Tampereen yliopisto

DOCUMENTATION

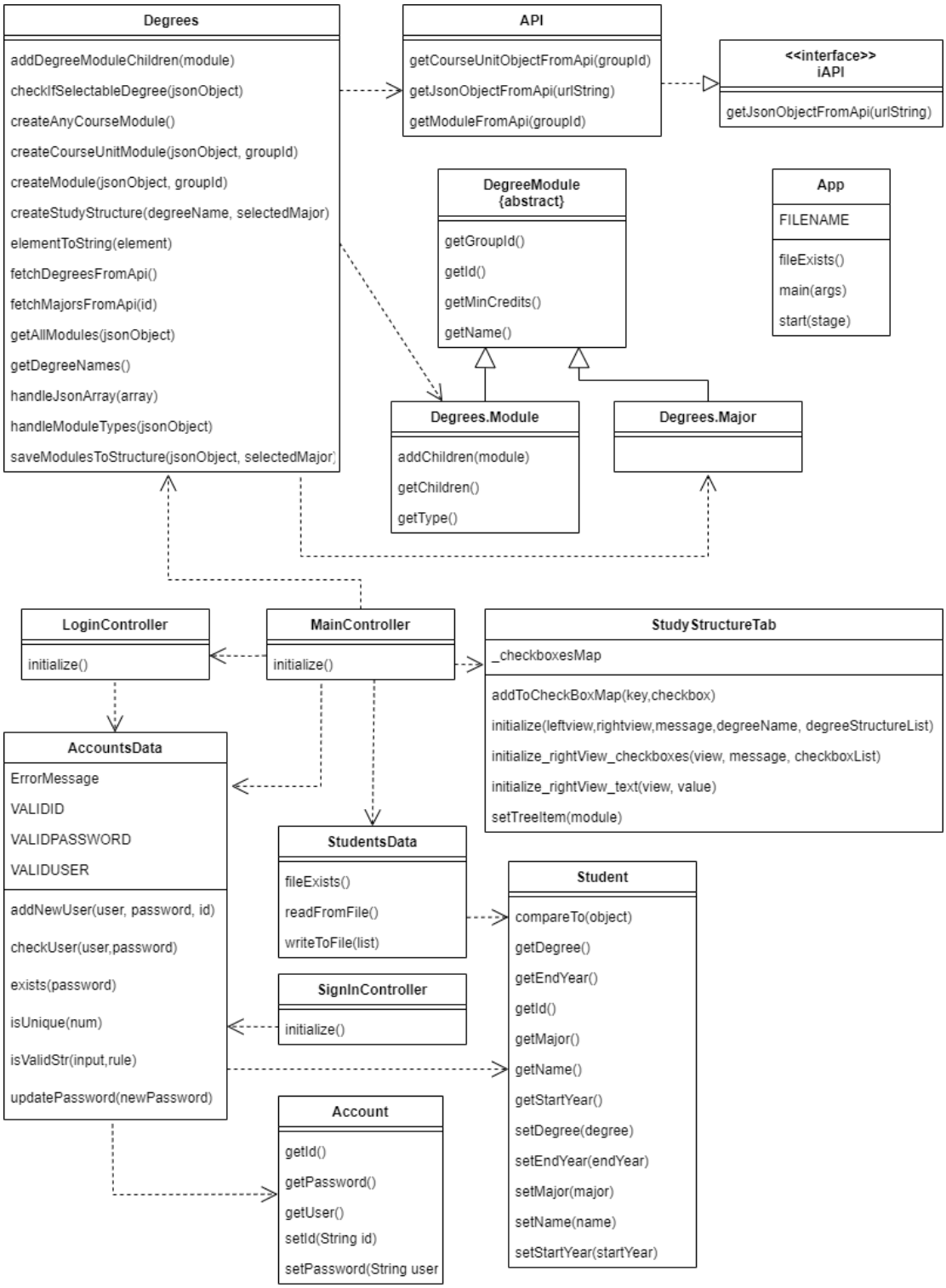
COMP.CS.140 Programming 3: Interfaces and Techniques

Heidi Seppi
Shuang Fan
Venla Viertola

TABLE OF CONTENTS

1. SOFTWARE STRUCTURE

Below is shown the class diagram which represents the structure of this program. There is also JavaDoc documentation available to be able to inspect the classes.



