

COMPS.CS.140 Dokumentaatio

Coding monkes

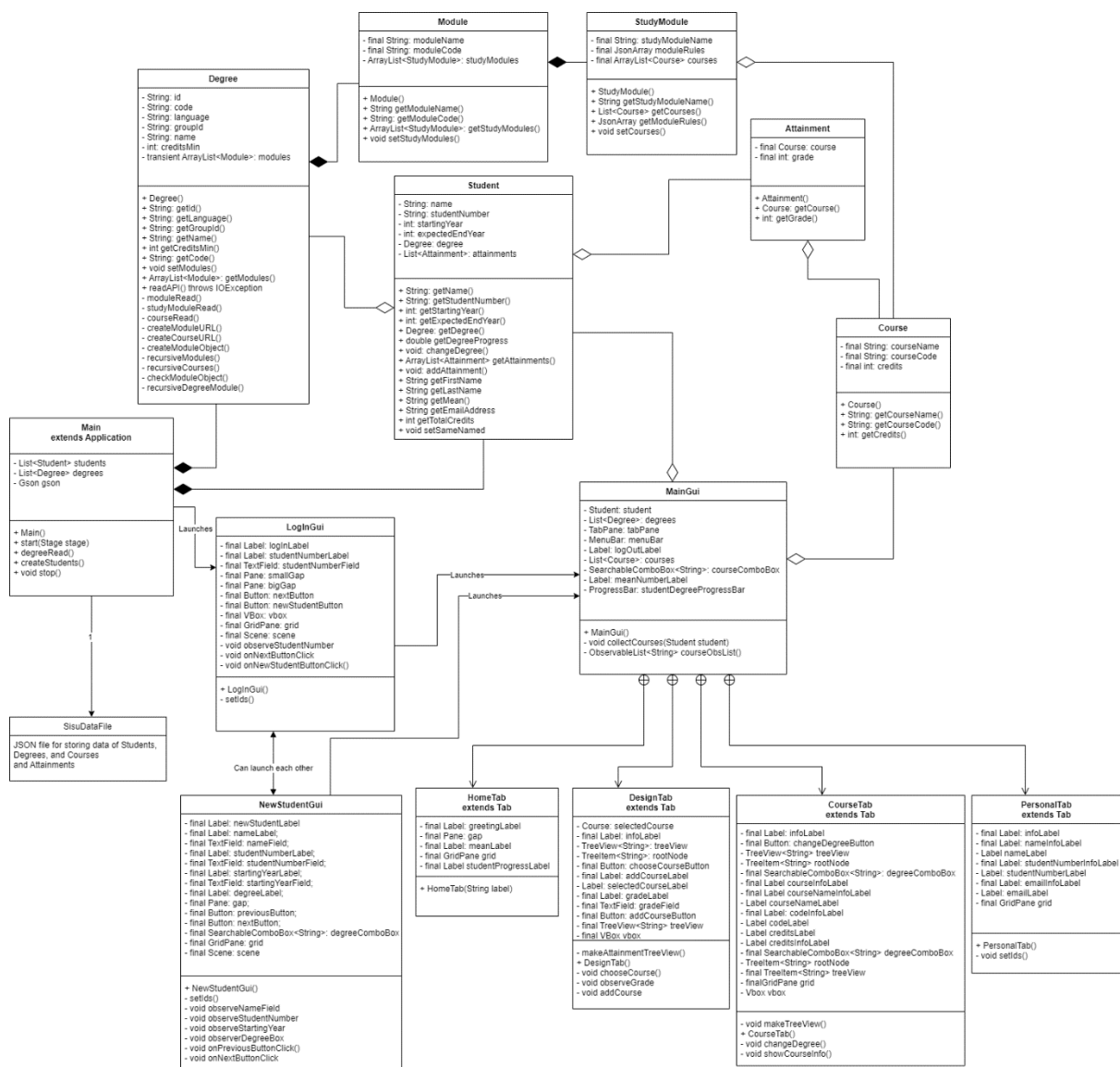
H299725 Onni Merilä

50121133 Ronja Lipsonen

50233597 Ville Niemi

Ohjelman rakenne

Ohjelma rakentuu pääohjelmasta, GUI-luokista sekä oppilaaseen ja tutkintoon liittyvistä luokista.

