

Projekti: Sisusta selvää

Ohjelma toteuttaa työkalun, jolla voidaan tarkastella Tampereen yliopiston kursseja Sisusta, sekä seurata opiskelijoiden edistymistä opinnoissaan.

Luokkien vastuujako

DegreeProgramme

- Luo tutkinto-ohjelman Sisu API:n JSON datasta

StudyModule

- Luo opintomodulin Sisu API:n JSON datasta

CourseUnit

- Luo kurssin Sisu API:n JSON datasta

Module

- Abstrakti yläluokka CourseUnitille, DegreeProgrammelle ja StudyModulelle
- Mahdollistaa rekursion API:ssa

JSONLogic

- Toteuttaa konkreettisen rajapinnan Sisun API:n kanssa tietojen saamiseksi
- Yksinkertaistaa tiedonhakua ja sen tallentamista opiskelijoille student.json-tiedoston kanssa

Sisu

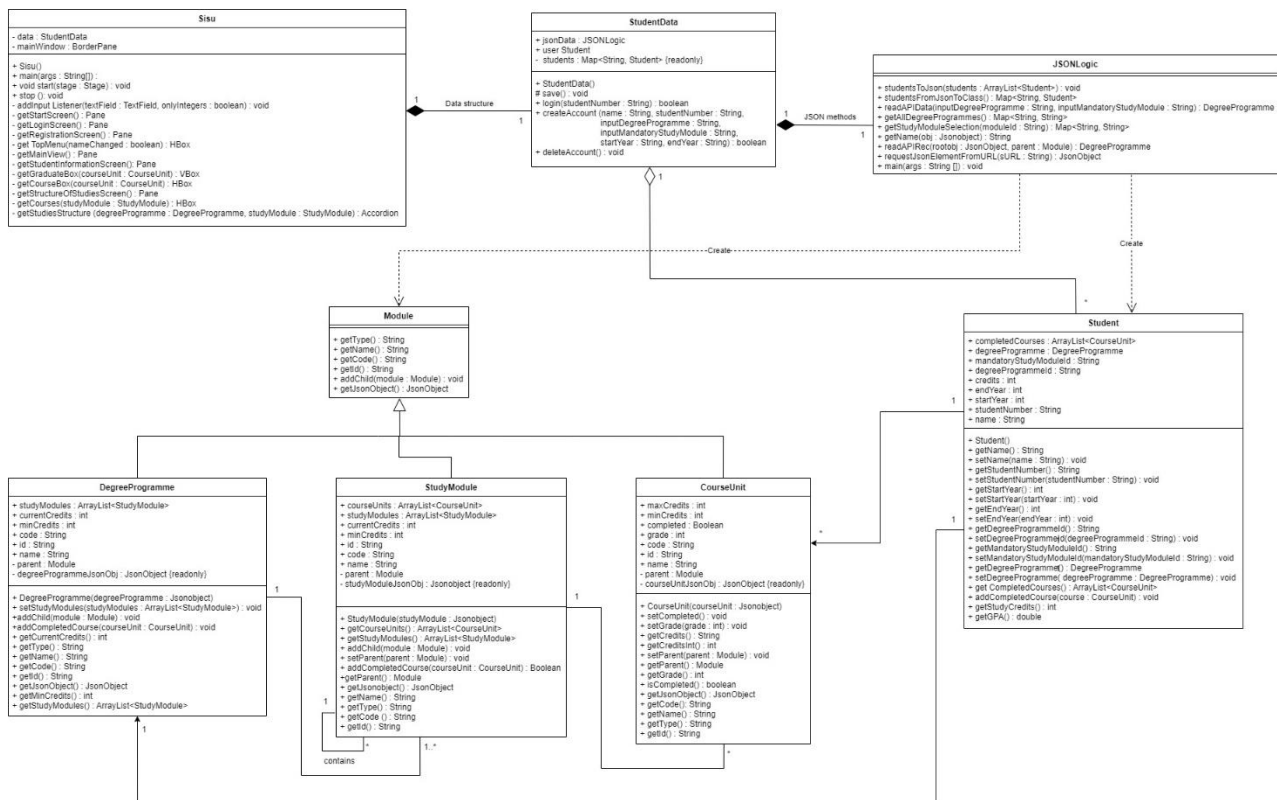
- Toteuttaa käyttöliittymän kokonaisuudessaan.
- Rajapinta luokan StudentData kanssa tietojen käsittelemiseksi

Student

- Toteuttaa opiskelijan Tampereen yliopistossa
- Tallettaa tiedot opiskelijasta
- Rajapinta DegreeProgrammen ja CourseUnitin kanssa opintotietoja varten

StudentData

- Toteuttaa tietorakenteen ohjelman käyttöön
- Tallettaa ohjelman ajon ajaksi tiedot opiskelijasta ja kursseista
- Rajapinta luokan JSONLogic kanssa tietojen saamiseksi Sisun apista ja käyttäjätietojen tallettamiseksi ohjelmaa sulkiessa



Kuva 1: luokkakaavio

Ohjelman toiminta ja käyttöohje

Ohjelma alkaa aloitusnäkyimestä, josta voidaan valita, halutaanko kirjautua sisään vai luoda uusi käyttäjä.

