DOCUMENTATION
Project: SISU Unraveled

By Diep Ngoc Luong, Nghi Khanh Quyen

I. Introduction

- The Sisu Unraveled project is an innovative program designed to provide students with a more efficient and user-friendly credit management system at Tampere University. As many students have found Sisu to be problematic, causing difficulties and frustration, our team recognized the need for a solution.

- Our program is designed to allow students to investigate the degree structures of study programs offered at the university, while also providing a simple way to monitor their progress as they complete courses and work towards their degree. Students have the freedom to define their own study program, mark completed courses, and view their progress over time.

- By utilizing Kori API, our program can retrieve information on all Tampere University degree programs, making it easier for students to calculate the number of credits required for their selected program. With Sisu Unraveled, students can now manage their degree progress with ease and confidence, without the added stress of dealing with Sisu's problems.

- We believe that Sisu Unraveled will revolutionize the way students approach their degree management, providing them with a reliable and effective tool that streamlines the process and ensures success.

II. Program use

- The purpose of the Sisu Unraveled program is to provide students with a reliable tool that allows them to easily track their study progress. The program is designed to answer critical questions regarding the student's degree, such as which degree program they are following, which courses have been completed or are planned, and how far they are into their studies.

- Upon launching the program, users will be directed to the Student Information tab, which features TextBoxes for the student's name and student number. These fields can be completed to save the information for later use, although this is optional. If the user chooses to save their information, they must press the Load from database button to load their data. If they opt not to save their information, or if this is their first time using the application, users can select their Academic Year and Degree Language Combo Boxes. Once all required fields have been completed, users can click on the Begin button to proceed.

- The application will then move to the Structure Studies tab, where users can choose their Degree and Track Combo Boxes. These options are already synced with the selected year and language. The program will display a study module in TreeView that contains a DropBox of courses on the left. When users click on the study module, the

TickBoxes of courses will appear on the right side. Users can click on finished courses, and the TreeView will update and display the Progress Bar below.

- Finally, when the user is finished using the program, they can click on the save and EXIT button to close the application. If the user has logged in before, their study progress will be saved for future use. If not, the program will simply close without saving any data.

III. Architecture structure

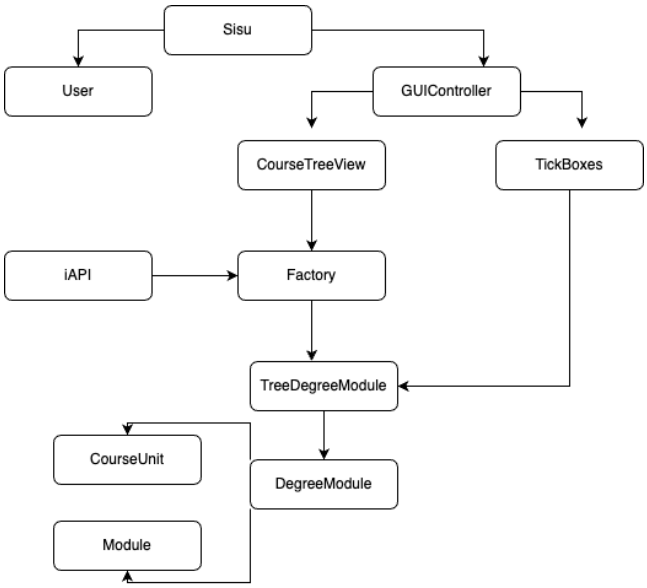

Figure A: The high-level architecture structure of Sisu project class

The architecture of the Sisu project is organized into two main packages:

- fi.tuni.prog3.module package includes four classes:
  - DegreeModule (an abstract base class).
  - Module: implement from DegreeModule class that represent other than course unit of module.
  - CourseUnit: implement from DegreeModule class that represent the course unit of module.
  - TreeDegreeModule. The degree structure is represented as a tree, with the DegreeModule as the base abstract class that contains several fields retrieved from Kori API requests, such as name, ID, and group ID. The TreeDegreeModule class stores a list of Module objects that represent the children nodes of the current DegreeModule object, thus connecting the node to its children nodes. This class is used to create the CourseTreeView of the degree structure and the TickBoxes for the classes.
- fi.tuni.prog3.sisu package includes nine classes:

- The Sisu class as the main class that handles the graphical user interface and input/output of the program.
- The GUIController class manages the contents inside Sisu's Combo Boxes, including the degree structure and course selections, displayed using the CourseTreeView class and the TickBoxes class, respectively.
- The CourseTreeView class displays the tree view of the degree structure after a user choses.
- The CourseTickBoxes class displays the check box for a user to select a course unit is completed.
- The Users class manages each user's basic information and selected courses.
- The Factory class calls all the necessary APIs to create the nodes needed for the program.
- The iAPI class, IreadAndWriteToFile, and SisuFileReader classes are utilized to help implement other functionalities of the program.

IV. Extra features

- In addition to the required features, our program includes several extra features to enhance the user experience and functionality of the system.
  - Implementing a hover-over feature for the Tick Boxes which displays additional information about the courses such as course names, course codes, number of credits, and grade scale. This data is obtained from the Kori API and is presented in a separate stage when the cursor hovers over the Tick Boxes, disappearing when the cursor is moved away.

  - Including the capability to save student information to an additional JSON file. The JSON file is structured as a map with the student number as the keys and the student information as the corresponding values. The student information includes the student number, name, degree ID, and chosen courses. This feature enables the program to read existing student information by prompting the user for their student number and retrieving the relevant data from the JSON file, which is then used to populate the appropriate fields in the user interface.

  - Implementing a feature to display the schedule of a student's progress in relation to the degree structure. This allows the user to view their progress in a clear and concise manner, helping them to plan their studies effectively and stay on track towards completing their degree.

  - Prioritizing a beautiful and user-friendly GUI and thorough documentation to ensure the best possible experience for the user. Furthermore, we have implemented a unit test suite to ensure the quality and stability of the program.

V. Limitation:

- The project has several limitations that need to be addressed:

- The program currently only supports English and does not have the option to switch to Finnish or other languages.
- The SisuTest class for the program can be further enhanced to provide more comprehensive coverage and ensure robustness, which includes the test cases for the user interface, which could be improved to cover more scenarios and detect potential errors.
- The program's stability may vary across different systems, which may affect its reliability and performance.
- Code efficiency can be improved to optimize resource usage and enhance the program's speed and scalability.
- The user interface still requires further refinement to enhance its usability, accessibility, and consistency with industry standards.

VI. Work division

- The team will collaborate on documentation, testing, and the interface of the application to ensure that the project meets the requirements and expectations.

- Dividing task: To efficiently manage the workload, the team has decided to divide the tasks as follows:

    - Diep Tran will be responsible for designing and handling the interface of the application, implementing any additional features, and designing the architecture of the module before implementing it. Diep will ensure that the user interface is user-friendly and easy to navigate, making it an enjoyable experience for users. Additionally, Diep will implement any additional features that are required to enhance the functionality of the application.

    - Nghi Quyen will be responsible for fetching the API and constructing the data structure required to implement the application. Nghi will work closely with Diep to ensure that the data structure is compatible with the interface and the overall architecture of the module. In addition, Nghi will be responsible for documentation and testing to ensure that the application is reliable and meets the project's specifications.

Together, the team will collaborate to ensure that the project is completed within the specified timeline and meets the expectations of the project. Regular communication and updates will be provided to ensure that the project is progressing as planned, and any issues will be addressed promptly.