Advanced Management of Data

Project Term Paper

SUBMITTED BY

Name : Md Yasin Arafat
Matriculation # 612254

Name : Rawfid Kabir Shuvo
Matriculation # 623238

Program: Automotive Software Engineering

May 31, 2021

PROJECT NAME

Work-Together Meeting Management System

Work-Together Meeting Management System (WMMS) is a project for the final exam of Advanced Management of Data. This project is targeted for the online student meeting system arranged by the "Fachschaftsrat Informatik – FSR:IF" during the COVID-19 pandemic. In this project, A simple system is developed where FSR:IF and student can arrange and manage work together meetings.

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Contents

Figures

Tables

1 Introduction

The aim of this project is to develop a system where students and authority (FSR:IF) can access and collaborate to create and arrange work together meetings comfortably. The existing system has failed due to the COVID-19 pandemic. So this project will offer a platform to schedule meetings so that an online get together can be possible.

2 Project

2.1 Primary Requirements

The primary requirement of the project is to build the project using PostgreSQL database where program logic implemented in PL/pgSQL language. Also, A frontend which we mostly created using PHP, for user interaction with the data from the database. The diagram below shows how the frontend will connect and communicate with the database. We will see in detail how this architecture works in the later part of this report.

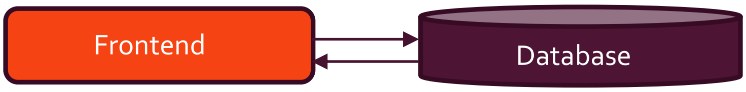


Figure-1: Structure [7]

2.1.1 Requirements Breakdown & Task Distribution

The table below illustrates the main requirements in more detail.

Application STACK	Task	Task Distribution
	Ui for home and login (default view)	Shuvo
Frontend	UI for FSR:IF view	Shuvo
	UI for Meeting selection for students	Arafat
	UI for Student profile view	Shuvo
	UI for meeting creation for FSR:IF	Arafat

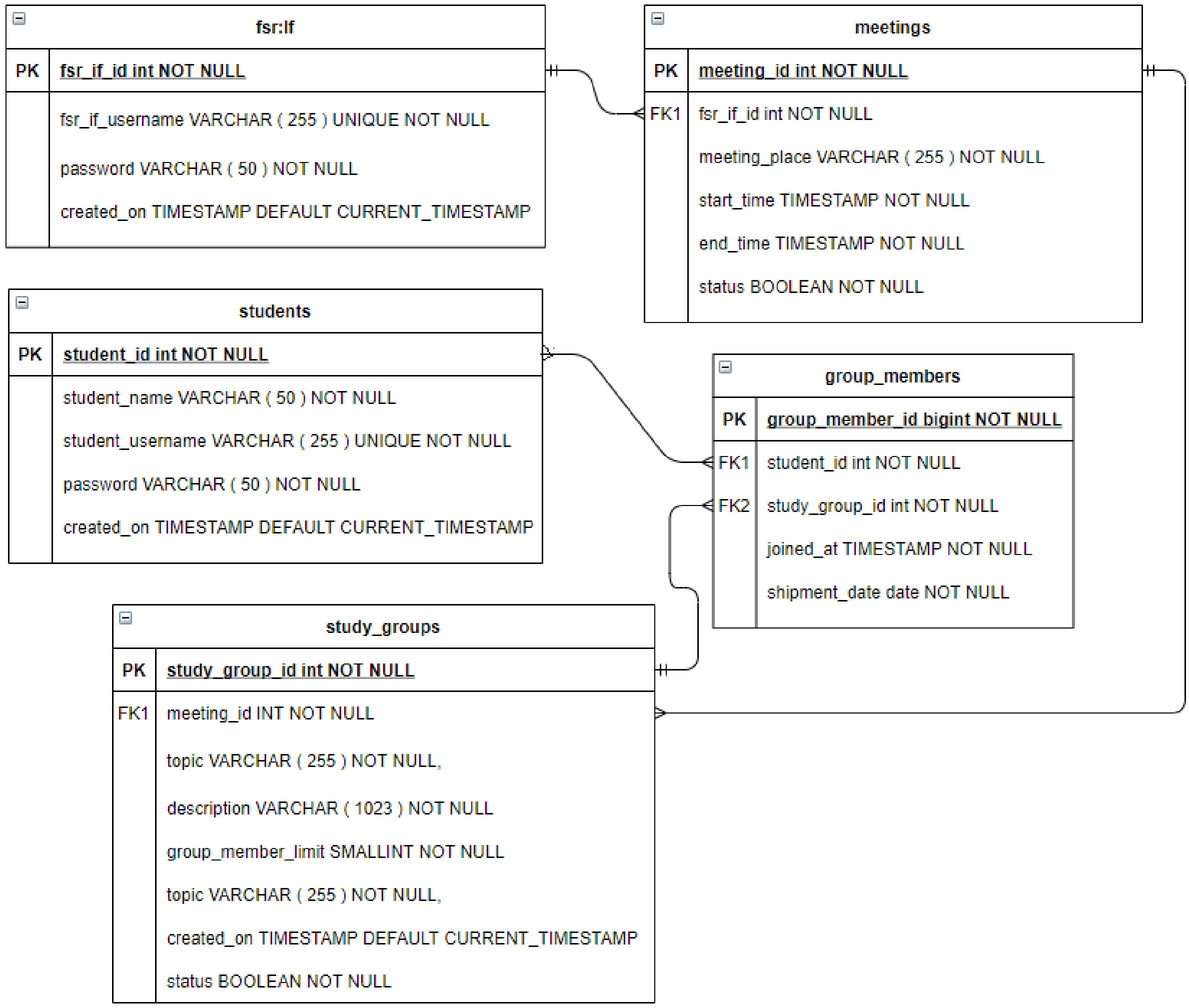
	UI for meeting handling for FSR:IF	Shuvo
	UI for study group creation and modification for student	Arafat
Database	Create tables	Shuvo, Arafat
	Create UML diagram for tables	Shuvo, Arafat
	Create relational schema	Shuvo, Arafat
Backend	Functions for adding meeting	Arafat
	Function for adding study group	Arafat
	Function for changing group status	Shuvo
	Function for editing study group	Arafat
	Function for editing meeting	Shuvo
	Function for editing student	Shuvo
	Functions for getting meeting details	Arafat
	Functions for getting study group details	Arafat
	Functions for setting group owner	Shuvo
	Functions for updating meeting	Arafat
	Function for updating study group	Shuvo
	Function for changing student name (demonstration purpose)	Arafat
	Function for removing meeting	Arafat
	Function for modifying group member limit	Shuvo
	Function for meeting time synchronizing with current time	Shuvo

