TECHNISCHE UNIVERSITÄT
CHEMNITZ

Advanced Management of Data

Project Term Paper

Professorship of Data Management Systems
Dept. of Computer Science

Supervising tutor:
Daniel Richter
Florian Hahn

Submitted by:
1.
Name : Jannatul Fardouse
Student ID: 621982
Study Course: Automotive Software Engineering
2.
Name : Jannatul Ferdous
Student ID: 562036
Study Course: Automotive Software Engineering
Date of submission: 31.05.2021

Abstract

This project term paper of **Advanced Management of Data** has been entitled to **"WorkTogether"**. A customized online study group and meeting management system has been developed to fulfil this course criteria.

There are two different mutual exclusive views: **FSR:IF and student**. The FSR:IF manages WorkTogether meetings as an administrative member. The students manage the study groups efficiently to discuss all the study related problems.

FSR:IF lets students to discuss and solve their study related problems by creating meetings and thus students can join or create study group to attend the corresponding meeting. FSR:IF follows up the study groups, meetings and student profile. On the other hand, students can be in only one study group at a time. If they try to join another study group, they will be removed automatically from the previous study group.

Keywords: PostgreSQL, Database, HTML, CSS, JS, PHP

Acknowledgement

This Project titled ""**WorkTogether**"" has been prepared to fulfill the requirement of completion of Computer Science section in Masters of Science in Automotive Software Engineering, Department of Computer Science, Chair of Computer Engineering.

We would like to express our heartiest gratitude to our supervising tutors, **Daniel Richter** and **Florian Hahn**, Faculty of Computer Science, **Chemnitz Universtiy of Technology**, for their continuous guidance and patience, and for giving us the opportunity to do this work.

Finally, our deepest gratitude and love to our parents for their support, encouragement,

Contents

List of Figures

List of Tables

1 Introduction

The **"WorkTogether"** is an online study group management system that enables the students to find suitable study group related to their courses and problems and solve them. The goal of this paper is to explain about creating a "user-friendly" as well as an effective study group management system between **FSR: IF** and **Student**.

FSR: IF is the student representative council of **Computer Science** at the **TU Chemnitz**, where the students are always getting suggestion for the study and examination regulations. As FSR: IF is the part of university committee, it also represents all the student of the Faculty of Computer Science. FSR: IF organizes the networking meetings with students of other faculties, "**I.e.**, **the Gaming & Cultural events**, **Grill Party evenings**, **Christmas Parties** or **Sports** in university. In addition, FSR: IF is the most important platform to make a study group and get the various opportunity for the study purpose.

2 Project: WorkTogether

This chapter describes the **Requirements** and **Implementation** of **"WorkTogether"** project. The main goal is to implement a prototype of such study group management system according to the requirements stated in our project description given by our honorable faculties.

2.1 Requirement

The top-level requirement for the project is to program the system using the PostgreSQL Database[1] where the logic of program are implemented in PL/SQL and a front-end for interaction with the data from the database[2]. For the simplicity of the communication with the database and the frontend, we have used the an interface. The data from the database is received by the frontend through the interface. The following diagram describes the scenario:

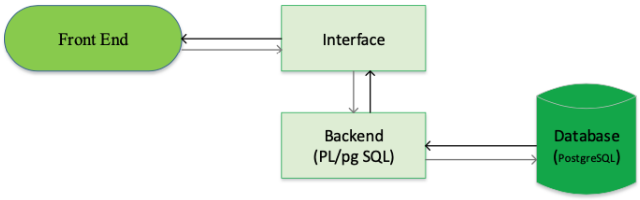

Figure 2.1: Project structure from top-view

2.1.1 Requirement Analysis

The system consists of two different mutual exclusive views : **FSR:IF** and **Student**[2]. The following table shows the requirements with the ToDos of each views.

APPLICATION STACK	TODOs
FSR:IF	UI to create a meeting with location, date & time information. UI to update a meeting along with visibility changing option. UI to overview of all available meetings. UI to delete a meeting along with all related information. UI to delete student from any study group.
Student	UI to register them-self as student & log in. UI to create a study group with topic, description and a student count limit. UI to update own study group. UI to overview of all join-able study groups. UI to join study group. UI to leave currently joined study group.
Database PostgreSQL	Stores all data. Stores all computational logic in PL/pgSQL
Frontend	Validate the data against the database. Shows all the functionalities.

Table 2.1: Requirement analysis

2.2 Implementation

According to the requirements we have designed and implemented Database, Frontend and Backend to connect Database and Frontend.

