**Assignment**

**JSF Web Application with MySQL and Neo4j Databases**

Contents

[Introduction 3](#_Toc530675076)

[MySQL Database 3](#_Toc530675077)

[Neo4j Database 4](#_Toc530675078)

[Marks 4](#_Toc530675079)

[Marking Scheme 4](#_Toc530675080)

[Plagiarism 4](#_Toc530675081)

[Skeleton 5](#_Toc530675082)

[Submission of the Project 5](#_Toc530675083)

[Overview of Web App 5](#_Toc530675084)

[Main Page 5](#_Toc530675085)

[Manage Courses Page 6](#_Toc530675086)

[Add Course Page 7](#_Toc530675087)

[Course Student Details Page 8](#_Toc530675088)

[Delete Course 9](#_Toc530675089)

[Manage Students Page 10](#_Toc530675090)

[Add Student Page 11](#_Toc530675091)

[Full Student Details Page 14](#_Toc530675092)

[Delete Student 15](#_Toc530675093)

[Connection Problems 17](#_Toc530675094)

### Introduction

Write a JSF dynamic web application that queries and updates a MySQL database and a Neo4j database.

The MySQL database is called *studentDB* and can be downloaded from the *Project* section of Moodle for this module (*studentDB.sql*).

### MySQL Database

The *studentDB* database consists of 2 tables:

|  |  |
| --- | --- |
| **Course** | |
| **Column Name** | **Details** |
| cID | A code representing the Course e.g. “BUS”,”DIG MEDIA” etc. |
| cName | The Course name e.g. “B.A. in Business Studies” etc. |
| duration | The duration of the course in years e.g. 3 |

|  |  |
| --- | --- |
| **Student** | |
| **Column Name** | **Details** |
| sid | A code representing the Student e.g. “G0033333” |
| cID | A code representing the Course e.g. “BUS”,”DIG MEDIA” etc. |
| name | The name of the student e.g. “Aine” |
| address | The address of the student e.g. “Tuam” |

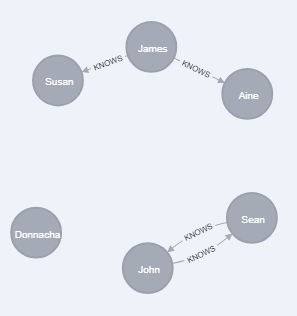
Full details of the Primary and Foreign Keys as well as datatypes can be found in the database itself.

### Neo4j Database

The Neo4j database can be downloaded from the *Project* section of Moodle for this module (*studentDB.db*) and should be imported into a **new** Neo4j database which contains no nodes.

When imported the Neo4j database should contain:

* 6 :STUDENT nodes
* 4 :KNOWS relationships



### Marks

This assignment is worth 50% of the marks for the module.

### Marking Scheme

85% of the marks will be awarded for implementing the functionality described in this document.

15% of the marks will be awarded for innovation and good programming practice.

Please describe your innovation (if any) in a document entitled *innovation.doc* which should be stored in the root folder of your project.

### Plagiarism

Plagiarism will be dealt with in accordance with the institute’s [Plagiarism policy](https://www.gmit.ie/sites/default/files/public/general/docs/3-5-plagiarism.pdf).

### Skeleton

A skeleton for this project (named GXXXXXXXX.7z) should be downloaded from the *Project* section of Moodle and used as a basis for this project.

### Submission of the Project

The zipped Eclipse project (named GXXXXXXXX.zip where GXXXXXXXX is your student number) should be uploaded to the *Project* section of Moodle no later than **Friday December 14th 2018 at 9:00am.**

### Overview of Web App

The web app should consist of the following pages:

#### Main Page

The *Main* page simply consists of 2 links:

* One to the *Manage Courses* page
* One to the *Manage Students* page

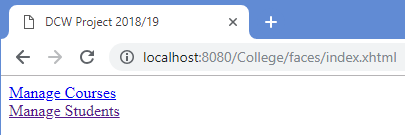


Figure 1 Main Page

#### Manage Courses Page

The *Manage Courses* page:

* Shows details of all Courses
* Has an *Add Course* button
* Has *Show Students* and *Delete* actions for each Course
* Has a link back to the *Home* page.

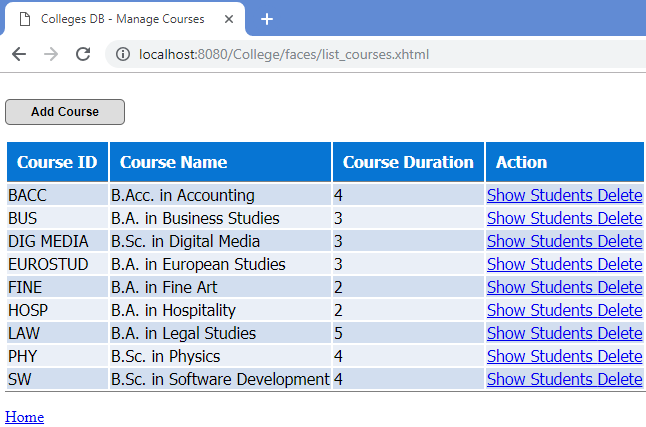


Figure 2 Manage Courses Page

#### Add Course Page

The *Add Course Page* allows the user to enter a Course ID, Name and Duration.