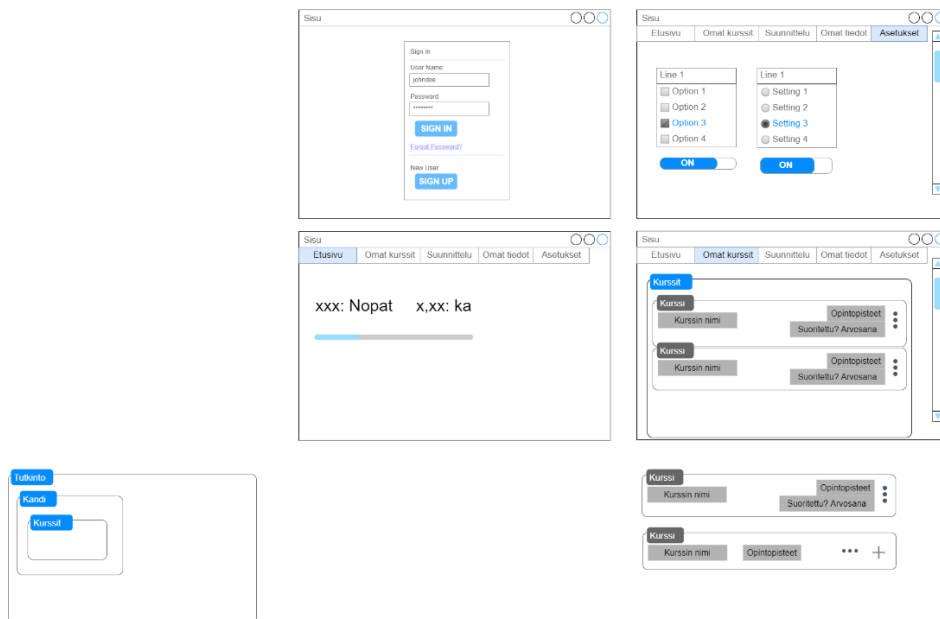


Kuva 1: Luokkakaavio

Luokkakaavio on laaja ja ei mahdu kokonaisuudessaan hyvin dokumentaatioon kuvana, joten alta löytyy linkki drawioon, jossa luokkakaavio on luotu.

Suunnittelu

Ohjelman suunnittelu tapahtui Telegram-ryhmässämme, Trellossa, Drawiossa sekä etäkokouksissa. Alla on esitetty kuvankaappaus GUI rautalankamalleista.



Kuva2: Rautalankamalli

Vastuujako

Ohjelmamme tärkeimmät luokat on esitetty alla. Muut luokat on jätetty listaamatta, sillä ne ovat joko niin yksinkertaisia tai niiden käyttö nivoutuu vahvasti alla listattuihin luokkiin.

Main, Degree – Ville

Main ja Degree ovat melko paljon yhteydessä toisiinsa, Main käynnistää ohjelman ja luo uudet oliot Degree-luokkaan. Degree-luokassa kaikki API:in liittyvät funktiot ja ominaisuudet esimerkiksi rekursiot ja tallentamiset tietorakenteisiin.

LogInGui, NewStudentGui ja MainGui – Ronja

Luokat on ilmoitettu samaan otsikkoon, sillä ne ovat kaikki yhteydessä toisiinsa ja sama toteuttaja on luonut ne kaikki samalla toimintaperiaatteella. Kaikki kyseiset luokat käyttävät samaa ikkunaa toteutuksessaan.

SisuTest - Onni

Ohjelman toiminta ja ominaisuudet

Toiminta

Main aloittaa ohjelman, joka käynnistää GUI:n ja alustaa ohjelman esimerkiksi lukemalla kaikki Degreeet ja luomalla students-listan JSON-tiedostosta "Sisudatafile.json". Tämän jälkeen ohjelma siirtyy sisäänkirjautumisikkunaan.

Ohjelma sisältää kolme eri ikkunanäkymää.