2.2.1 Database Design

For developing this system database tools and technologies have been needed. We have used PostgreSQL which is open source relational database management system (RDBMS). All data has been saved on this system. Thus our main program logic is implemented in PL/pgSQL. The procedural language PL/pgSQL adds many procedural elements, e.g. **control structures**, **loops** and **complex calculations** to extend standard SQL.

2.2.1.1 Database Unified Modeling Language (UML)

To visualize the system general-purpose modeling language is used and thus it is known as **Unified Modeling Language (UML)**[3]. The conceptual schema of the database supporting the implementation is depicted in the UML diagram below. It displays all of the relevant tables and their connections in accordance with the project specifications.

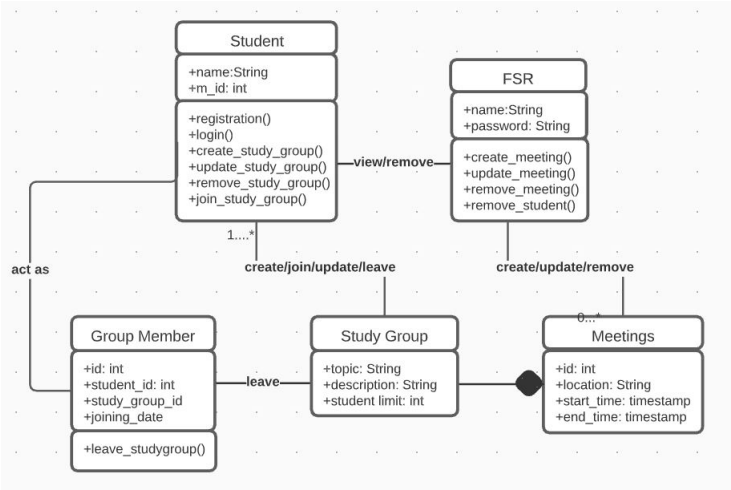

Figure 2.2: UML Diagram of Database

In general, there the 5 tables, 14 functions and one trigger have been created to serve the application requirements. The relations among different tables are indicated using different relations.

2.2.1.2 Relational Schema

The **Relation Schema** provides the relation's design and structure, including the relation name and a set of attributes/field names/column names[4]. Relational or

Database schema is a collection of meta-data. In the figure below, we have shown the attributes each table contains in our system.

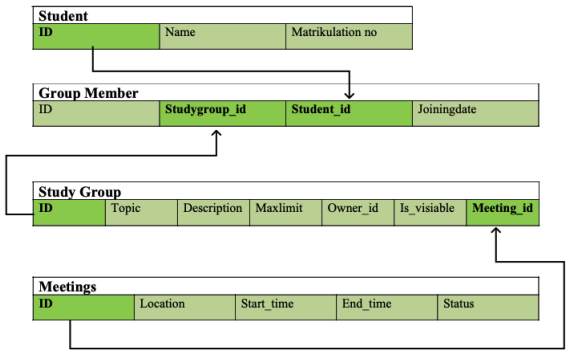

Figure 2.3: Relational Schema

2.2.1.3 Use Case Diagram

A **Use Case Diagram's** main objective is to represent a system's dynamic aspect. It depicts the system's exterior view. It takes into account both internal and external influences on the system[5].

FSR:IF Use Case Diagram:

In our project's perspective, FSR:IF can add meeting. He has the right to remove any meeting at any moment with all the related information as well as remove any student from the system. FSR:IF can update any information of a meeting such as location, date and time. He can also view the list of the study groups having all student profiles.

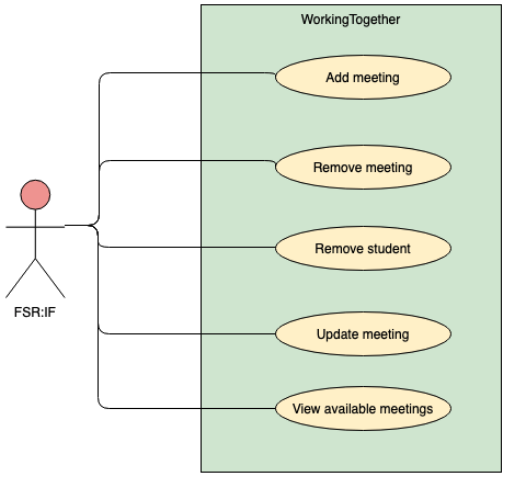

Figure 2.4: FSR:IF Use Case Diagram

Student Use Case Diagram:

From the student perspective, a student can register himself as a student and login to proceed. He/she is allowed to create study group and update any information of his/her own study group. He can join only in one study group at a time. He can leave any study group.

