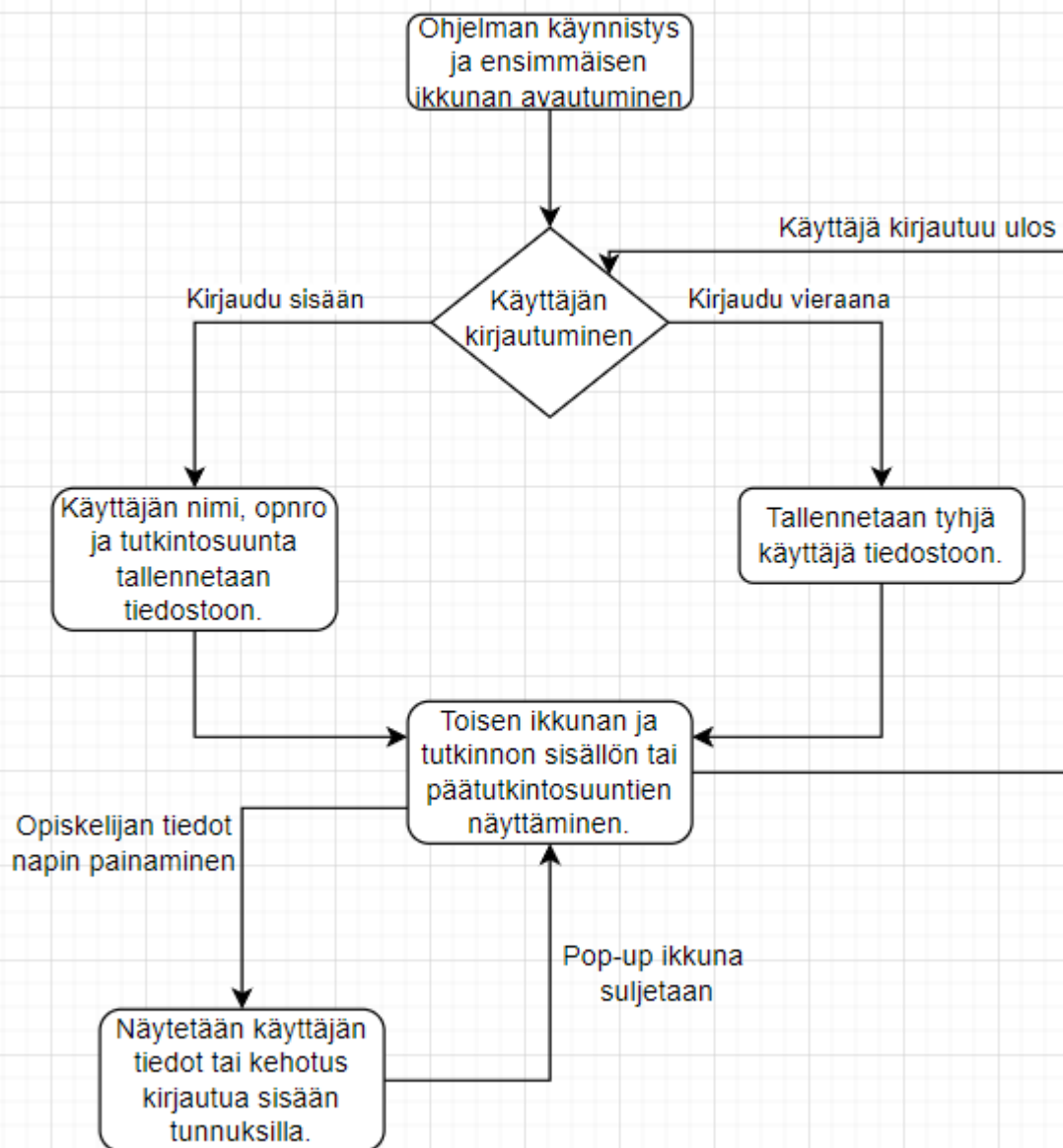
Kun ohjelma käynnistetään, aukeaa pääikkuna, johon käyttäjällä on mahdollisuus syöttää nimensä, opiskelijanumeronsa sekä valita tutkinnon pääaine. Käyttäjän tiedot luetaan kentistä ja tallennetaan kirjautumisen yhteydessä studentrecord.json tiedostoon hyödyntäen luokkaa user, kun käyttäjä painaa nappia *kirjaudu sisään*.