If an existing Course ID is entered an error should be displayed.

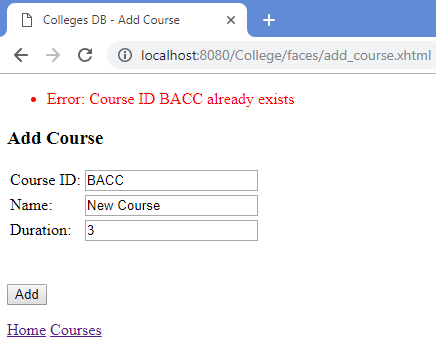


Figure 3 Course ID already exists

If Country ID, Name or Duration are not specified appropriate error messages are shown:

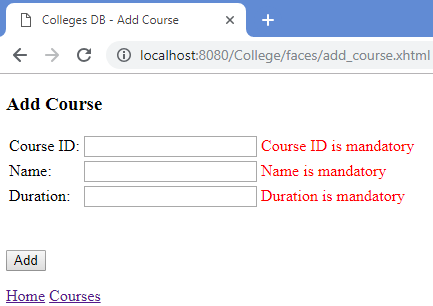


Figure 4 Data not entered

When a Course has been added successfully, the user is returned to the *Manage Courses Page.*

#### Course Student Details Page

When the *Show Students* action is pressed on the *Manage Courses* page, the user is brought to the *Course Student Details* page.

This page shows the *name* and *address* of all students doing with the particular course.

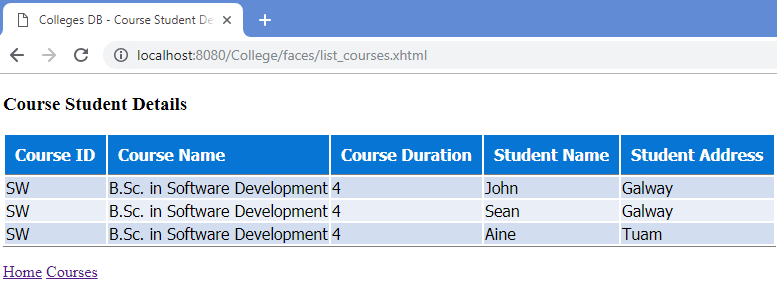


Figure 5 All students doing the course SW

If there are no students doing a particular course, nothing should be show:

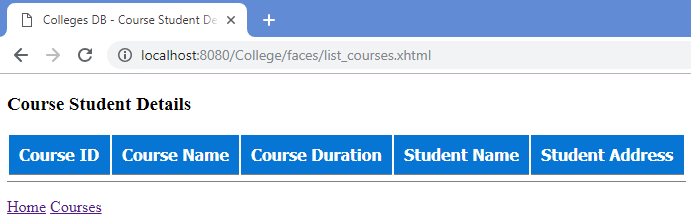


Figure 6 There are no students doing the course BACC

#### Delete Course

When the *Delete* action is pressed on the *Manage Courses* page, if the particular course to be deleted has no students, is deleted from the database and the user remains on the *Manage Courses* page which is updated to show the new list of courses.

If the particular course to be deleted has students, an error message is shown and the course is not deleted from the database.

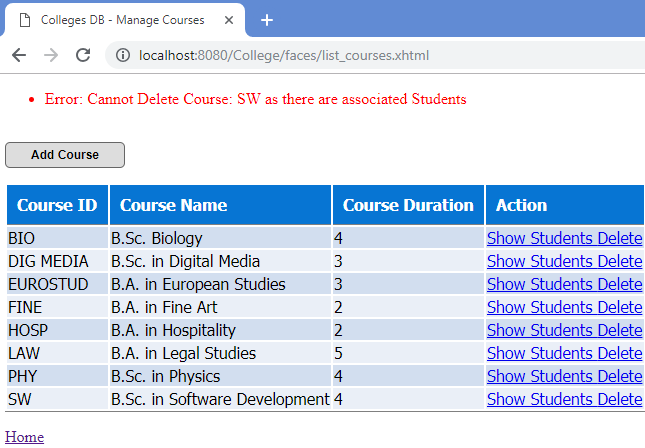


Figure 7 Cannot delete courses that have students

#### Manage Students Page

The *Manage Students* page:

* Shows details of all Students
* Has an *Add Student* button
* Has *All Details* and *Delete* actions for each Student
* Has a link back to the *Home* page.

