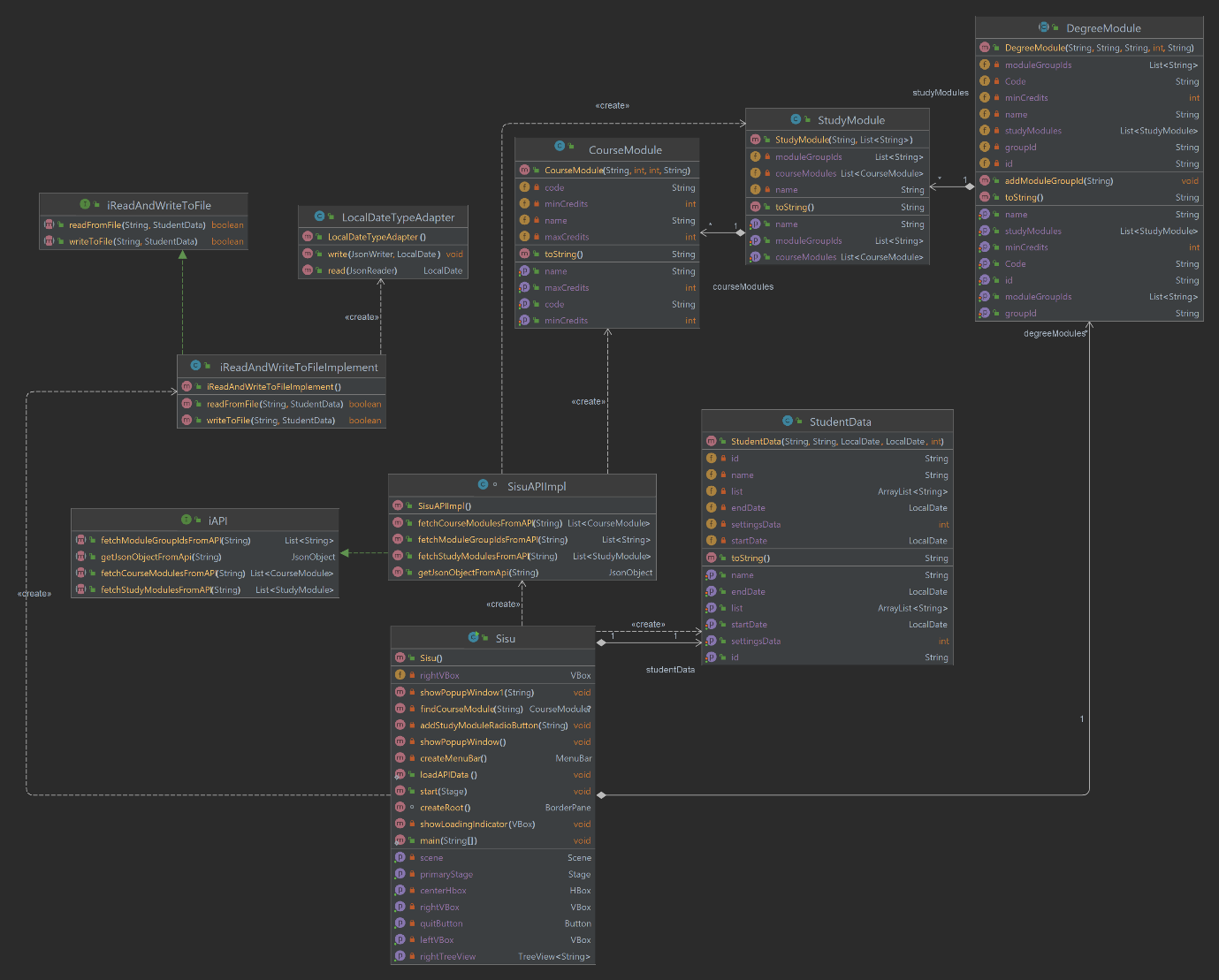
**Project Documentation Programming 3.**

**1. Project Structure UML**

****

**2. Responsibilities of the key classes**

First, we have the Sisu.java which is the main class of the project where all the user interfaces are implemented on this specific class.