2. KEY CLASSES AND THEIR RESPONSIBILITIES

API	Implements iAPI interface. Fetches data from Kori API
Account	Stores user's account information such as id, username and password
AccountsData	Contains methods for calling Account class actions
DegreeModule	An abstract class for storing information on modules and courses
Degrees	Contains information about Sisu degrees. Saves all degree and major options and updates degree structure accord to user's choices
Student	Stores user's study information such as student number, name, starting year, expected graduating year, and studying degree
StudentData	Class is used to define student's basic information, and reading and writing updated information to students.txt file

3. PROJECT FUNCTIONALITY

The program fetches data from Kori API and updates the graphical user interface according to user's choices. User is able log in or create a new account in the signing in page. After logging in, the program shows user's information. If the user account is new account, user needs to update their information first before being able to check the study structure tab. If the user's account already existed in the system, the program displays user's current information (name, student number etc.) and user is able to change their details every time they want. User can display selected degree's study structure and choose courses using checkboxes. The program shows each study group's courses as checkboxes when user clicks on the group name. If group name is just main title, the program displays just the title name. User's selected courses will be saved to the system after saving with user's details. User can log out and log back in with different accounts without closing the whole program.

Minimum requirements:

- The program compiles
- The program contains one dialog window (e.g. the initial dialog)
- The program is able to show the selected study program structure in the main window
- The program uses inheritance in implementing the study program structure

Intermediate requirements:

- A graphical user interface has been fully implemented by the team (the simple readily provided JavaFX project is not used)
- The program gets the degree structures from the Sisu API and utilizes the provided interface in implementing the API handling
- The program works i.e. the degree shown can be changed and the progress situation of the student's degree is shown.
- The program handles errors in file handling.
- Unit tests have been implemented for the program

Additional features:

- **Student accounts management**

The program can create a new account for a new student. The required student number (id) cannot be the same to make sure one student only has one account to manage their own study. But the username can same.

- **Student study storing management**

The students' study can save to a file and read again when the student login again later. And the student can save their newest study result as they wish.

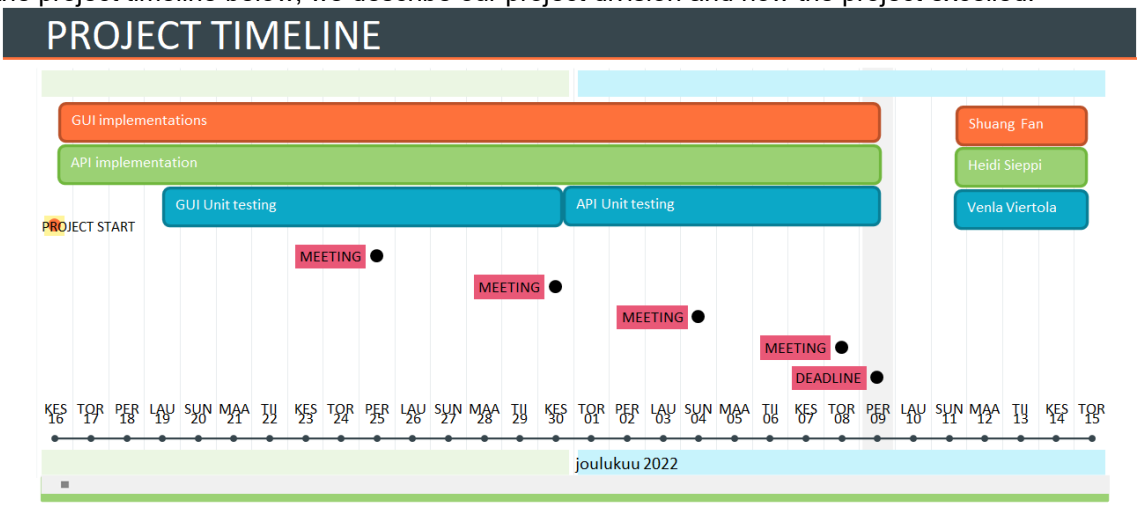
Classes containing pre and post conditions

Class	Pre conditions	Post conditions
AccountsData		Account exists
API	Kori API responds	