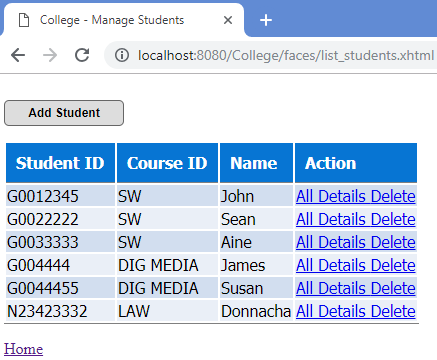


Figure 8 Manage Students Page

#### Add Student Page

The *Add Student Page* allows the user to enter a Student ID, Name, Address and Course ID.

If an existing Student ID is entered an error should be displayed.

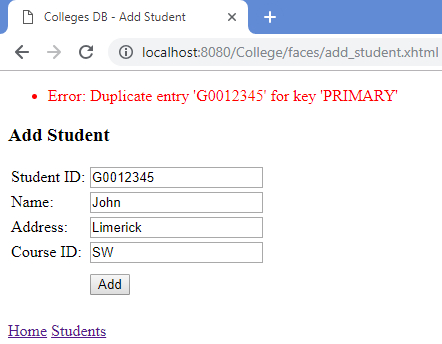


Figure 9 Existing Student ID entered

In the MySQL database, the Student name is unique, so if an existing name is attempted to be added to the database an error is shown.

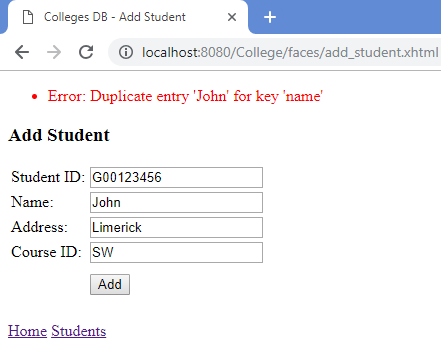


Figure 10 Existing Student Name entered

If a non-existent Course ID is entered an error should be displayed.

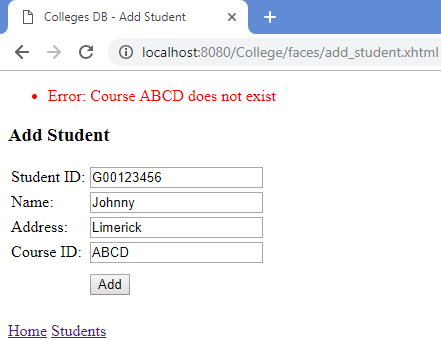


Figure 11 Non-existent Course ID

All fields are mandatory, except the address field, so if data is not entered appropriate error messages should be shown.

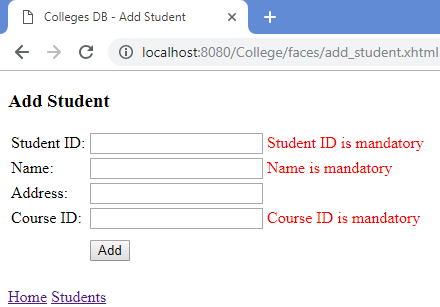


Figure 12 Mandatory fields not entered

When a Student is entered that satisfies all the above criteria, he/she is added to the MySQL database and the user returned to the *Manage Students* page, where the new student is shown:

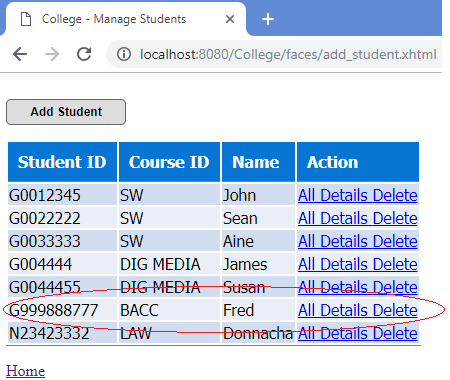


Figure 13 New Student shown

In addition, a new node with the label *:STUDENT* and an attribute *name* with the student’s name, and an attribute *address* with the student’s address is added to the Neo4j database.

(If no address was entered for the student, the node in the Neo4j database should still have an *address* attribute, but no value for it.)

