

**Project Description**

**Project Databases**

**Semester 2**

**Year 1**

**5**

Domain Design, Engineering and Computing

Bachelor of Engineering

Mathematical Engineering

Version: 1.0

Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Remarks** | **Status** |
| 1.0 | 13-4-2015 | H. Drillenburg |  | Final |
|  |  |  |  |  |
|  |  |  |  |  |

# Table of contents

[Table of contents 2](#_Toc416689911)

[2 Project Description 3](#_Toc416689912)

[2.1 Introduction 3](#_Toc416689913)

[2.2 Link between project and competencies 3](#_Toc416689914)

[2.3 Goal of the project 3](#_Toc416689915)

[2.4 Educational purposes of the project 3](#_Toc416689916)

[2.5 Attendance requirements 3](#_Toc416689917)

[2.6 Case description 4](#_Toc416689918)

[2.6.1 Situation 4](#_Toc416689919)

[2.6.2 General project approach 4](#_Toc416689920)

[2.6.3 Context and main research question 4](#_Toc416689921)

[2.7 Deliverables 5](#_Toc416689922)

[2.8 Timing 5](#_Toc416689923)

[2.9 Assessment criteria 6](#_Toc416689924)

[2.9.1 Grading 6](#_Toc416689925)

[2.9.1 Product 7](#_Toc416689926)

[2.9.2 Report 7](#_Toc416689927)

[2.9.3 Presentation 8](#_Toc416689928)

[2.9.4 Process 8](#_Toc416689929)

[2.9.5 Git grading 9](#_Toc416689930)

[3 Appendix A: Conceptual Data Model 10](#_Toc416689931)

[3.1 Data model 10](#_Toc416689932)

[3.2 Data Dictionary 11](#_Toc416689933)

[3.3 Attributes 11](#_Toc416689934)

[4 Appendix B: Functionality 12](#_Toc416689935)

[4.1 Functional requirements 12](#_Toc416689936)

[4.2 Technical requirements 12](#_Toc416689937)

# Project Description

## Introduction

The second project of year 1 integrates many of the courses previously (and during this term) taught, and adds the element of a real client to it. It focuses on the use of research techniques to solve a real-world problem. As the courses of terms 3 and 4 have been focusing on database technology, whereas terms 1 and 2 involved web programming, the project combines those two aspects and asks the student to create a web application with a database.

## Goal of the project

The goal of this project is to provide the client with an application which will allow the faculty to offer negotiated study units, students to enroll in them, with the application supporting the desired workflow.

## Competencies and learning objectives

A project has two purposes: developing competencies which cannot be trained in more constrained modules (Manage, Research, Professionalize) and integrating the other competencies in one case study (the five other competencies.) Therefore, the eductional purpose of this project is twofold:

* Integrating the courses from this year into a real-life case, allowing students to experience the cohesion and interdependency between them. This will be achieved by answering the research question by building an application suiting the purpose of the client.
* Learning to apply one specific phase of the research cycle, in this case the phase 'Context and problem analysis.' This will be achieved by specifically having to research the underlying question, interviewing the client once, and figuring out what's already known about the problem.

The competencies are developed in the following learning objectives. After successfully finishing this project, the student can:

* Work in a team to provide a solution to a customer (8.1.5)
* Use methods in applied research to solve a practical problem and prove the solution to be correct (7.x)
* Create and work with a project plan, using a prestructured research and development structure (5.1.2)
* Analyse the needs of the client and translate the results of this analysis to a web-based application (1.1.3)
* Design this web application using prescribed methods and technologies (2.1.3)
* Implement the design including the database, using prescribed methods and techniques (3.1.2)
* Use a version control system to coordinate the work within the team (4.1.1)
* Create and maintain basic functional and technical documentation for the application (4.1.2)
* Present the solution to the client, both written and orally (6.2.3)
* Receive feedback on the results of work and improve by reflecting on this feedback (8.1.2)

## Attendance requirements

During the project, each students' attitude and individual performance is assessed. For this, attendance is necessary. It can of course happen that someone is not able to attend. Therefore it is allowed - though not advisable - to be absent twice during project meetings. Being absent more than twice during the term, ***regardless of the reason***, means failing the project immediately, as in that case grading individual performance is not possible. Being late for a scheduled meeting is unprofessional and will be regarded as being absent.

## Case description

### Situation

The International Business Studies faculty offers a four-years programme in business studies. Part of this programme consists of mandatory courses such as business administration and language skills, but a significant part of the programme can be chosen from a list of options. These for-choice courses are called Negotiated Study Units (NSU's.) The list of NSU's varies; many of them are offered at other faculties, and they may change every year. The faculty wants to be able to offer the courses in a structured way, using a website / web application for this.