(1)

Kirjautumisnäkyimässä syötetään opiskelijanumero. Aiemmin rekisteröidyllä opiskelijanumerolla pääsee sisään, muussa tapauksessa ohjelma ilmoittaa käyttäjälle virheellisestä syötteestä. (2)

Rekisteröintinäkyimässä käyttäjä syöttää nimensä, opiskelijanumeronsa ja valitsee valikosta tutkinnon. (3) Halutessaan hän voi valita aloitusvuoden ja lopetusvuoden, mutta vain kummatkin eikä vain toista. Ohjelma ilmoittaa virheellisistä syötteistä.

Mikäli kirjautuminen tai rekisteröityminen onnistuu, avautuu ohjelma päänäkyymään. Päänäkyimässä käyttäjä voi tarkastella keskiarvoaan, sekä suoritettuja opintopisteitä. Luvut päivittyvät opintojen edetessä.

(4)

Yläpalkista käyttäjä voi navigoida sivujen välillä tai kirjautua halutessaan ulos.

Opiskelijatietonäkyimestä käyttäjä voi tarkastella syöttämiään tietoja. Tiedoista pystyy muokkaamaan nimeä ja aloitus, sekä lopetusvuotta. Käyttäjän voi myös poistaa, jolloin se ei ole enää palautettavissa. (5)

Tutkintorakennenaäkyimässä käyttäjä voi tarkastella tutkintoonsa kuuluvia kursseja. Kursseja voidaan merkitä suoritetuksi merkitsemällä kurssilaatikon oikeassa laidassa olevaan tekstikenttään arvosana. Arvosanat ovat 0–5, nollan ollessa suoritusmerkintä ilman arvosanaa. (6)

Kirjautuessa ulos, palataan takaisin aloitusnäkyymään. Ohjelman sulkeutuessa tallennetaan kaikki opiskelijoiden olemassa olevat ja muutetut tiedot ensi kertaa varten. (7)

Mitä tapahtuu taustalla JSONLogic ja muut:

Yllä olevassa toiminnan selosteessa on numeroita (x), joita selitetään auki tässä luvussa.

(1) Ohjelman käynnistyessä kutsutaan students.json tiedostosta olemassa olevat opiskelijat studentsFromJsonToClass() funktiolla. Luetaan students.json tiedosto helposti Gsonin avulla. Funktio palauttaa mapin, jossa on avaimena opiskelijanumero ja arvona opiskelijan Student luokka ilman tutkinto-ohjelmaa. Tämän avulla voidaan kirjautua olemassa olevalle käyttäjälle.

(2) Kirjaututaan olemassa olevalle käyttäjälle ja pohjustetaan opiskelijan tutkinto-ohjelma readAPIData():n avulla. Student luokassa on tutkinto-ohjelman ja mahdollisen opintosuuntauksen id:t, joilla readAPIDataa kutsutaan. readAPIData lukee tutkinto-ohjelman Sisun APIsta ja muokkaa tietyn opintosuuntauksen, mikäli sellainen on. Tutkinto-ohjelma rakentuu DegreeProgramme olioksi, joka sisältää StudyModuleita, jotka sisältävät muita lisää StudyModuleita tai kursseja. Aikaisemmin suoritettut kurssit merkataan juuri rakennettuun tutkinto-ohjelmaan suoritetuiksi.

(3) Käyttäjä valitsee tutkinto-ohjelman Sisun APIsta haetuista tutkinto-ohjelmista getAllDegreeProgrammes(). Valitulla tutkinto-ohjelmalla voi olla jokin pakollinen opintosuuntausvalinta, kuten Tieto- tai Sähkötekniikka. Tällöin annetaan käyttäjälle lisävalikko vaihtoehtoista getStudyModuleSelection().

Rekisteröityessä luodaan uusi Student, johon annetut parametrit asetetaan. Uusi käyttäjä lisää StudentDatan students mappiin.

(4) Opiskelijan tiedot tulevat Student-oliosta

(5) Käyttäjän muokkaamat tiedot muokkaavat vain ajoaikaisia tietoja Student-olioista tai StudetDatasta.

