Datenbanken und Web-Techniken
Project Task
Summer Semester 2021

1. Introduction

A school wants to switch its grading system from paper to digital in order to have a better overview of the test results and the performance of all pupils. In addition, it should also offer an improved communication between teachers and pupils (and their parents). The new system should provide a reasonable user rights management and different views for each user role.

Your task is to implement a first prototype as a web application. The focus will be to provide a simple and fast method for teachers to record the results of tests of all their pupils and also to get an overview of tests and subjects for both pupils and teachers with some basic export functionality. Additionally, a basic messaging system for communication should be implemented, too. The detailed task description is given on the following pages.

2. Task Description
2.1 Preliminary remarks

The following practical task may be processed in groups of up to two participants. If more than one participant is present, it should be indicated for all parts (programming, term paper), who was responsible for which part. For groups, the additional tasks mentioned at the end are obligatory (individuals may of course solve them, too).

The subtasks may be solved by the usage of any database management system, any programming language, any frameworks and any web service API techniques, i.e. there are no restrictions in choice.

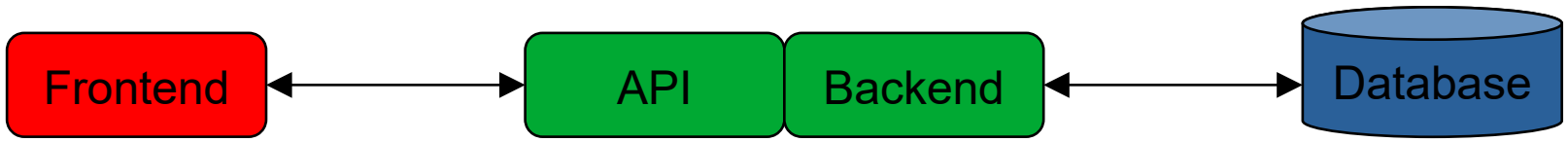
The submission consists of:

1. a PDF-file of the term paper in paper format A4 and
2. a separate ZIP-archive (with a maximum file size of 10 MiB) that contains the sources and a small manual on how to use the sources to get a working web application.

2. Task Description
2.2 Programming

The program consists of

1. a database for storing the data,
2. a backend for processing the data,
3. a web service application programming interface for providing the data and
4. a frontend for displaying the data.



2. Task Description
2.2.1 Database

1. All data is stored in the database, i.e. no additional storage (like files or web storage) is used.
2. No plaintext passwords are stored in the database.
3. The frontend must never communicate directly with the database.

2. Task Description
2.2.2 Backend

1. The backend processes the data.
2. It handles all communication with the database.
3. It provides the web service API to be access by the frontend.

2. Task Description
2.2.3 Web Service Application Programming Interface

1. The web service application programming interface provides communication between the frontend and the backend as well as the database.
2. It delivers the data from the database to the frontend.
3. It provides the user input from the frontend for processing in the backend.
4. API access failures are caught and responded with a meaningful error message (and possibly a corresponding HTTP status code).

2. Task Description
2.2.4 Frontend

1. The frontend is a web application to be accessed by a web browser.
    a) All common web browser are supported, i.e. Mozilla Firefox, Apple Safari and Chromium derivates.
2. Usability aspects are taken into account.
3. The design is appealing.
4. Users start with logging in to the application with their username and password.
    a) Neither username nor password may be stored or cached at the frontend side (i.e. in the browser).
5. Users belong to one of three different mutual exclusive roles: admin, teacher and pupil.
    a) After login they continue with the view corresponding to their role.

2. Task Description
2.2.4 Frontend

6. The admin view provides the management of users including the options to:
   a) list all available users,
   b) add a new user,
      i. A user role has to be selected upon user creation, which cannot be changed later.
      ii. Each user must be assigned at least a username, a password, a forename and a surname.
      iii. Each user is identified by a system provided ID, which cannot be changed.
   c) change the attributes of a user and
   d) remove a user.
      i. Teachers cannot be removed while they are assigned to at least one subject, that is not archived,
      ii. Dependent information like test results of pupils get removed along with them.
7. The teacher view and the pupil view provide the change of their own attributes.

2. Task Description
2.2.4 Frontend

8. The admin view provides the management of classes including the options to:
   a) list all available classes,
   b) add a new class identified by a unique name,
   c) change the attributes of a class including the options to:
      i. change the class name,
      ii. manage subjects (see next page for detailed description),
      iii. assign pupils and
         - Pupils cannot be assigned to more than one class, i.e. assigning them to a new class automatically deassigns them from the previous class.
      iv. deassign pupils.
         - Pupils keep their achieved test results by the usage of archived subjects (see next page), i.e. archived copies of non-archived subjects get created just for them to store their corresponding test results.
   d) remove a class.
      i. Pupils get deassigned and dependent subjects get removed or archived (see next page).