4. DIVISION OF WORK

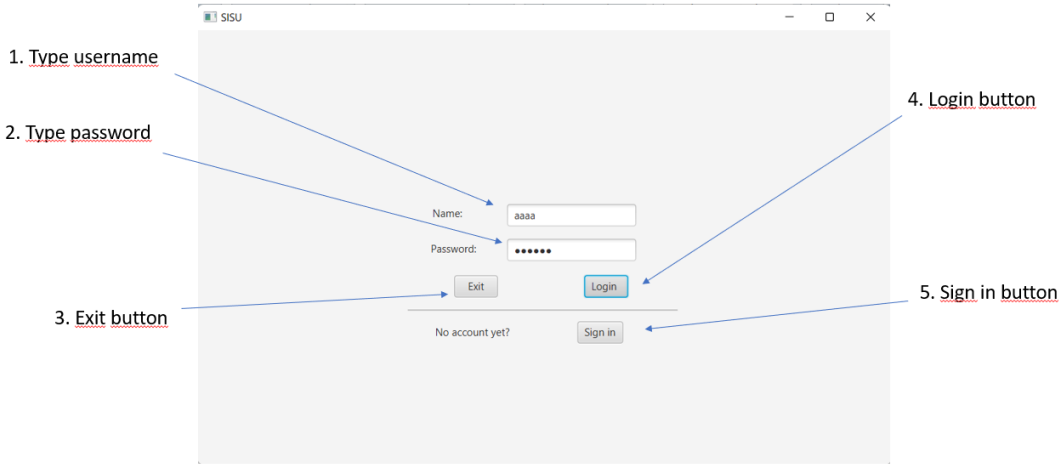
We divided our project to three categories, and these categories were divided amongst our group as follows: GUI, API and testing. Using Git we decided to work each on different branches and then after agreeing together about the changes in the each branch, we merged the branch changes to main branch. At the end of the project work GUI and API needed to be merged together so we also worked a bit on other's responsibility category.

In the project timeline below, we describe our project division and how the project excelled.



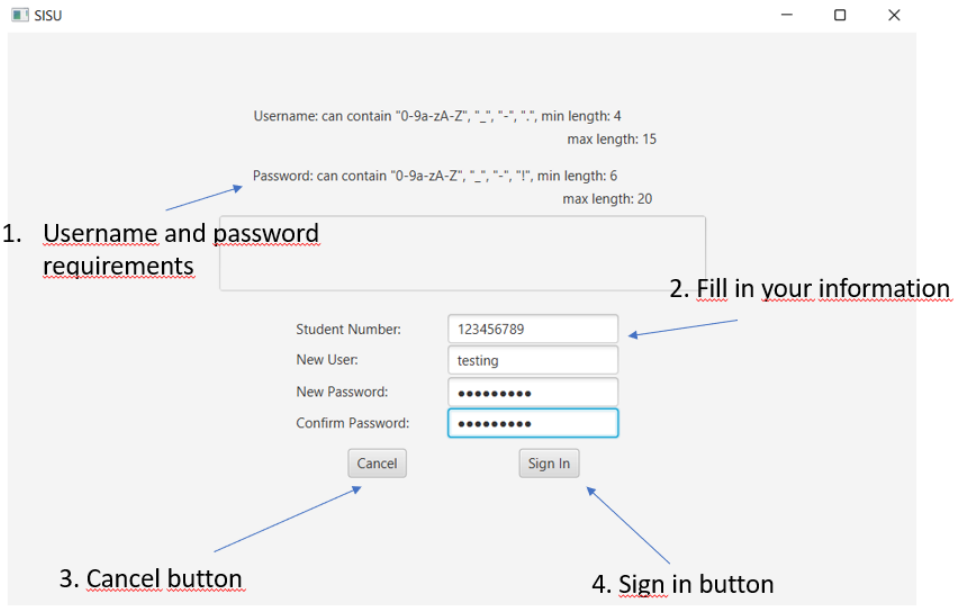
5. USER MANUAL

5.1 Starting screen



1. A field to type in your username
2. A field to type in your password
3. Exit-button that ends the program and shuts down the QUI
4. Login-button, by clicking that, you can proceed to the next window
5. Sign in -button, click here to redirect you to a sign in page to create a new account