In order to enrol for a NSU, a student has to conform to several requirements. He (she) may not have been enrolled for the same NSU previously. Sometimes, a certain course has to be followed before another one can. There may be more requirements. The application should allow students to enrol, but only those courses for which he is eligible should be shown. Enrollment is possible until a certain date - to be set by the faculty -, after which the system closes for the students.

After enrolment closes, the faculty decides which courses will actually proceed. In order for a course to be given, a minimum number of students need to participate. This threshold is set in advance by the offering faculty. A course which does not have the minimum number of participants will not proceed. All students who did enrol for this course should get an email to tell them so; they will receive the opportunity to enrol for another course, so the system should allow this.

There's also a maximum number of participant for a course, also set by the offering faculty in advance. As soon as this maximum has been reached, enrolling in this course should be disabled.

Part of the information analysis for this project has already been performed by a Business IT student. The resulting data model is shown in Appendix A. There may still be a few flaws in it, so check it instead of unthinkingly using it.

### General project approach

Students will work in groups. The project starts with a kick-off in week 1, where the client will also sketch her problem and the imagined solution. The groups will then proceed to get a clear image of the context and problem, design and implement a solution, and write a report to prove their solution is suitable for solving the challenge posed.

An outline of the activities and milestones is given in paragraph 5.7 - Timing.

### Context and main research question

Any project starts with getting context and goal clear. This project is no exception to this rule of thumb. It is strongly advised to spent some time thinking on these and discussing them in the group, afterwards discussing this milestone with the consultant as soon and thoroughly as possible.

## Deliverables

In this project, there are several deliverables:

1. The product itself. The product is a working web application doing what the client needs and agreed upon. It includes a script to create the database and give it the initial data (for example a zipcode table) and software working on this database
2. Necessary documentation for the software. This includes a step-by-step installation guide, a guide for users so they know how to use the software, and a description of how to set up a test.
3. The research report. This includes the description of the context, research question and subquestions, research strategy, research design and data collection, conclusions and any necessary appendices.
4. The way you performed the project is a separate deliverable itself, and one which will be graded by the tutor. It does not have a large impact on the final grade, but if it's a fail grade, the project cannot be improved upon and has to be redone in full next year. Professional use of GIT is part of the process.

## Timing

|  |  |  |
| --- | --- | --- |
| **Week** | **Activity** | **Milestone** |
| 1 | Kick-off  Writing project plan  Planning and distribution of work  Creating interview schema (ask lecturer English for help!) | Project plan - draft  Chapter 1 - Introduction |
| 2 | Interview with client  Defining context  Defining research question  Designing front-end | Project plan - final  Design web pages  Chapter 2 - Context and research question |
| 3 | Setting up GIT  Creating first database version  Start creating code | Chapter 3 - Research Strategy |
| 4 |  | Chapter 4 - Research Design |
| 5 | Creating final database structure |  |
| 6 | Optimizing the database | Chapter 5 - Results |
| 7 | Tying up loose ends  Testing and documenting | Chapter 6 - Conclusions  Chapter 7 - Recommendations |
| 8 | Finishing report |  |
| 9 | Handing in report on Monday 09:00  Giving presentation | Final report  Project files  Presentation |
| 10 | Repairing product and / or presentation (if failed) | Resit product and / or presentation |

We can discuss the chapters one at a time, giving you feedback on the whole draft. Make sure to use this opportunity.

## Assessment criteria

### Grading

After grading the product and analysing the groups' collaboration, a final grade can be given. The final grade will be based on the following factors:

1. Product: Group's grade between 0 and 10
2. Report: Group's grade between 0 and 10
3. Presentation Group's grade between 0 and 10
4. Process, collaboration Group's grade between 0 and 10
5. Individual performance Adjustment to group's grade

The product makes up 50% of the final grade. The report counts for 20% of the grade. The presentation counts for 15%; the way the group worked together counts for 15% of the final grade.

**MPORTANT:**

The first four parts have to be a pass grade in order to pass the project. When failing the product or report, the group has six weeks to hand in an improved version. Failing the presentation can be remedied by redoing it a week after the first try. The Individual Performance can only be redone by redoing the whole project, and might not be possible this year.

Individual performance will be assessed during the whole term. Failing it might mean the student will be removed from the group before the end of the project, thereby failing the project.