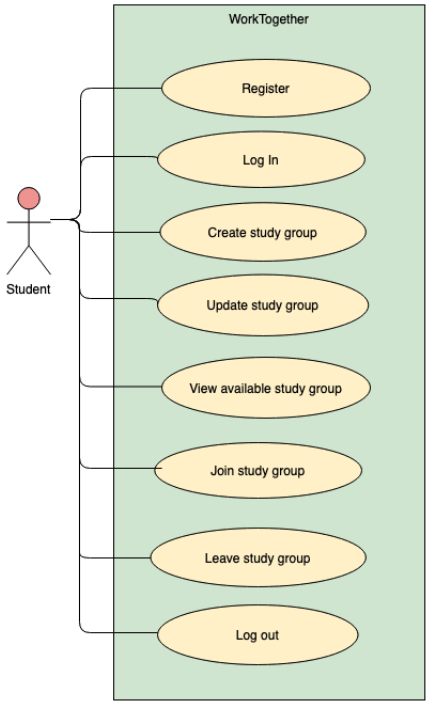

Figure 2.5: Student Use Case Diagram

2.2.2 Frontend

2.2.2.1 HyperText Markup Language (HTML)

HTML is the essential markup language for archives that contain pages and other information that can be seen in any internet browsers. HTML is utilized for making the visual web utilization of the web pages. It contains numerous tags (such as **<!DOCTYPE>, <html>, <head>** etc). In the pages, we can undoubtedly embed the pictures and components and can be utilized to make instinctive designs by utilizing HTML. As a matter of fact, HTML has created determined to characterize the construction of archives, e.g., headings, passages, records and so forth to work with the trading of logical data between specialists. It can embed content written in lingos, for instance, JavaScript and Cascading Style Sheets, which impact the conduct of HTML pages. Presently a day, HTML is broadly used to organize site pages utilizing different labels accessible in the HTML language. [6][7]

2.2.2.2 Cascading Style Sheets (CSS)

CSS is a format language for addressing the semantics of the introduction of an archive in a markup language like **HTML** or **eXtensible Markup Language (XML)**. CSS is the most well-known application is styling website pages written in HTML. It addresses how components ought to be delivered on screen or on other media. CSS is intended to empower the detachment of show substance like format, content, shading, structure, text styles, and so on Its simpler to stay aware of more consistency in construction and update, and to offer considerably more plan options. Effectively acquaint various styles with various watchers.[9][10]

2.2.2.3 JavaScript (JS)

JavaScript(commonly abbreviated as JS) is a lightweight, interpreted, object-oriented programming language with first-class functions. It's best known as a scripting language for Web pages, although it's also used in a variety of non-browser applications. It's a multi-paradigm, prototype-based scripting language that supports object-oriented, imperative, and functional programming techniques. JavaScript is a client-side scripting language that may be used to create and program how web pages react when an event occurs. JavaScript is a scripting language that is both simple to learn and powerful, and it is commonly used to control the behavior of online pages.[12]

2.2.2.4 Bootstrap

Bootstrap is a free and open-source CSS system coordinated at responsive, portable first front-end web advancement. It contains CSS-and (alternatively) JavaScript-based plan layouts for typography, structures, catches, route, and other interface segments. T he most unmistakable segments of Bootstrap are its design parts, as they influence a whole site page. The fundamental design part is classified **"Compartment"**, as each and every component in the page is set in it. Engineers can pick between a fixed-width compartment and a liquid width holder. While the last consistently fills the width of the site page, the previous uses one of the five predefined fixed widths, contingent upon the size of the screen showing the page: More modest than 576 pixels , 576–768 pixels , 768–992 pixels , 992–1200 pixels , Bigger than 1200 pixels when a compartment is set up, other Bootstrap format segments execute a CSS Flexbox design through characterizing lines and segments.[8]

2.2.3 Backend

2.2.3.1 Hypertext Preprocessor (PHP)

PHP began as a touch open source project that evolved as more and more people acknowledged how useful it had been . Rasmus Lerdorf unleashed the first version of PHP way back in 1994. PHP could also be a recursive acronym for "PHP:

Hypertext Preprocessor(earlier called, **Personal Home Page**)". PHP could also be a server side scripting language that's embedded in HTML. it's wont to manage dynamic content, databases, session tracking, even build entire e-commerce sites. It is integrated with sort of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.[11]

2.2.4 Application Architecure

The structure of the application has been divided into three main logical components. The first component- "assets" provides the necessary functionalities to create the views for the frontend. The "*.php" contains all the logical functionalities and commands which connect the backend with frontend. Additionally, the "classes" contains the necessary SQL scripts to initialize the database. These scripts initializes the tables and the PL/pgSQL procedures and triggers necessary for the application.