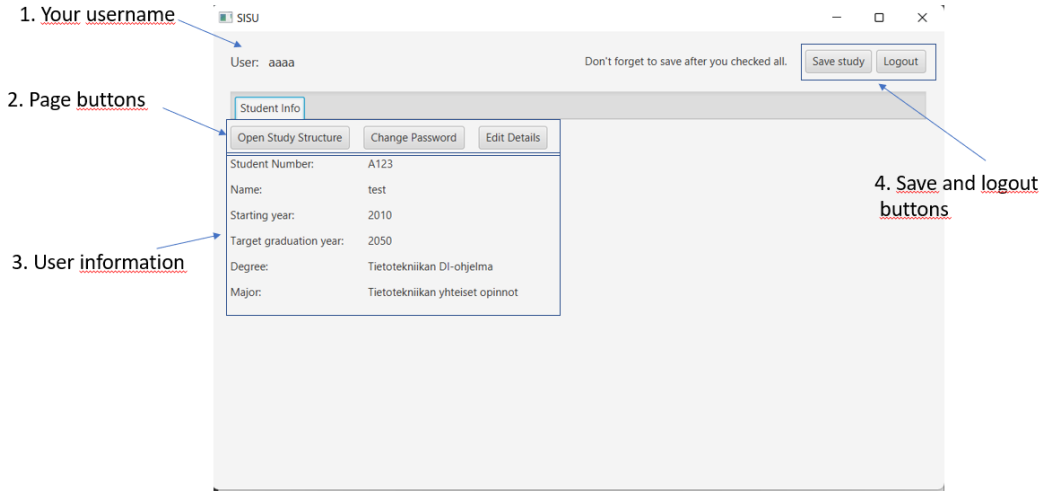
5.2 Sign in screen



1. Here are username and password requirements
2. Fields to fill in your preferred information (not that student number is unique and cannot be changed later)

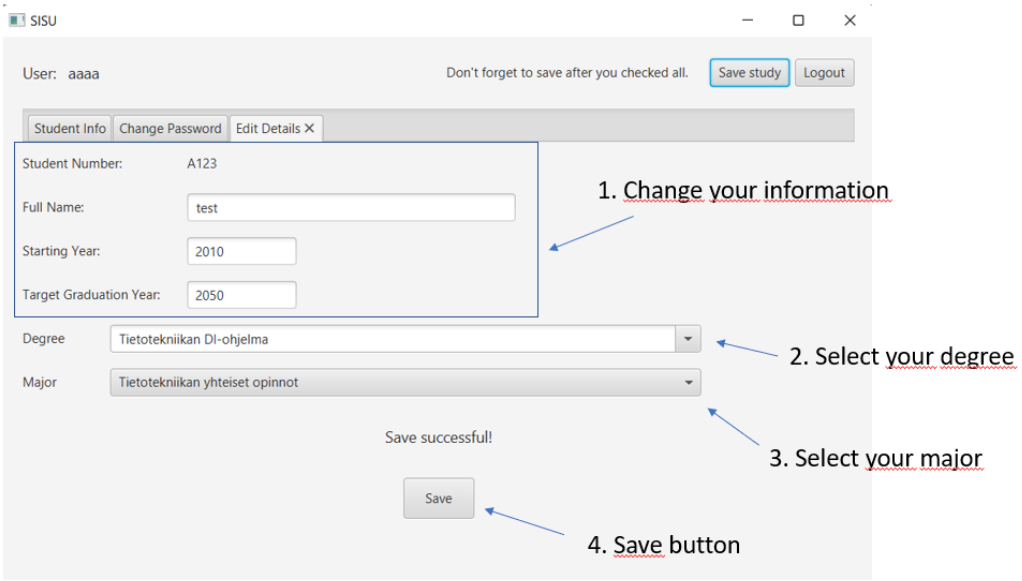
3. Cancel-button that redirects you back to the starting page
4. Sign in-button for saving your account if everything's correct

5.3 Student information screen



1. This is where your username is displayed
2. Clicking these buttons you can open different tabs
3. User information is displayed here
4. Save study and logout buttons, save your study information and logout (exits to starting screen)

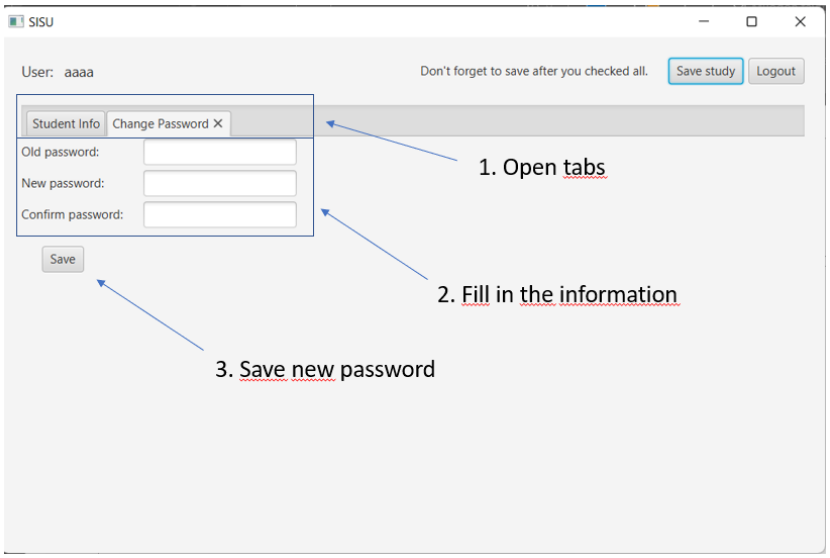
5.4 Edit details screen



1. Here you can change your information
2. Select your degree from the drop-down list or write the name of the degree. Degree name must be written correctly before program can recognize it.
3. Select your major from the drop-down list
4. Save information by clicking save-button