Attendance during project meetings is compulsory. Being absent more than twice during the term, ***regardless of the reason***, means failing the project immediately, as in that case grading individual performance is not possible.

Items will be marked according to this table, and the final result will be converted to a grade between 0 and 10.

|  |  |
| --- | --- |
| **Description** | **Grade** |
| Absent | 0 |
| Poor | 1 |
| Insufficient | 2 |
| Average | 3 |
| Good | 4 |
| Excellent | 5 |

Individual performance is not scaled from 1 to 5. Instead, it is a percentage between 0% and 200%, reflecting whether you performed above, on or below groups average. The final grade, assuming a pass for individual performance, is calculated as *groupGrade \* sqrt(percentage/100).* This means that with a high group grade, you still pass with below average performance. However, getting a fail grade *because of* below average performance cannot be repaired, only redone in full.

### Product

The product will be assessed based on the following criteria:

|  |  |  |  |
| --- | --- | --- | --- |
| **Product** | | | |
| **Nr** | **Aspect** | **Grade** | **Comments** |
| 1 | Design of front end web pages |  |  |
| 2 | Logical and physical database design |  |  |
| 3 | Scripts generating database schema and providing initial data |  |  |
| 4 | Code accessing the database |  |  |
| 5 | Code generating the front end web pages |  |  |
| 6 | Database optimizations (indexing & transactions, potentially triggers & stored procedures) |  |  |
| 7 | Functional and technical documentation - including language skill |  |  |
|  | Final grade product |  |  |

### Report

The report will be assessed based on the following criteria:

|  |  |  |  |
| --- | --- | --- | --- |
| **Report** | | | |
| **Nr** | **Aspect** | **Grade** | **Comments** |
| 1 | Analysis and description of the context |  |  |
| 2 | Research question and subquestions |  |  |
| 3 | Research strategy & design |  |  |
| 4 | Conclusions and recommendations |  |  |
| 5 | Report structure - logical connection between chapters and general structure |  | If this item is less than 3, report fails. |
| 6 | Language skill |  |  |
|  | Final grade product |  |  |

### Presentation

The presentation will be assessed based on the following criteria:

|  |  |  |  |
| --- | --- | --- | --- |
| **Presentation** | | | |
| **Nr** | **Aspect** | **Grade** | **Comments** |
| 1 | Structure (Introduction-core-conclusions) |  |  |
| 2 | Information level |  |  |
| 3 | Attitude, fluency |  |  |
| 4 | Reflection and feedback |  |  |
| 5 | Quality of aids used |  |  |
|  | Final grade presentation |  |  |

### Process

The process will be assessed based on the following criteria:

|  |  |  |  |
| --- | --- | --- | --- |
| **Process** | | | |
| **Nr** | **Aspect** | **Grade** | **Comments** |
| 1 | Structure and content of the final project plan - version available in week 7 |  |  |
| 2 | Programmes and minutes of meetings, punctuality, completeness |  |  |
| 3 | Project meetings with tutor and / or consultant, using and giving feedback |  |  |
| 4 | Project file, timeliness, completeness and contents |  |  |
| 5 | Written reflection on how the project was conducted |  |  |
| 7 | Use of GIT during the project |  |  |
| 6 | Language and design of documents distributed during the project |  |  |
|  | Eindcijfer procesgang |  |  |

### Git grading

In using GIT, the following criteria will decide the partial grade:

**Grade: 1 (out of 5)**

The gitlog is unclear

The team doesnt use GIT (roles, gitignore)

Only 1 teammember is active

Rare commits on the master branche

The team avoids conflicts (large portions of the project committed)

**Grade: 3 (out of 5)**

The gitlog messages are all filled out and clear

The team uses a role based centralised workflow

Some teammembers are active (development commits)