We have SisuAPIImpl Class that implements iAPI Class it has methods to fetch and process required data from SISU.

getJsonObjectFromApi(String urlString) this method takes a URL as an argument and returns JSON data the following URL is passed

https://sis-tuni.funidata.fi/kori/api/module-search?curriculumPeriodId=uta-lvv-2021&universityId=tuni-university-root-id&moduleType=DegreeProgramme&limit=1000

as an argument which is the API for degree modules of SISU, the degree modules are processed with and a groupdIdsList of each degree module is stored in the DegreeModule.java Class

then we call fetchModuleGroupIdsFromAPI() method of SisuAPIImpl and recursively process study modules in each degree module using the stored groupdIdsList data of each degree module using the following API URL

https://sis-tuni.funidata.fi/kori/api/modules/by-group-id?groupId="+degreeModules.get(i).getGroupId()+"&universityId=tuni-university-root-id

This is how we get study modules in each degree program; the study modules are stored in StudyModule.Java Class along with moduleGroupIds

The module groups are recursively processed to fetch the courses for each study module using the following method in SisuAPIImpl Class

fetchStudyModulesFromAPI(String apiUrl)

DegreeModules class stored all degree prorgams available at SISU api, the study modules class stores all study modules available at SISU API and then fainlly the CourseModules class stores all courses available for each study module at SISU API this is all very complex and recursive process as mentioned above.

Then we have studentData.Java Class that stores studen's name, ID start and end date along with selected courses of the student,

we use this class as a model for iReadAndWriteToFileImplement.Java which implements iReadAndWriteToFile.Java class that has methods to store

and load student object to and from JSON file student\_data.json

then we have LocalDateTypeAdapter that is a custom class built for GSOB to parse dates for student\_data.json file

Finally we have our main class Sisu that holds all over GUI elements like treeView that holds degree programs and the tree like study data fetched from SISU API, this class also has all the methods to handle user operations like saving data to file loading from file setting to change the number of degree programs to be fetched.

**3. Project Functionality**

The project functionality was shown in a way where student could log in into their own student account by login in with their credential’s information starting with name, student number, start year of study, and end year of study. After they managed to log in on their account, the students were given 20 list of degree programs that are fetched from the SISU API by default, but if they wanted to find more degree programs, they could set the result by clicking onto the File->Settings->Set Limit on how many lists of degree programs that they wanted to view. After the system had fetch the data, we need to refresh the GUI by clicking the refresh button on the panel which will refresh the window and displaying an updated degree program. Students could now proceed on selecting the degree of programs that they wanted to choose and after they had finished selecting their choice of degree and courses, they should click File->Save in order to save the selected choice in a form of JSON.

**4. Division of work**

Charles = UML, GUI, API, Documentation

**5. User Manual**

Before running the project, make sure to export the necessary JAR Files into the library. There are four jar files that are attached within the folder project. Starting by opening the project first with ‘idea64’ and then in the project folder right click -> open module settings -> select sisu -> add dependancies -> add the needed jar files -> apply

After inserting the necessary JAR Files, proceed on running the project and the first that pops on your screen is the login page. Enter your name, student number, start date and end year. After you had filled out all of the necessary credentials you will then be redirect to the main page where you can see a tree view of the degree programme where by default it only shows you for 20-degree programme. You could also unfold the tree structure by clicking on the module. If you want to search for more degree programme, you could follow these instructions by clicking ‘File’->’Settings’ and just type down the amount of list of degree programme that you want to view. After you click the set limit, you need to refresh the window page in order for the page to be updated with the limit that you have set.

You can look into the courses based on the degree that you choose by clicking the dropdown button beside the degree list and you could select it by clicking it twice. Then you could save the study structure that you have selected by clicking ‘File -> Save’. Students could log in with their same credentials and they could just load the their data by clicking ‘File -> Load’.

**6. Known bugs or missing features**

Well, here are some known bugs or missing features in our project such as:

* When double clicking the degree that is shown on the panel, it doesn’t hover down and shows the courses that could be selected based on that degree, but it automatically adds the degree to the list of courses that we wanted to choose.
* The bullet option is not working properly to confirm the selected courses. So, by clicking twice in the selected courses would add the courses to the study structure.
* Unit testing isn’t implemented in yet.