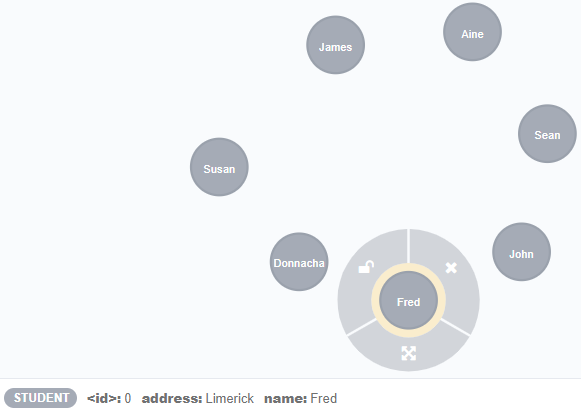


Figure 14 Student added to Neo4j database

#### Full Student Details Page

When the *All Details* action is pressed on the *Manage Students* page, the user is brought to the *Full Student Details* page, where the following information is shown:

* Student ID
* Student Name
* Course ID of the course the student is doing
* Course Name of the course the student is doing
* Course Duration of the course the student is doing

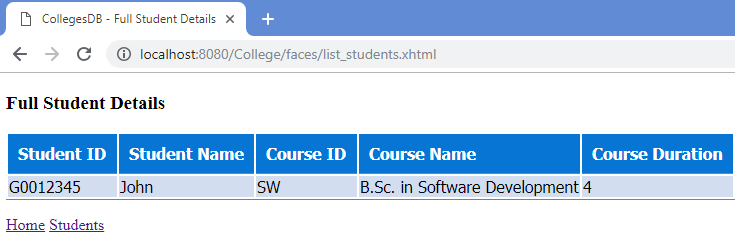


Figure 15 Full Student Details page

#### Delete Student

When the *Delete* action is pressed on the *Manage Students* page two scenarios are possible:

* Scenario 1, the student node in the Neo4j database has no relationships to any other node.

For example, the student *Donnacha*, that student is deleted from the MySQL database and the Neo4j database and the user is returned to the *Manage*

*Students* page where the updated list of students is displayed.

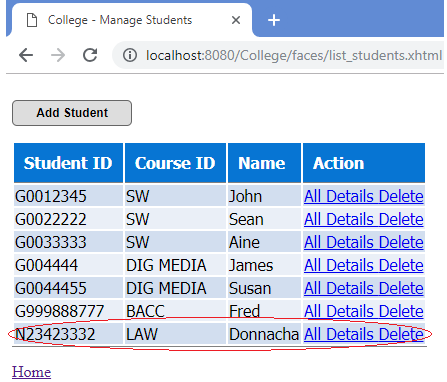


Figure 16 Student deleted from both databases

* Scenario 2, the student node in the Neo4j database has relationships to other node(s).

For example, the student *John*, that student is not deleted from the MySQL database, nor is he deleted from the Neo4j database.

An error message telling the user that this student has relationships is displayed.

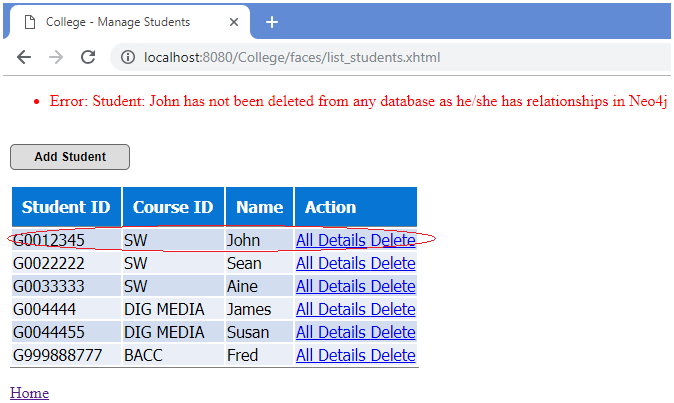


Figure 17 Student not deleted from either database

### Connection Problems

If the MySQL or Neo4j databases are uncontactable at any time, an appropriate error message must be shown:

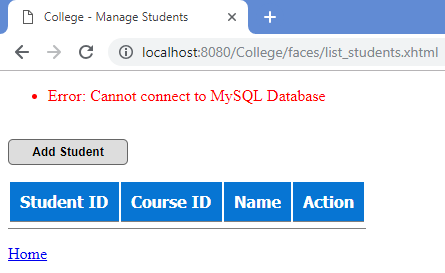


Figure 18 MySQL database uncontactable while trying to Manage Students

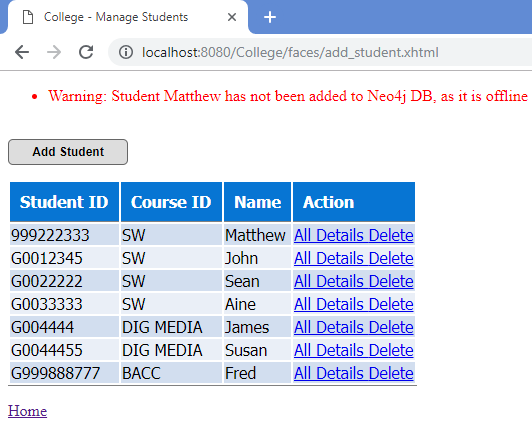


Figure 19 Neo4j database uncontactable while trying to add student