(6) Kurssi merkitään suoritetuksi annetulla arvosanalla, jolloin kutsutaan Student-olion addCompletedCourse()-funktiota. Funktio merkitsee annetun kurssin suoritetuksi ”ylhäältä päin” DegreeProgramme -> StudyModule -> CourseUnit ja lisää kurssin completedCourses-listaan.

(7) Ohjelman sulkeutuessa tallennetaan ajo-aikana muokattu tieto students.json-tiedostoon. Käytetään Gsonin valmista ja helppoa kirjastoa, joka luo annetuista olioista (Student) json-muodon halutulla tavalla. Aikaisempi, vanhaa tietoa omaava students.json kirjoitetaan yli uudella ja päivittyneellä students.json-tiedostolla.

Toteutetut lisäominaisuudet:

- Opiskelija-asetukset. Käyttäjältä pyydetään rekisteröityessä tiedot. Tietoja voi muokata ja ne talletetaan JSON muodossa tiedostoon ohjelman sulkeuduttua. Ohjelman avautuessa uudestaan voidaan käyttäjäprofiili ladata uudelleen tiedostosta kirjautumisruudun kautta.
- Opiskelijan eteneminen peilataan tutkintorakenteeseen. Sisun API:sta saadaan tutkinnon vähimmäisopintopistemäärä, johon verrataan suoritettuja opintopisteitä. Näkymä esitetään aloitusruudulla.
- Kurssien keskiarvot. Kurssien keskiarvot lasketaan ja esitetään aloitusruudulla. Suoritusmerkinnällä läpäistyjä kursseja (arvosana = 0) ei lasketa mukaan keskiarvoon.
- Sisun API:sta haetaan kurssien opintopistemäärät, jotka esitetään tutkinnon rakennenäkyssä.'

Muu lisätyö:

- Yksikkötestauksia on toteutettu ympäri ohjelmaa.
- Grafiikan taso. Otimme tavoitteeksemme luoda mahdollisimman autenttisen Sisun. Ohjelman elementeille on luotu oma tyylitiedosto Sisu.css, josta itse määritellyt tyylit asetetaan käyttöliittymään. Ohjelmasta löytyy värien lisäksi vektorigrafiikkaa (painikkeiden nuolet) sekä kuvia (taustakuva, sisulogo, kurssin suorituslogo, ohjelman kuvake)
- Erittäin hyvä käyttöliittymä. Käyttöliittymästä on tehty selkeä ja intuitiivinen. Virheelliset syötteet on estetty.
- Laadukkaasti dokumentoidut rajapinnat ja luokat. Luokista on toteutettu Javadocs selkeät dokumentaatiot.

Työnjako

Sovittu

- Aapo tekee Sisun API:n ja Jsonien kanssa juttuja ja muokkaa datan käytettävään muotoon.
- Kasper tekee käyttöliittymää ja luo apuluokan ohjelman ajon ajaksi tietorakenteeksi

Toteutunut:

- Aapo loi CourseUnitin, DegreeProgrammen, JSONLogicin, Modulen ja StudyModulen
- Kasper teki Sisun resursseineen, Studentin ja StudentDatan.
- Molemmat testasivat ohjelmaa ahkerasti.
- Kasper kirjoitti dokumentaatioon pohjan, Aapo auttoi viimeistelyssä.
- Kasper teki luokkakaavion
- Aapo teki Javadocin
- Molemmat oppivat toistensa työstä

Tiedossa olevat ongelmat ja puutteet

- Tekstikenttien tarkastelut voisivat olla tarkemmat. Opiskelijanumerosta ei tarkasteta mitään, sillä emme olleet aivan varmoja Tampereen yliopiston opiskelijanumeropolitiikasta. Aloituss- ja lopetusvuosista tarkastetaan, että vuosi on 1-4 merkin kokonaisluku. Emme ottaneet kantaa ajan rakenteeseen, emmekä yliopiston opinto-oikeuksien pituuteen. Vuosia ei verrata keskenään ja siten aloitusvuosi voi olla myöhäisempi kuin lopetusvuosi ja tutkintoaika voi kestää vuosituhansia.
- Emme keksineet tapaa saada vapaavalintaisiin opintoihin kurssivaihtoehtoja, koska se olisi ollut turhan työlästä vapaavalintaisten opintojen ollessa mitä tahansa tutkintorakenteen ulkopuolisia opintoja.
- Testaus voisi olla parempaa. TestFX vei lukuisia turhia tunteja yrittäessä saada se toimimaan. Hienot TestFX-testit jäivät siis tekemättä, mutta yksinkertaiset JUnit5-testit ovat olemassa eri luokille.