Advanced Management of Data
Project Task
Summer Semester 2021

1. Introduction

The student representatives of faculty computer science (Fachschaftsrat Informatik – FSR:IF) offered a recurring WorkTogether event in pre-corona-times. This WorkTogether used to be an organized meeting of study groups which took place in the common room at Straße der Nationen several times each semester. Generally, students were able to come there and meet other students with similar problems to be solved together. As there will hopefully be a time after Corona, the FSR:IF intents to continue the organisation of such WorkTogethers.

In the past, a common challenge has been to find a suitable study group. One could have come there with a problem, but it was uncertain if somebody was there, willing to work on the same problem. Therefore the FSR:IF likes to offer a study group management tool for its WorkTogether meetings. Your task will be to implement a prototype of such a tool according to the requirements 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.

The submission consists of:

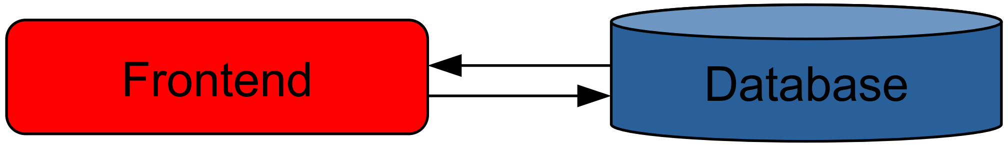
1. a PDF-file of the term paper in paper format A4 and
2. a separate ZIP-archive (with a maximum size of 10 MiB) that contains
   - the sources,
   - a script to initialize the database and
   - a small manual on how to use the sources to get a working program.

2. Task Description
2.2 Programming

The program consists of

1. a PostgreSQL-database with the program logic implemented in PL/pgSQL and
2. a frontend for interaction with the data from the database. There are no restrictions concerning the programming language of the frontend, but if you like to help the FSR:IF with a prototype that is easy to integrate into existing infrastructure, you might use the TU Chemnitz Autoren- und Layoutsystem (TUCAL) offered by the URZ (which is based on PHP):

   ➔https://www.tu-chemnitz.de/urz/www/tucal.html



2. Task Description
2.2.1 Database

1. The database is provided by PostgreSQL.
2. Most of the program logic is implemented directly in the database using PL/pgSQL as main programming language.
3. All study group data (and related information) is stored in the database.

2. Task Description
2.2.2 Frontend

1. There are two different mutual exclusive views: FSR:IF and student
   a) it can be switched easily between both and there is no need for complex user management
      i. students are identified by a user providable name, which can be changed easily for demonstration purpose
2. The FSR:IF view manages WorkTogether meetings including the options to:
   a) list all available meetings to get an overview
      i. including the option to list all corresponding study groups with all information (including joined students)
   b) add a new meeting
      i. each meeting takes place at a location and has a certain date and time information for starting and ending
      ii. starting time of a meeting must be before ending time and both cannot be set into the past
   c) change attributes of a meeting including the option to change visibility of a meeting
      i. default state of a new meeting is hidden (so it has to be changed to visible in a second step)
      ii. if ending time is reached, meeting state has to be switched to hidden by the system automatically
   d) remove a meeting along with all related information (especially all corresponding study groups)
      i. only hidden meetings may be removed

2. Task Description
2.2.2 Frontend

3. The student view starts with selection of one of all available WorkTogether meetings
    - a) hidden WorkTogether meetings must not be shown
4. The student view continues with study group management including the options to:
    - a) create new study group
        - i. each group has a topic, a description and a student count limit (minimum limit of 2, maximum unlimited)
        - ii. creator gets assigned as group owner and is joined to group automatically
    - b) change attributes of own study group (only available if student is group owner)
        - i. changing student count limit below current count doesn’t lead to forced leaves
    - c) list all joinable study groups (groups with student count limit reached must not be shown)
    - d) join study group
        - i. students must not join more than one group at once, so joining automatically leads to leaving previous group
    - e) leave currently joined study group (only available if joined)
        - i. upon group owner leaving, ownership gets assigned to next student in group, that joined after owner
        - ii. as soon as last student leaves a study group, the group gets deleted

2. Task Description
2.2.2 Frontend

5.Usability aspects are taken into account.

6.For demonstration purposes there are available at least:

- a)3 different WorkTogether meetings, whereof
  - i. 1 is in the past
  - ii. 2 are in the future
- b)3 different study groups for each of these 3 WorkTogether meeting, whereof
  - i. 1 is of unlimited size
  - ii. 1 is joined by students until the limit
  - iii.1 is joined by a number of students one before the limit (so the next student joining will close the group)

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 12 pages of content.
2. On the cover page there is the name, study course and matriculation number of all students recorded.
3. An overview of all utilized technologies (excluding PostgreSQL and PL/pgSQL) is given along with a short motivation why they were chosen to solve a certain subtask.
4. The database tables are visualized using both Unified Modelling Language and Relational Schema.
5. 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.
6. It is discussed under which conditions a distributed database could be used with the project and how this distribution can be achieved in a good way. In case there are multiple options, no more than 3 different scenarios are described.
7. A good form and a balanced ratio of pictures and text is kept.
8. All used sources, libraries and technologies are referenced.

3. Examination

The examination consists of a 10-minute presentation, which should meet the same criteria as the content of the term paper, but the focus is on technologies and program functionality.

So the project and the used technologies should be presented, while UML, the Relational Schema and the distributed database discussion can be kept short (no more than 1 scenario here). 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 2021-04-26 12:00 CEST (UTC+2)
- Submission of project:
  - until 2021-05-31 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 2021-06-03 12:00 CEST
  - via OPAL:

    ➔https://bildungsportal.sachsen.de/opal/auth/RepositoryEntry/20312915968/CourseNode/1617158120350182003
- Oral exam and presentation:
  - starting 2021-06-07
    - until we are done (probably within one week, but depending on our time and the number of groups)
  - 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