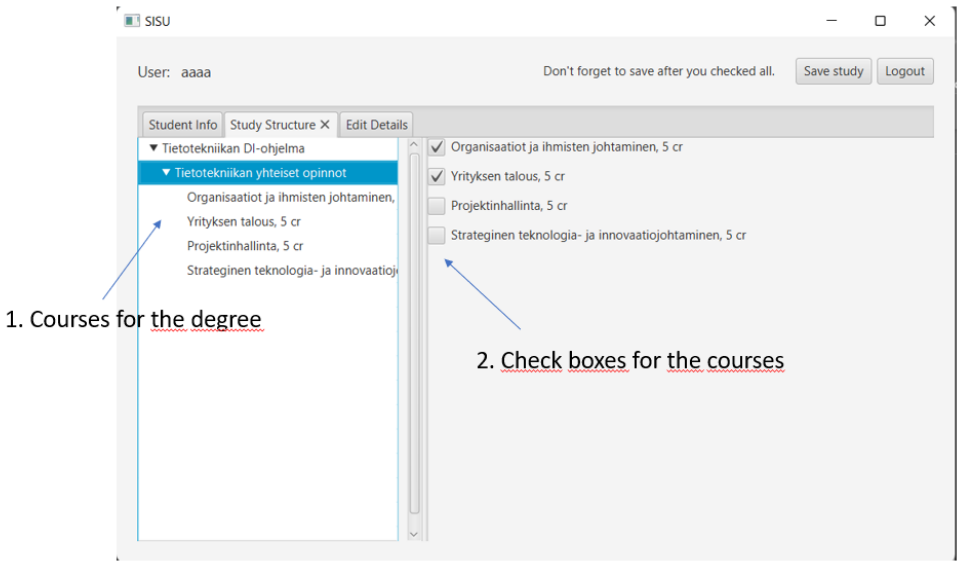
5. After saving is complete, program shows “Save successful”! text.

5.5 Change password screen



1. Here you can view opened tabs
2. Fill in the old and new passwords
3. Save-button saves the changed password

5.6 Study structure screen



1. Here you can view courses and programmes for your degree
2. Here you can check boxes of the courses you’d like to mark

6. KNOWN BUGS AND MISSING FEATURES

- Program doesn’t save all the major selection choices correctly. If some degree doesn’t have any selectable majors, program shows normal study modules (for example “tutkinto-ohjelman yhteiset opinnot”) as a major option under the “Edit details” tab in Major drop-box. This is because it was difficult to filter possible major options from Sisu’s data since we didn’t find any straight way to recognize what module is normal study module or major study module.

- When user logs in, it takes a long time to fetch data for main degree options, possible major options and selected degree’s structure if user has selected it previously. This is due to complex structure of Sisu degrees from Kori Api which caused functions on De-grees.java class to be complex. Functions are not as efficient as they’re supposed to be. When user changes their degree and major options and saves their choices, it also takes a while to fetch and save new degree structure depending on how large the selected degree and major structure is. Information about fetching and saving the data is shown in output window.

- Unit tests passed and are described in appendix

APPENDIX

APPENDIX 1: Test cases

Description	Test case	Input	Expected output	Real result
testGetJsonObjectFromApi	JsonObject not null	https://sis-tuni.funi-data.fi/kori/api/module-search?curriculumPeriodId=uta-lvv-2021&universityId=tuni-uni-versity-root-id&moduleType=De-greeProgramme&limit=1000	True	True
testGetModuleObjectFromApi	JsonObject not null	GroupID: otm-87fb9507-a6dd-41aa-b924-2f15eca3b7ae	True	True
testIsValidStrUpper	Uppercase string	String input = "ABCDEFG"	True	True
testIsValidStrNumer	Numeral string	String input = "1234567"	True	True
testIsValidStrLower	Lowercase string	String input = "abcdefg"	False	False
testIsValidStrSpecial	Special characters	String input = "__--!!"	True	True
testIsValidStrFalsey	Falsey input	String input = "HJKLÖÄ"	False	False
testFileExistsTrue	File is found	file	true	true
testReadFromFile	Reading is not null	file	true	true