The team uses the master and feature branches

The team demonstrates conflict resolution in merging/rebasing

**Grade: 5 (out of 5)**

The gitlog documents documents a complete and readable history

The team uses a role based distributed workflow

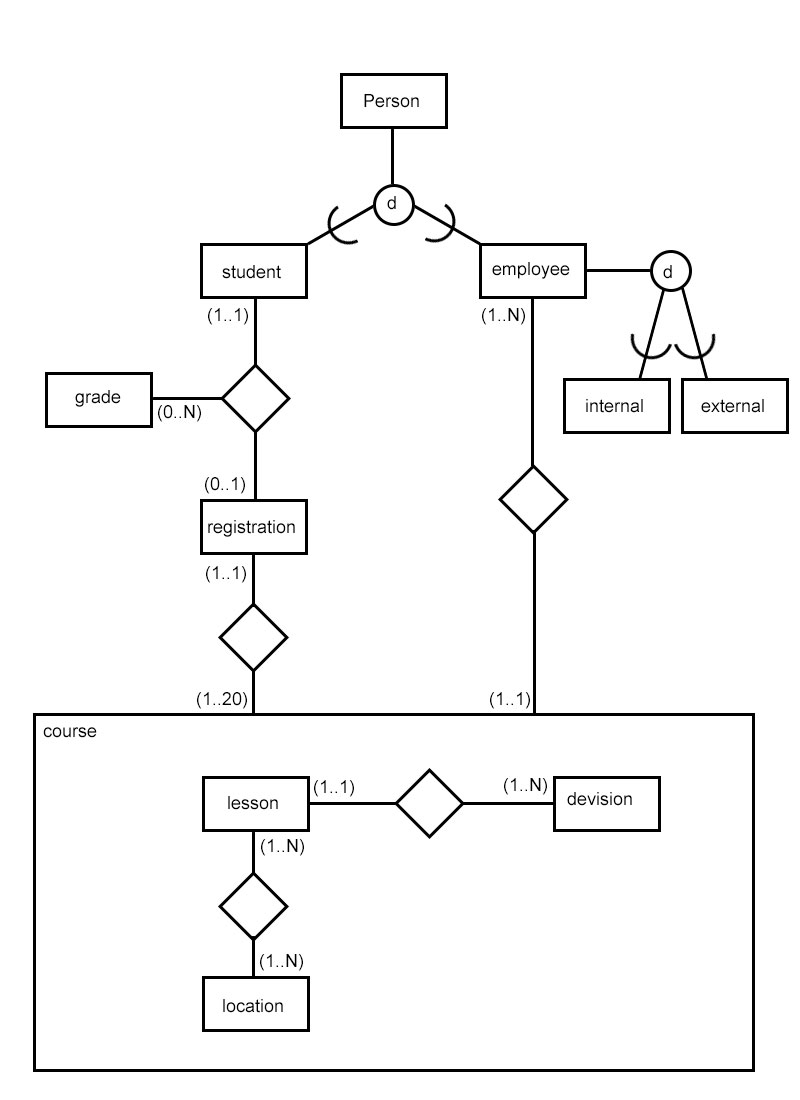
All teammembers are active (development/documentation commits)

The team uses feature branches

The team demonstrates role base pull request in merging/rebasing ​

# Appendix A: Conceptual Data Model

## Data model



## Data Dictionary

**Person:** *Individual, a human being with rights and duties. Difference is made between students and employees.*

**Grade:** *Achieved results for the courses concerned, if applicable.*

**Registration:** *Application for a specific course, considering the valid requirements.*

**Lesson:** *Teaching knowledge, skills and attitudes, considering preselected goals.*

**Study:** *Programme of 240 EC with individual accreditation.*

**Location:** *Physical or logical place to locate the course.*

**Division:** *Education (group)*

**Course:** *Collection of subjects, amount of EC, contains lessons under the responsibility of an education (in case of obligation).*

## Attributes

**Student**

* First name
* Surname
* Phone number
* Student number
* Email

**Employee**

* First name
* Surname
* Address
* Residence
* Postal code
* Phone number
* Email
* Type

**Grade**

* Year
* Semester
* Grade

**Course**

* Name
* Cost

**Lesson**

* Days
* Start
* Duration
* Capacity

**Division**

* Description
* Name

**Registration**

* Time

**Location**

* Place
* Range

# Appendix B: Functionality

While the information analysis focused on the data to be stored, some information about functional and technical requirements were already discovered. These findings are mentioned here.

## Functional requirements

The client requests a system which makes it possible to offer courses by an online registration system. The courses can be manufactured and removed by a few specific employees of the school, while they are administrator. The available courses can be taken only once, but it is possible for the administrator to modify the system so the student can take a course another time, in case it failed the first time. Every course has a limit of students; this limit is determined by the administrator. The course contains a number of lessons which take place at variable times and locations. Every course is an optional subject in a certain field of study, which can be lectured by internal or external employees.

## Technical requirements

While it is not allowed to connect to the system and the network of Inholland, employees need to register the students manually in the system.

An export function must be available, to provide the administrator with information of the courses and students.

After registration students must receive a confirmation by email.

Students log in with their Inholland email address.

The system can be opened and closed, based on a timer function.

A taken course stays available for the administrator, which can change it and publish it again.

**Password policy**

No username

At least one uppercase and one special character

At least six characters.