Knowledge Traceability in Higher Education

Tommy Andersson - anetom-6@student.ltu.se

Robin Danielsson - robdan-7@student.ltu.se

Marcus Eriksson - amueri-6@student.ltu.se

Wilma Krutrök - wilkru-7@student.ltu.se

Jesper Nilsson - ejeino-7@student.ltu.se

Johan Rodahl Holmgren - ojaohe-3@student.ltu.se

Introduction

This report is part of an assigned project in the course "*Project in computer science and engineering*". The purpose of the project is to give students, teachers and student counsellors a tool for viewing and evaluating information about the knowledge given in different courses. The tool itself will hopefully provide students with a wider understanding of how different courses are connected and why parts of them are important for future studies. Teachers will be able to see how changing a part of a course would affect other courses given to the same students.

Background

Project owner Jan van Deventer stated a problem with students not having the knowledge needed when the course starts. To solve this problem Jan wanted a software for teachers to easily see what the students know and how long time has passed since they took a course with a specific "knowledge component". The software should also be used by students to keep track of the importance to learn specific parts of courses.

One attempt to solve this problem has been done. The solution was using Google Sheets and Matlab which turned out to be very inefficient. One big challenge was that teachers named the knowledge components differently, resulting in chaos. The program was not able to see if something was misspelled, used shortening or other similar things.

Problem description

The main problems are divided into different parts. One is that some courses overlap while some courses have a missing link. Prerequisites is not obvious between teacher nor is what is to be taught for future courses. When interviewing the teachers it became apparent that most of the teachers had to do some changes in their teaching because students did not have the knowledge needed. Some of the reasons is that students had not learned it or the course had not included it.

Assignment

To solve the problem a software will be implemented. The software will be web based and interactive. Users should be able to search for programs and courses to see how the courses are connected and what knowledge component each course require and give. This will motivate the user group students, and help all user groups to plan the optimal course order. Teachers should be able to see what knowledge components they should teach and on what level.

Delimitation

Because of the limited time set the project group will not focus on collecting the data that is needed for the solution to work. This job will be placed on the examiners together with a group of engaged teachers. The system will not be fully integrated with the system the university is currently using.

System Design

The software is planned to be implemented as a Java program. In that way classes and objects can be implemented. Java also got the libraries needed to communicate with a database. Courses and knowledge components will be implemented as objects with different parameters. Courses will be connected to knowledge components containing the taxonomy level. It will also make it possible to make the connections between the different objects clearer and easier to change.

All the data about the courses and the knowledge components will be stored in a Neo4J[1] database with the same structure as in the node objects. The reason for choosing Neo4J is because it is fast with graph relationships and the way it is constructed makes it easy to create and connect nodes. Storing the data in a database will give a good overview and access to the data. Connection to the front-end program from the Java program will go through a server.
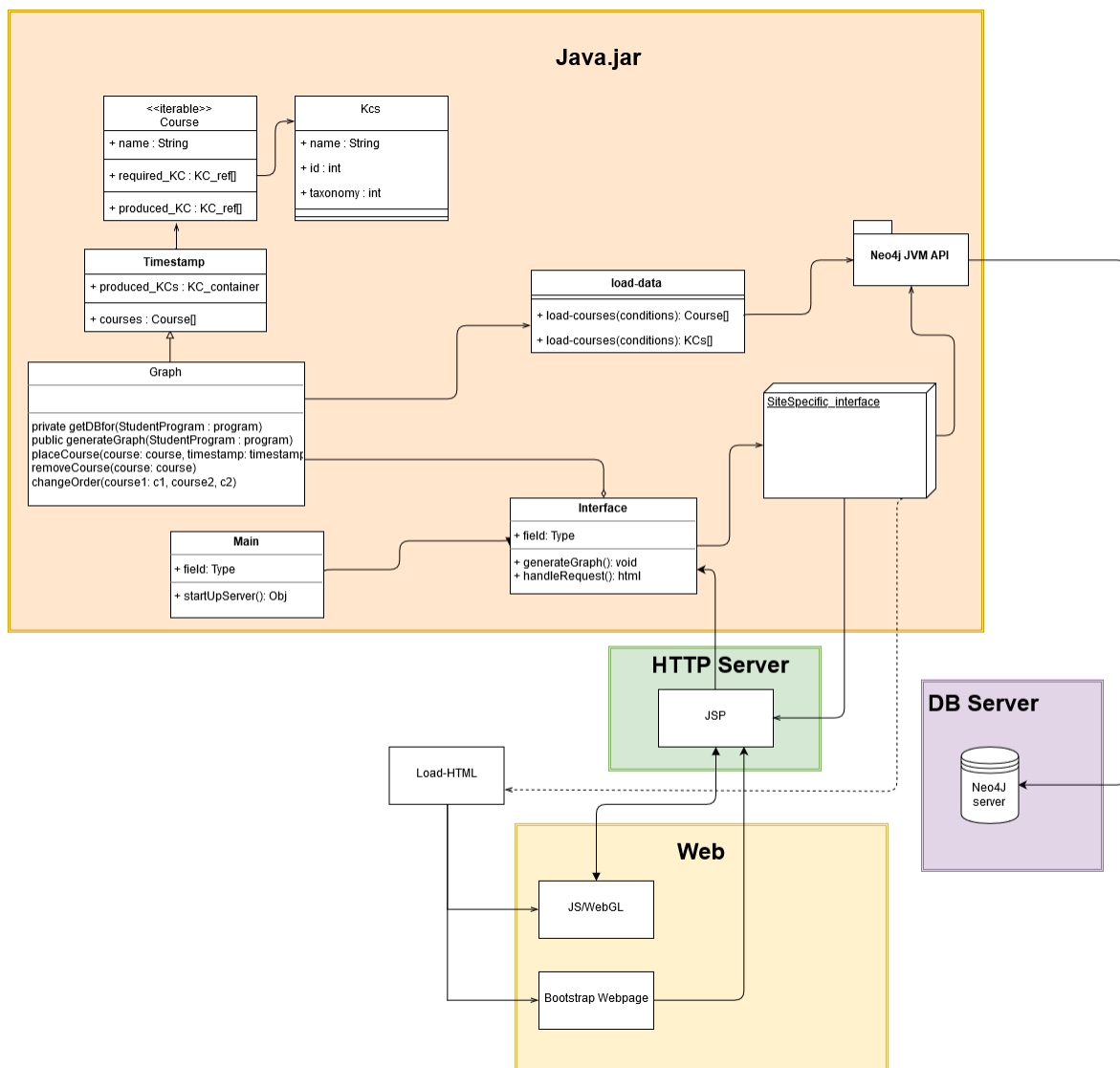
Figure 1: A first design of the UML-diagram.

Figure 1 is an overview of our system design. The user will access the system using a web browser. The Java program will contain a JSP (Java Server Pages) class running in a Tomcat container and connect to the web browser using HTTP requests. All database communication will be handled by the Java program. Courses and knowledge components will be represented as nodes in a graph. Using this method it will be easy and fast to make changes in the graph and return the changes back to the user.

1: Neo4J website. 2019-12-11. https://neo4j.com