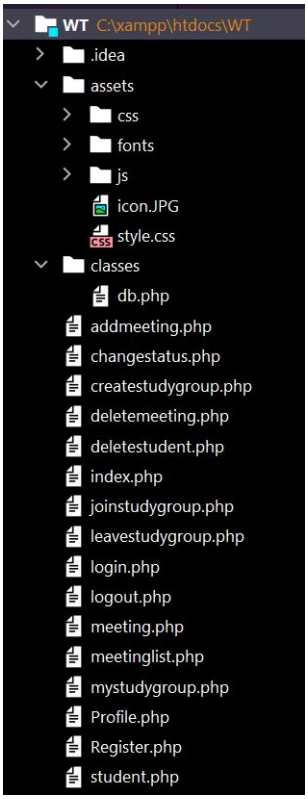

Figure 2.6: Application Architecture

3 Functional Factors

This chapter illustrates some of the functions that have been developed and the description of some basic logic that is applied to the implementation of the application.

3.1 Explanation

3.1.1 Create function

- The FSR:IF can add new meeting providing location, start and end time of the meeting.
- A student can create a new study group with topic, description, student count limit.

3.1.2 Remove function

- The FSR:IF can remove a meeting with all related information only if the meeting is hidden.
- The FSR:IF can remove students from any study group.

3.1.3 Update function

- The FSR:IF can update any information of a meeting such as location, start of end time of the meeting as well as the visibility.
- A Student can update any information of a study group he/she owns such as topic, description, student count limit.

3.1.4 Visibility function

- The FSR:IF can update the visibility of a meeting as the default state of visibility is hidden. So FSR:IF can enable showing the meeting.

3.2 Analysis

This leave_group function has been developed to carry out the factor of leaving of the owner of a study group. When an owner of a study group leaves the group, the second member of that group automatically gets the privilege to be the owner of this group. In case, if the group has only one member left and the member leaves the group, then the group does not exist anymore. Thus, the **Create, Update, Remove** and **Visible** functions have been utilized in one scenario.

```
CREATE FUNCTION public.leave_group(l_studygroup_id integer, l_student_id integer) RETURNS character varying
    LANGUAGE plpgsql
    AS $$
        Declare

    ownerid int;
    new_owner_id int;

    BEGIN

    DELETE FROM groupmember WHERE studygroup_id = l_studygroup_id and student_id = l_student_id;

    UPDATE studygroup set is_visiable = TRUE WHERE studygroup.id = l_studygroup_id;

    ownerid := (select studygroup.owner_id from studygroup where studygroup.id = l_studygroup_id);

    new_owner_id := (select groupmember.student_id from groupmember where groupmember.studygroup_id = l_studygroup_id order by joiningdate asc limit 1);


    if new_owner_id is NULL THEN

        DELETE FROM studygroup WHERE studygroup.id = l_studygroup_id;

    elsif ownerid =l_student_id THEN

        UPDATE studygroup set owner_id = new_owner_id WHERE id = l_studygroup_id;
            UPDATE studygroup set is_visiable = true WHERE id = l_studygroup_id;

    END IF;

    return 'Successfully Removed';

END;
```

Figure 3.1: Leave Group Function

In below figure, we have provided the PHP script of update study group function where any kinds of update reflects in the database.

```
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $topic = $_POST['topic'];
    $description = $_POST['description'];
    $maxlimit = $_POST['maxlimit'];
    $owner_id = $_SESSION['student'];
    $status = True;

    $sql = pg_query($db, "select update_studygroup('$id','$topic','$description','$maxlimit', '$owner_id', '$status');");
    while ($row = pg_fetch_row($sql)) {

        if ($row[0] == 'Error') {
            echo "Error";
        }
        else{
            header("Location: studygrouplist.php");
        }
    }
```

Figure 3.2: PHP Script of Update Study Group

4 Distributed Databases

Basically, using a **Distributed Database Management System(DDBMS)**, anyone can manage a distributed database as if it were all stored in a single location. It allows the creation, retrieval, updating, and deletion of distributed databases. It synchronizes the databases periodically and provides access mechanisms by which the distribution becomes transparent to the users[13].

In our project, we implement a gradual system which can use at any perspective for any institutions..

