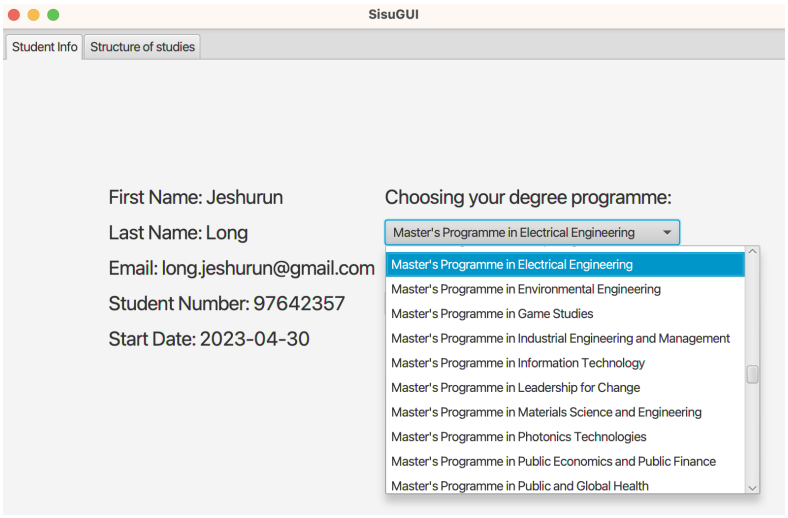
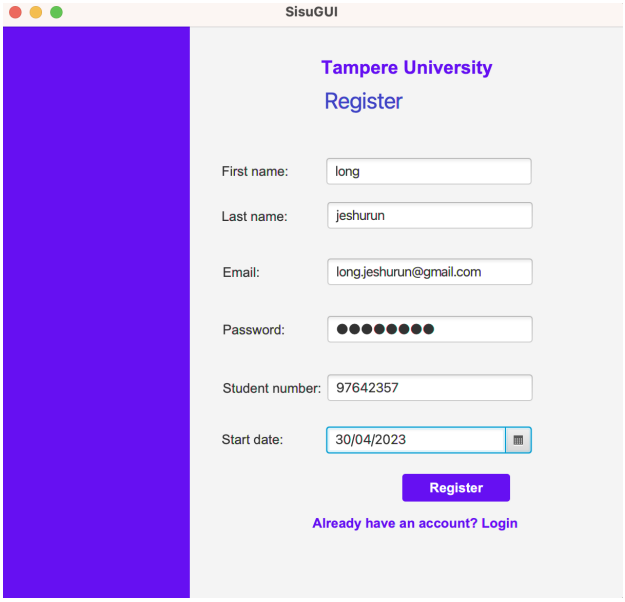
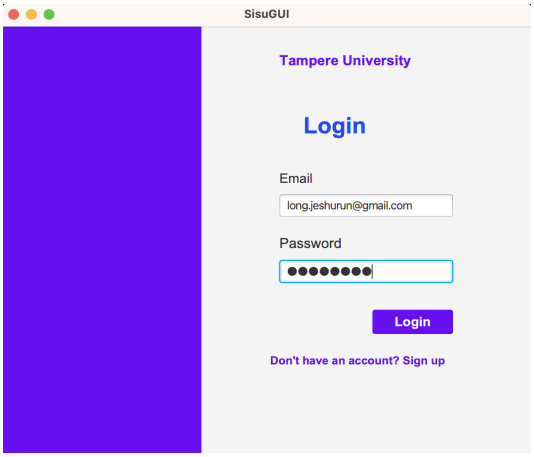
COMP.CS.140 PROGRAMMING 3 SOFTWARE DEVELOPMENT PROJECT

SISU Unraveled







This software project was authored by:
Nguyen The Long
Student number: 151317891,
Email: long.nguyen@tuni.fi
and
Ezeobi Jeshurun
student number: 151240272,
Email: jeshurun.ezeobi@tuni.fi

Table of Content:

1. Introduction:

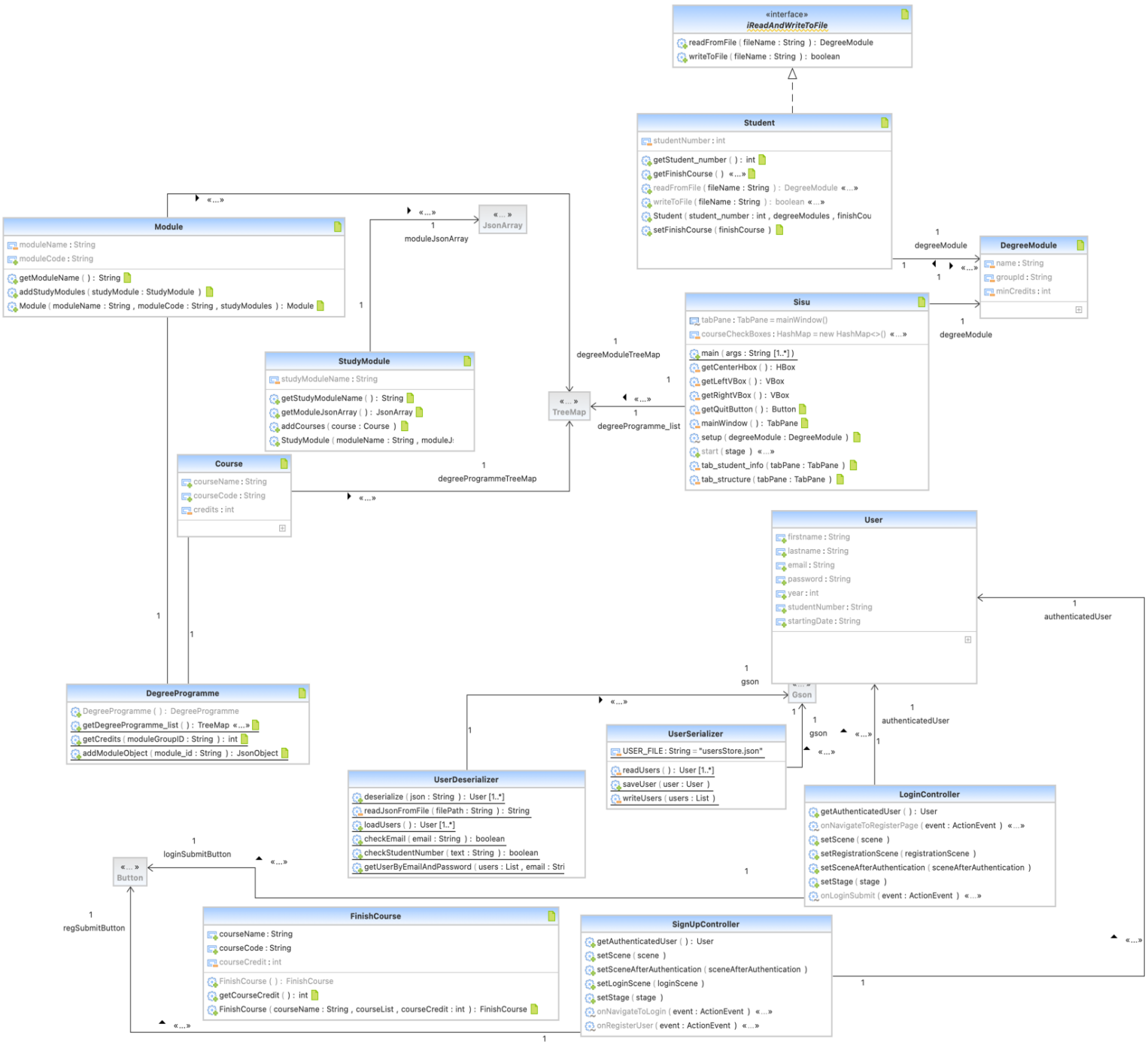
Our objective is to develop an optimized version of Sisu - a web-based student information system that simplifies administrative tasks, provides a user-friendly experience for students and staff, and offers various features for academic planning, scheduling, and reporting. Our aim is to enhance the system's functionality, efficiency, and performance while maintaining its seamless integration with other university systems such as the student admission system, degree audit system, and the student financial aid system. The optimized version of Sisu should help universities streamline their administrative processes and enable students to access their academic records easily and conveniently.