Kirjaudu sisään -napin painaminen edellyttää, että käyttäjä on syöttänyt kenttiin tietonsa, sekä valinnut päätutkinnon. Jos näin ei ole, ohjelma näyttää käyttäjälle puutteelliseen kenttään liittyvän virheviestin ja pyytää täydentämään puuttuvan kentän järjestyksessä nimi, opiskelijanumero tai tutkintosuunta. Halutessaan käyttäjä voi kirjautua myös anonyymisti painamalla nappia *kirjaudu vieraana*, jolloin tietoja ei syötetä ja tiedostoon tallennetaan tyhjä käyttäjä. Jos kyseistä tiedostoa ei ohjelman hakemistosta löydy, tulostetaan konsoliin viesti *Error: No previous student data, creating new student record* ja kyseinen tiedosto luodaan. Tiedostoon tallennetut käyttäjät tunnistetaan opiskelijanumeron perusteella, jolloin opiskelijanumeroduplikaatteja ei sallita, mutta saman nimiset käyttäjät sallitaan. Saman opiskelijanumeron omaavalla käyttäjällä ei myöskään voi olla useampaa kuin yksi päätutkinto. Jos käyttäjä yrittää luoda opiskelijanumeroduplikaatin tai valita virheellisen tutkinnon, näyttää ohjelma hänelle virheviestin tähän liittyen ja estää kirjautumisen.

Käyttäjän kirjautuessa sisään, ohjelman ikkuna vaihtuu näkymään, jossa valitun tutkinnon rakenne näytetään. Käyttäjän kirjautuessa vieraana, näytetään vain mahdolliset päätutkintosuunnat. Uudessa ikkunassa on myös navigaatiopalkki, josta käyttäjä voi avata omat tietonsa, kirjautua ulos tai vaihtaa ohjelman kieltä. Käyttäjän painaessa opiskelijan tiedot nappia, avautuu uusi ikkuna, jossa näytetään tämänhetkisen käyttäjän tiedot. Käyttäjän ollessa kirjautuneena vieraana, opiskelijan tiedot -ikkunassa näytetään kehoitus kirjautua sisään tunnuksilla.

Ohjelmaan toteutettiin lisäominaisuutena mahdollisuus vaihtaa näkyvän ikkunan kieltä. Oletuskielenä on suomi ja se on jokaisen ikkunan primäärikieli. Käyttäjällä on mahdollisuus vaihtaa ohjelman ikkunan kieltä suomen ja englannin välillä painamalla ikkunassa näkyvää Finnish tai English nappia.

Ohjelman toimintaa on esitetty seuraavassa kuvassa vuokaavion avulla.



Luokkien vastuujako

Ohjelman toteutuksesta löytyy 5 luokkaa, joilla kaikilla on oma tarkoituksensa. Luokkien nimet ovat Javan periaatteiden mukaan samat kuin tiedostojen, joista ne löytyvät.

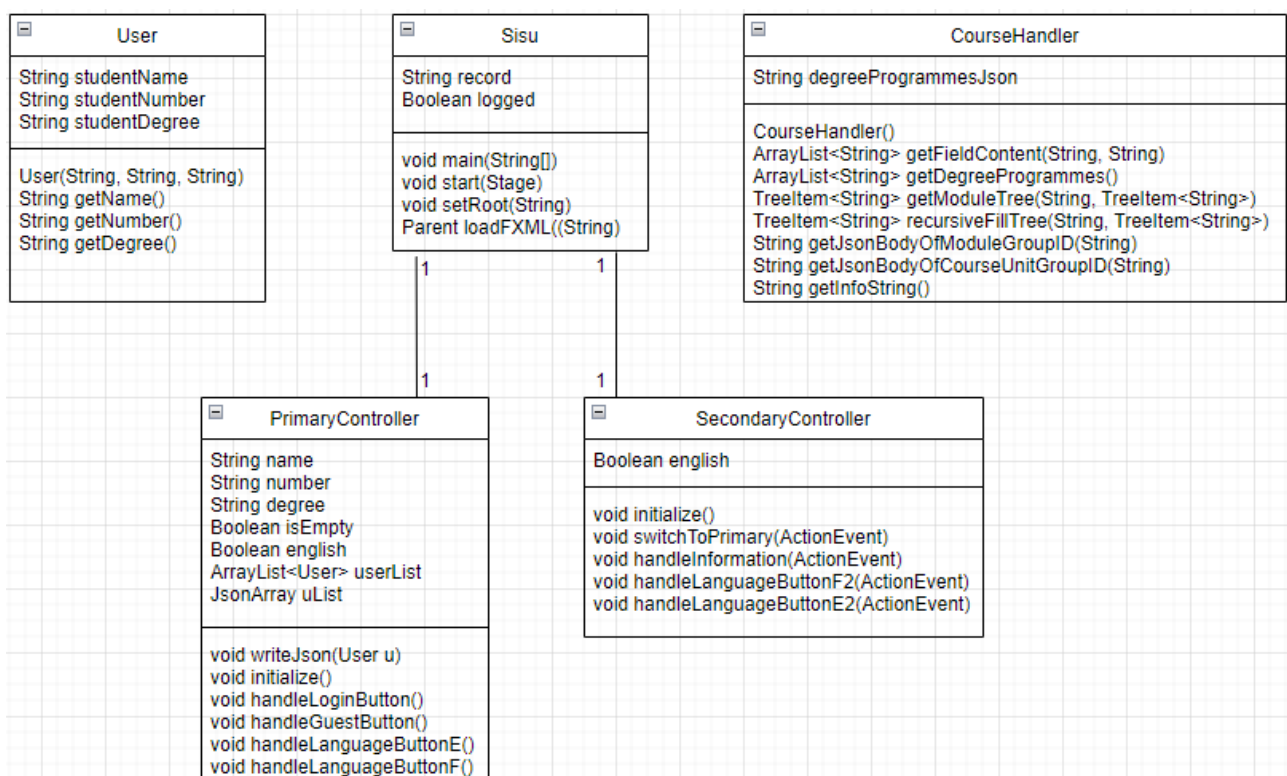
CourseHandler on apuluokka, jolla ohjelman API kutsut käsitellään. Luokka muovaa kutsusta saadut tiedot oikeaan formaattiin ohjelmaa varten. Apuluokkaa käytetään SecondaryControllerin yhteydessä.