2. Task Description
2.2.4 Frontend

9. The admin view provides the management of subjects as part of the attribute change in class management including the options to:
   a) list all available subjects,
   b) add a new subject,
      i. Each subject is identified by a class wide unique name and has exactly one teacher, i.e. an already existing teacher has to be assigned to a new subject upon creation.
   c) change the attributes of a subject,
   d) archive a subject and
      i. Only subjects with dependent tests can be archived.
      ii. No further changes can be made anymore to archived subject (including dependent information like tests).
      iii. Archived subject keep existing, even if the corresponding class gets removed.
      iv. Archiving cannot be made undone.
   e) remove a subject.
      i. Only subjects without dependent tests can be removed.

2. Task Description
2.2.4 Frontend

10. The teacher view provides the selection of one of the assigned subjects and afterwards the management of tests (whereat manipulations are not allowed for archived subjects) including the options to:
    a) list all pupils along with their average grades (i.e. the arithmetic mean over the grades of all tests),
    b) add a new test,
        i. Each test must be assigned at least a name and a date.
    c) change the attributes of a test including the options to:
        i. change the test name and date
        ii. list all pupils along with their grades in the test
        iii. add or change the grade of one pupil in the test
        iv. import the grades of all pupils in the test, e.g. upload a CSV or ODS file for batch grading
    d) remove a test.
        i. Already provided test results of pupils get removed along with the test.

2. Task Description
2.2.4 Frontend

11. The pupil view provides an overview about subjects and tests including the options to:
    a) list all assigned subjects along with their average test grade (i.e. the arithmetic mean over the grades of all tests of a subject) and
    b) select one assigned subject and list all corresponding tests along with their grades.
12. For demonstration purposes there are available at least:
    a) 1 admin
    b) 2 classes, whereof each has at least:
        i. 3 subjects, whereof at least 1 has 0 tests and another 1 has 3 tests
        ii. 4 pupils
    c) 1 pupil, which is not assigned to any class
    d) 3 teachers, whereof at least:
        i. 1 is assigned to 0 subjects
        ii. 1 is assigned to subjects of at least 2 different classes and at least 2 different subjects of 1 class

2. Task Description
2.2.5 Additional tasks for groups

1. The views of all roles provide a messaging system including the options to:
   a) write a message to a certain user or all members of an active (i.e. non-archived) subject,
      i. Pupils are restricted to (other members of) their own active subjects.
      ii. Teachers may write to any user except for pupils, which are not members of their own active subjects.
      iii. Users cannot write messages to themselves.
   b) get notifications about new messages and
   c) read messages.
      i. Messages are removed by the system after seven days (even unread messages).
2. The views of all roles provide export facilities for each required list.
   a) The export file format has to be different from the presentation format, i.e. just providing an HTML file is not sufficient, but CSV, ODS or PDF would be.
3. The web service API has a security layer for authentication, e.g. JSON Web Token (JWT).

2. Task Description
2.3 Term Paper

A term paper is to be written, that satisfies the following conditions:

1. There is an amount of about 10 pages of content per project participant (i.e. 20 pages for groups).
2. On the cover page there are the name, the study course and the matriculation number of all participating students recorded.
3. An overview of all utilized technologies (i.e. DBMS, programming and other languages, frameworks, web service API implementation, ...) is given along with a short motivation why they were chosen to solve a certain subtask. They are also classified in the context of the lecture.
4. The project is presented in such a way that, after reading, you know all parts of the program and their functionalities without having explicitly executed them.

2. Task Description
2.3 Term Paper

5. A good form and a balanced ratio of pictures and text is kept.
6. All used sources, libraries and technologies are referenced.
7. The appendix (i.e. not part of the required pages) includes a complete web service API documentation (i.e. just listing examples is not sufficient) containing:
   a) a list of all endpoints of the web service application programming interface,
   b) for every endpoint the list of the parameters and the return values each with type and meaning and
   c) for complex structures also the inner structure is to be documented respectively.

3. Examination

The examination consists of a 15-minute presentation, which should meet the same criteria as the content of the term paper, but the API documentation is omitted here.

So the project and the used technologies should be presented and put into the lecture context. The presentation should also include a live demonstration of the project or a demonstration video demonstrating all parts of the practical task. For group work, the presentation time should be divided equally between both students.

Afterwards some questions (primary concerning the project and the term paper) are given. Finally there will be a short consultation and you will be informed about your mark.

4. Dates

- Handout of task description:
  - starting 2021-05-31 12:00 CEST (UTC+2)
- Submission of project:
  - until 2021-07-05 12:00 CEST (UTC+2)
    - late submissions will be accepted, but your final mark will be reduced by 1/3 per 8 hours or parts thereof
      - your mark will be reduced by 1 per day, so you cannot pass, if you submit after 2021-07-08 12:00 CEST
  - via OPAL:
    ➔https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/297435137/CourseNode/161715812006310 3003
- Oral exam and presentation:
  - between 2021-07-12 and 2021-07-23
  - appointment allocation will be done via OPAL, too
  - examination will take place via video conference as default
    - we will try to offer personal meetings in a meeting room in the university as alternative if required