- Distributed Organizational Units: With the increase in globalization, most organizations in the current times are subdivided into multiple units that are physically distributed. In this case, if our **WorkTogether** project needs to subdivide, we can implement the distributed database so that the overall database of WorkTogether can be distributed.
- Data Sharing: In future, if our WorkTogether project needs to communicate with multiple organizational units and share the data and resources, we may need to use distributed database. This will demand common databases or replicated databases that should be used in a synchronized manner.
- Database Recovery: One of the common techniques used in distributed database is replication of data across different sites. Replication of data automatically helps in data recovery if database in any site is damaged. This can be used in case we want to implement backup and recovery in WorkTogether project.

5 Conclusion

The **WorkTogether** project has been developed successfully with all the requirements that was stated in our project proposal. Technologically, we have used **HTML, CSS, JavaScript, Bootstrap, PHP** and **PostgreSQL**. Besides that, we have used **Notepad++** to write our code, **pgAdmin** to work with **PostgreSQL** and **Xampp** to run the project in server.

Finally, we have came across about **Distributed Database Management System(DDBMS)** which can be used globally for the further development of this project by any institution.

Bibliography

[1] *Group P.G.D: PostgreSQL*, https://www.postgresql.org/

[2] *Richter, D.: Advanced Management of Data Project Task Summer Semester2021,* https://www.tu-chemnitz.de/informatik/DVS/lehre/AMD/Project/AMD%20-%20Project2.pdf

[3] *GeeksforGeeks, UML 17th February, 2021*, https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/

[4] *GeeksforGeeks, Relation Schema in DBMS 12th June, 2020*, https://www.geeksforgeeks.org/relation-schema-in-dbms/#:~:text=Relation%20schema%20defines%20the%20design,would%20have%20an%20associated%20domain.

[5] *GeeksforGeeks, Use Case Diagram, 16th June, 2020*, https://www.geeksforgeeks.org/use-case-diagram-for-library-management-system/

[6] *Spiros Mancoridis, Introduction to the HTML*, https://www.cs.drexel.edu/s̃piros/teaching/SE101/slides/html.pdf?fbclid=IwAR35Z4_Wcyq-xTk87j0D8M7jVcrsVXi5_dTjuB4VL80uOpk1h-Rt20aOi0w

[7] *Tutorialspoint, HTML - Overview*, https://www.tutorialspoint.com/html/html_overview.htm?fbclid=IwAR1yIeeb2UGZopebMHKe_6QDURKJ1G7e6sJdaZfNJrBByi3JPMclvQkJ1rM

[8] *Wikipedia, Bootstrap (front-end framework)*, https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)?fbclid=IwAR1wirvjTjZDTO9wSWJnVtHQvV2FGCPzBKEq5c2EQgDuRfj6Eu1My3JWCH4

[9] *Wikipedia, CSS*, https://en.wikipedia.org/wiki/CSS?fbclid=IwAR03FWq0H1ANQm0lURmXqx0MWTn9AQOXGcnIoC9WF5I7Dn2rHt9gTD7ivbc

[10] *W3School, CSS Introduction*, https://www.w3schools.com/css/css_intro.asp?fbclid=IwAR3TiYYfJccIBZ9664n8D_TyBxQSvQXpuhXK6NDKU8aU8PyH0iXuk73FjU

[11] *Tutorialspoint, PHP Introduction*, https://www.tutorialspoint.com/php/php_introduction.htm?fbclid=IwAR2o9Fx_deRybR6EQ1BWhFUJkBej1ldhWG9zsiW6fMY6vkAAoHgiA9X_0kE

[12] *MDN web docs, About Javascript*, https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript?fbclid=IwAR0X8J1RIjh_127LCqK9ILCzFDcyL1vsDx0J6GbJsnG7mkyNYcBEnCbnTGM

[13] *Tutorialspoint, Distributed DBMS - Distributed Databases*, https://www.tutorialspoint.com//distributed_dbms_databases.htm?fbclid=IwAR3ZhpjXMBtJSD2V4s8FyvtEhXYHMdeU-BS6ZRKTukrMF3CasnKPXQinCZM

Contribution

As per the project description, we divided our tasks in such a way that both of us can actively participate in both the frontend and the database parts. According to this, the participant **1. Jannatul Fardouse** was responsible for the backend development And the participant **2. Jannatul Ferdous** was responsible for the frontend development. The design of database had been established by the active participation of both of us in our daily meeting. Finally, the term paper of this project has been written through continuous discussion by both of us.
For writing the report, we have choosen an online platform **"Overleaf"** which provides to work on LATEXin a common platform.