Sisäänkirjautumisikkunassa ohjelma lukee käyttäjän syöttämän opiskelijanumeron ja tarkistaa löytyykö se opiskelijatiedot sisältävästä JSON-tiedostosta. Tästä ikkunasta pääsee myös ikkunaan, jossa luodaan uusi opiskelija.

Uusi opiskelija – ikkunassa käyttäjä voi luoda kokonaan uuden opiskelijan, mikä ohjelman ajon aikana tallentuu listaan ja ajon jälkeen kyseinen lista tallentuu JSON-tiedostoon. Kyseisestä ikkunasta pääsee takaisin sisäänkirjautumisikkunaan tai pääikkunaan.

Käyttäjä valitsee opiskelijalle tutkinnon, jonka jälkeen kutsutaan readAPI() funktiota, joka löytyy Degree-luokasta. Tällöin ohjelma hakee tiedot juuri kyseiselle Degreeelle, jolloin ohjelman suorittaminen on melko nopeaa. Tiedot haetaan rekursiivisesti API:n syövereistä, josta löytyy monia eri tasoja moduulien ja kurssien myötä.

Ohjelman pääikkunassa käyttäjä voi tarkastella etenemistään ja keskiarvoaan. Käyttäjä voi lisätä kursseja arvosanan kera ja vaihtaa tutkintonsa. Vaihtamisen jälkeen ohjelma hakee API:sta tiedot uudesta tutkinnosta, joka voi joskus kestää muutaman sekunnin.

Käyttäjä voi myös tarkastella suorittamiaan kursseja, tutkintoonsa kuuluvia kursseja ja henkilötietojaan. Käyttäjä voi myös kirjautua ulos pääikkunasta takaisin sisäänkirjautumisikkunaan.

Toteutetut ominaisuudet

Ohjelmassa on toteutettu kaikki perus- ja täten myös minimiominaisuudet.

Toteutetut lisäominaisuudet

- Kurssin katselunäkymä. Klikatessa kurssia, kyseisestä kurssista näytetään lisätietoja näytöllä.
- Opiskelija-asetukset. Ohjelman käynnistyksen yhteydessä avataan asetussikkuna, josta voi asettaa ohjelmaa käyttävän opiskelijan tiedot. Suorituksen tilanne haetaan ja tallennetaan JSON-tiedostoon.
- Opiskelijan etenemisen aikataulun esittäminen tutkinnon rakenteeseen peilaten.

- Sisun API:sta löytyvän jonkin tietokokonaisuuden esim. arviointikohteiden esittäminen osana ohjelmaa (opintopisteet).
- Pääikkunasta voi kirjautua ulos. Ohjelma muistaa ajonkin aikana luodut oppilaat.
- Ohjelmassa voi olla usea käyttäjä. Käyttäjät voivat kirjautua sisään opiskelijanumeron perusteella.
- Ohjelma laskee opiskelijan keskiarvon ja päivittää sitä sen mukaan, kun kursseja lisätään.
- Opiskelijalle luodaan personalisoitu sähköpostiosoite. Osoite on "etunimi.sukunimi" muodossa ja "@tuni.fi" päätteinen. Jos tietokannasta löytyy useampi samanniminen, lisätään sukunimen perään numero, sen mukaan montako saman nimistä löytyy. Nimistä poistetaan heittomerkit, jos niitä löytyy (esim. Ronald Mc 'Donald, ronald.mcdonald@tuni.fi) Sukunimi poistetaan, jos se olisi sama kuin etunimi.

Esi- ja jälkiehtoja sisältävät luokat

Kyseiset luokat ja niiden riippuvuudet on esitetty laajemmin kuvassa 1. Alla on esitetty suhteet tiivistetyssä muodossa.

- StudyModule: riippuvainen Course-luokasta.
- Module: riippuvainen StudyModule-luokasta
- Degree: riippuvainen Module-luokasta
- Attainment: riippuvainen Course-luokasta
- Student: riippuvainen Degree-luokasta, Attainment-luokasta sekä Course-luokasta
- MainGui: riippuvainen LogInGui-luokasta, HomeTab-luokasta, DesignTab-luokasta, CourseTab-Luokasta sekä PersonalTab-luokasta
- NewStudentGui: riippuvainen LogInGui-luokasta sekä MainGui-luokasta
- LogInGui: riippuvainen NewStudentGui-luokasta sekä MainGui-luokasta
- Main: riippuvainen Student-luokasta, Degree-luokasta sekä LogInGui-luokasta.