PrimaryController on luokka, joka kontrolloi ohjelman ensimmäistä ikkunaa. Luokka sisältää nappien tapahtumankäsittelijät, sekä kirjoittaa dataa studentrecord.json tiedostoon käyttäjän kirjautuessa sisään. Perii luokan Sisu.

SecondaryController on luokka, joka kontrolloi ohjelman toista ikkunaa. Luokka sisältää ikkunan nappien tapahtumankäsittelijät. Luokka myös luo uuden ikkunan, missä opiskelijan tiedot näytetään, kun käyttäjä painaa nappia *Opiskelijan tiedot*. Perii luokan Sisu.

Sisu on ohjelman pääluokka, mikä käynnistää ohjelman ja hakee tyyliasetukset tiedostosta outlook.css.

User on ohjelman apuluokka, jota käytetään käyttäjän datan tallentamisessa oikeaan muotoon.



Työnjako projektin aikana

Projektin alkaessa, huomattiin projektin koostuvan pääpiirteittäin kahdesta osiosta: GUI ja API. Tämä pääpiirteinen jako toimi myös projektin työnjakona, eli Antti-Jussi Isoviita toteutti ohjelman GUI puolen ja Aleksi Puranen ohjelman API puolen. Työnjaon ollessa selkeä, kummatkin osapuolet pystyivät keskittymään omaan työskentelyynsä, mutta tarvittaessa myös auttamaan toista ratkaisemaan ongelmia.

Ongelmat ja puutteet

Projektin aikana ongelmia ja harmaita hiuksia tuottivat uudet aihealueet, joita kurssin aikana ei ollut käsitelty, esimerkiksi TestFX viitekehys ja API kutsujen käsittely. Tämän vuoksi projektista ei löydy kaikkia toivottuja ominaisuuksia. Puuttuvia ominaisuuksia ovat yksikkötestaus TestFX viitekehyksellä. Sisusta haettu JSON-data oli myös osittain haasteellista käsitellä haluttuun muotoon.

Alun perin ohjelmaa suunniteltaessa kirjautumisominaisuus toteutettiin toiveessa lisätä myös mahdollisuus seurata opiskelijan etenemistä valittujen kurssien muodossa, mutta aikarajoitusten vuoksi etenemisen seuranta jäi toteuttamatta. Kuitenkin kirjautumisominaisuus jäi ohjelmaan.

Ohjelmaan on toteutettu perustoteutuksesta:

- Graafinen käyttöliittymä kokonaan itse
- Ohjelma hakee tutkintojen rakennetiedot Sisun API:sta
- Ohjelma toimii, näytettävää tutkintoa voi vaihtaa

Toteutettuja lisäominaisuuksia:

- Opiskelija-asetukset (kirjautuminen sisään), kuitenkin suorituksen tilanne ei ole implementoitu
- Oma lisäominaisuus: kirjautuminen vieraana, jolloin voi tarkastella valittavia tutkintoja
- Oma lisäominaisuus: kielen vaihtaminen