2. Scope:

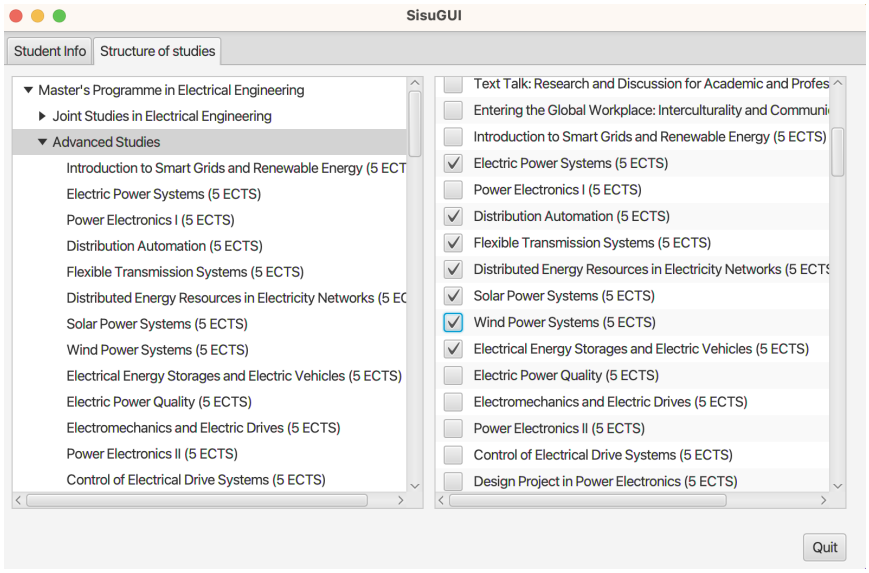
- Utilizing object-oriented design to create a Sisu management system.
- Incorporating external libraries into the implementation.
- Integrating a graphical user interface as a component of the program.

- Conducting unit testing on personal code.
- Collaborating with a team to develop the Sisu Unraveled software.

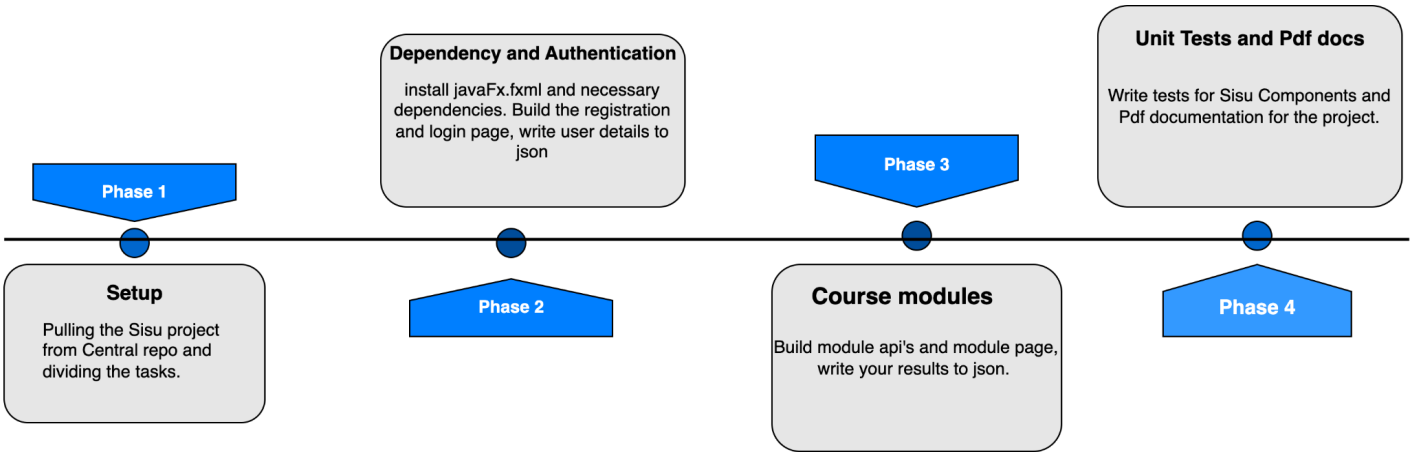
3.UML representation of the Sisu Application:



4. The Sisu Unraveled module Page:



5. Project Plan:



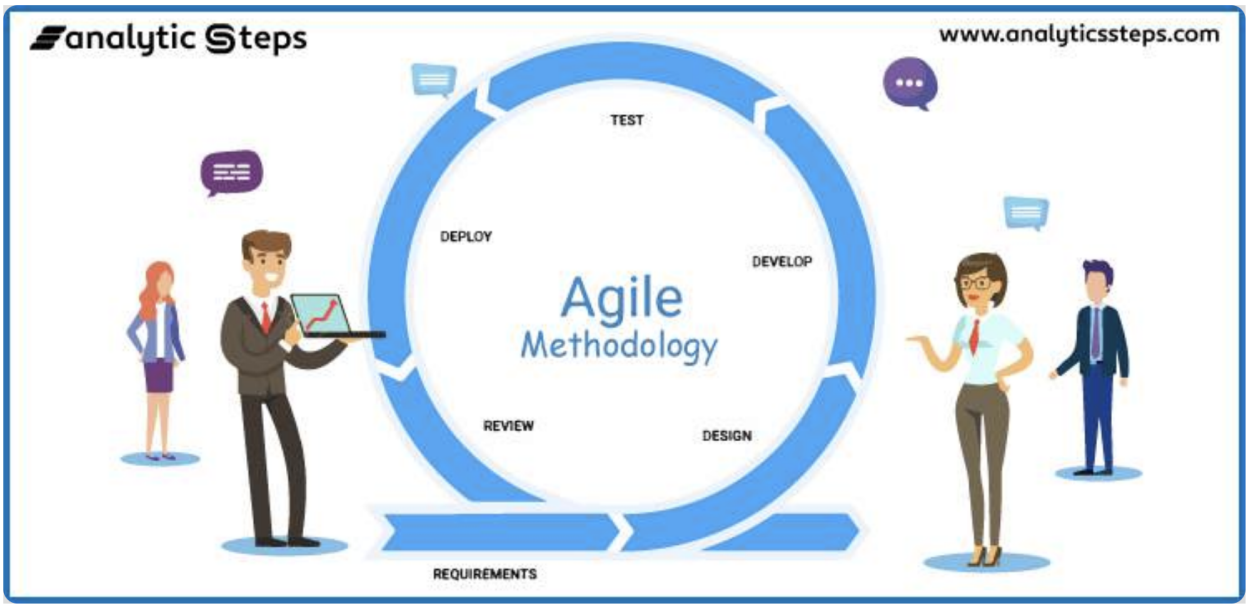
6. Software Features:

A Login page is shown to the user once the application starts. If the student has an account they can login, also the login page has a link which redirects the user to the registration page if they don't have an account. The login page has an email and password section while the registration page has these fields: firstname, lastname, email, registration year, password, automatically generates students number. The program will display the main window after providing the necessary initial information, which includes details on the study program and orientation, among other essential data. In this main window, the user has the option to modify the study

program and orientation. Additionally, the student's progress can be viewed in the form of completed courses, and the user can mark new courses as complete from this view.

To ensure the security of personal data, such as student information, a JSON file is employed to manage this information within the program. When the user clicks the program's exit button, any progress made by the student is saved to the JSON file.

7. Agile life Cycle Model was used:



Picture credit: www.analyticssteps.com

This helped us iterate through the project, improving performance and functionalities in each iteration.

8. Packages and Dependencies Used:

org.openjfx:javafx-controls:13: This dependency is used for JavaFX controls, which provides UI components like buttons, text boxes, etc.

com.google.code.gson:gson:2.9.0: This dependency is used for Google's Gson library, which is used to convert Java objects into JSON and vice versa.

org.openjfx:javafx-fxml:13: This dependency is used for JavaFX FXML, which provides a markup language for defining user interfaces.

com.google.code.gson:gson:2.8.8: This dependency is also used for the Gson library but with a different version.

The `build` section of the POM file defines the build instructions for the project, including the plugins used for compiling and running the code.

org.apache.maven.plugins:maven-compiler-plugin:3.8.0: This plugin is used for compiling the code with the specified version of the Java compiler.

org.openjfx:javafx-maven-plugin:0.0.4: This plugin is used for building and running the JavaFX application.

- The mainClass element specifies the main class of the application.
- The executions section defines the different execution configurations for the plugin.
- The default-cli execution is used for running the application from the command line.
- The debug execution is used for attaching a debugger to the application.
- The ide-debug execution is used for debugging the application from within an IDE.
- The ide-profile execution is used for profiling the application from within an IDE

9. Additional Features:

1. Unit tests have been implemented for the graphical user interface. Use the TestFX framework for testing JavaFX programs.

2. Student settings. At the start of the program a settings window is opened and the student using the program can be set. The situation for the degree is read and stored in a JSON file.