Kaikki Gui-luokat ovat riippuvaisia myös Student-luokasta, Course-luokasta sekä Degree-luokasta.

Sovittu ja toteutunut työnjako

Alla olevassa taulukossa on sovittu ja toteutunut työnjako tiivistettynä.

Kaavio 1: Työnjako

Nimi	Onni	Ronja	Ville
Sovittu	Päävastuu rakenteiden ja luokkajakojen sekä GUI:n suunnittelusta, sekä luokkien testauksesta.	Päävastuu GUI:sta ja siihen liittyvistä luokkien luonnista.	Päävastuu tietojen hausta Sisun API:sta ja niiden tallentaminen ohjelman rakenteisiin.
Toteutunut	Suunnittelu toteutunut sovitulla tavalla. On toiminut yleisenä apuna	GUI:n luokat luotu sovitusti. Ikkunoiden ulkoasujen ja koodin rakenteen suunnittelu	Haku API:sta on toteutettu tyydyttävällä tasolla, API haku on

	ohjelman koodaamisessa ja siistimisessä. Testaus tehty sovitusti.	tehty yhdessä Onnin kanssa. GUI:ssa vaaditut muut luokat kuten Student.java, Degree.java luotu ja suunniteltu.	toteutettu sen verran järkevästi, että ohjelman suorittaminen on melko nopeaa. Loi myös Degreessä olevat API hakujen toteutukset, kuten rekursiiviset funktiot.
--	---	--	---

Käyttöohje

- Ohjelman käyttäjällä tulee olla koneellensa ladattuna ja ajoin lisättynä JavaFX plugin.
- Ikkunoita ei voi säätää eri kokoisiksi.
- Jos käyttäjä on jo käyttänyt ohjelmaa, voi hän kirjautua luomallaan opiskelijanumerolla sisään. Jos ei, tulee käyttäjän luoda uusi käyttäjä "Uusi oppilas" napin kautta.
- Oppilaan nimen tulee alkaa ja päättyä kirjainmerkkiin, mutta se voi sisältää tiettyjä erikoismerkkejä kuten "ä", "'" ja "-".
- Opiskelijanumero toteuttaa Tampereen yliopiston käyttämiä muotoja eli HXXXXXX tai XXXXXXXX, missä X on numeromerkki.
- Opintojen aloitusvuosi tulee ilmoittaa lukuna väliltä 1960–2022.

HUOM! Pääikkunan avautuessa saattaa syntyä pieni viive!

- Pääikkunassa käyttäjä voi ilmoittaa suorittamiaan kursseja valitsemalla kurssin "Omat suoritukset" sivulla ja painamalla "Valitse" nappia.
- Käyttäjä ilmoittaa kurssin arvosanan numerona 1-5 välillä ja painaa "Lisää" nappia.
- "Kurssinäköymä" sivulla käyttäjä voi vaihtaa tutkinnon vaihda napilla (HUOM! Tässä kohtaa saattaa syntyä pieni viive).
- Käyttäjä voi kirjautua ulos takaisin kirjautumissivulle "Omat tiedot" kohdan alta löytyvästä napista.

Ongelmat ja puutteet

- Ohjelma ei toimi, jos API on alhaalla tai internet ei toimi.
- Ohjelma ei huomioi AnyModuleRuleja sekä AnyCourseUnit -elementtejä API:sta, koska näille ei löytynyt järkevää käyttötapaa.
- Jotkin moduulit saattavat jäädä tallentamatta kurssien lukujen yhteydessä, jolloin kurssit ovat suuremman kokonaisuuden alaisuudessa.

Linkit

- Drawio: https://app.diagrams.net/#G1VXQ3aAd5tMgvfsfe6lcM4TLJ5babsD_R
- Trello: <https://trello.com/b/BK7nfwbV/ohj3-projekti-coding-monkeys>