Table-1: Main requirement breakdown

2.2 Database

2.2.1 Relational Schema

Relational schema denotes to the meta-data that describes the erection of data within a convinced domain. A relational schema for a database is the framework of how data is prearranged. It typically stipulates which columns in which tables comprise orientations to data in other tables, frequently by comprising primary keys from other table so that rows can be easily joined.



	fsr:lf
PK	fsr_lf_id int NOT NULL
	fsr_lf_username VARCHAR (255) UNIQUE NOT NULL
	password VARCHAR (50) NOT NULL
	created_on TIMESTAMP DEFAULT CURRENT_TIMESTAMP

	meetings
PK	meeting_id int NOT NULL
FK1	fsr_lf_id int NOT NULL
	meeting_place VARCHAR (255) NOT NULL
	start_time TIMESTAMP NOT NULL
	end_time TIMESTAMP NOT NULL
	status BOOLEAN NOT NULL

	students
PK	student_id int NOT NULL
	student_name VARCHAR (50) NOT NULL
	student_username VARCHAR (255) UNIQUE NOT NULL
	password VARCHAR (50) NOT NULL
	created_on TIMESTAMP DEFAULT CURRENT_TIMESTAMP

	group_members
PK	group_member_id bigint NOT NULL
FK1	student_id int NOT NULL
FK2	study_group_id int NOT NULL
	joined_at TIMESTAMP NOT NULL
	shipment_date date NOT NULL

	study_groups
PK	study_group_id int NOT NULL
FK1	meeting_id INT NOT NULL
	topic VARCHAR (255) NOT NULL,
	description VARCHAR (1023) NOT NULL
	group_member_limit SMALLINT NOT NULL
	topic VARCHAR (255) NOT NULL,
	created_on TIMESTAMP DEFAULT CURRENT_TIMESTAMP
	status BOOLEAN NOT NULL

Figure-2: Relational Schema

Relational schema has a very important role for developing the project. First we did this relational schema even before starting to write program. Because relational schema helps us to understand how the tables of the database are connected to each other using foreign key and primary key.

2.2.2 Uml diagram

UML stands for Unified Modeling Language which is used for specifying, visualizing, constructing, and documenting the artifacts of the systems. The bellow UML diagram represents the database tables of the system and their attributes as well as the methods.

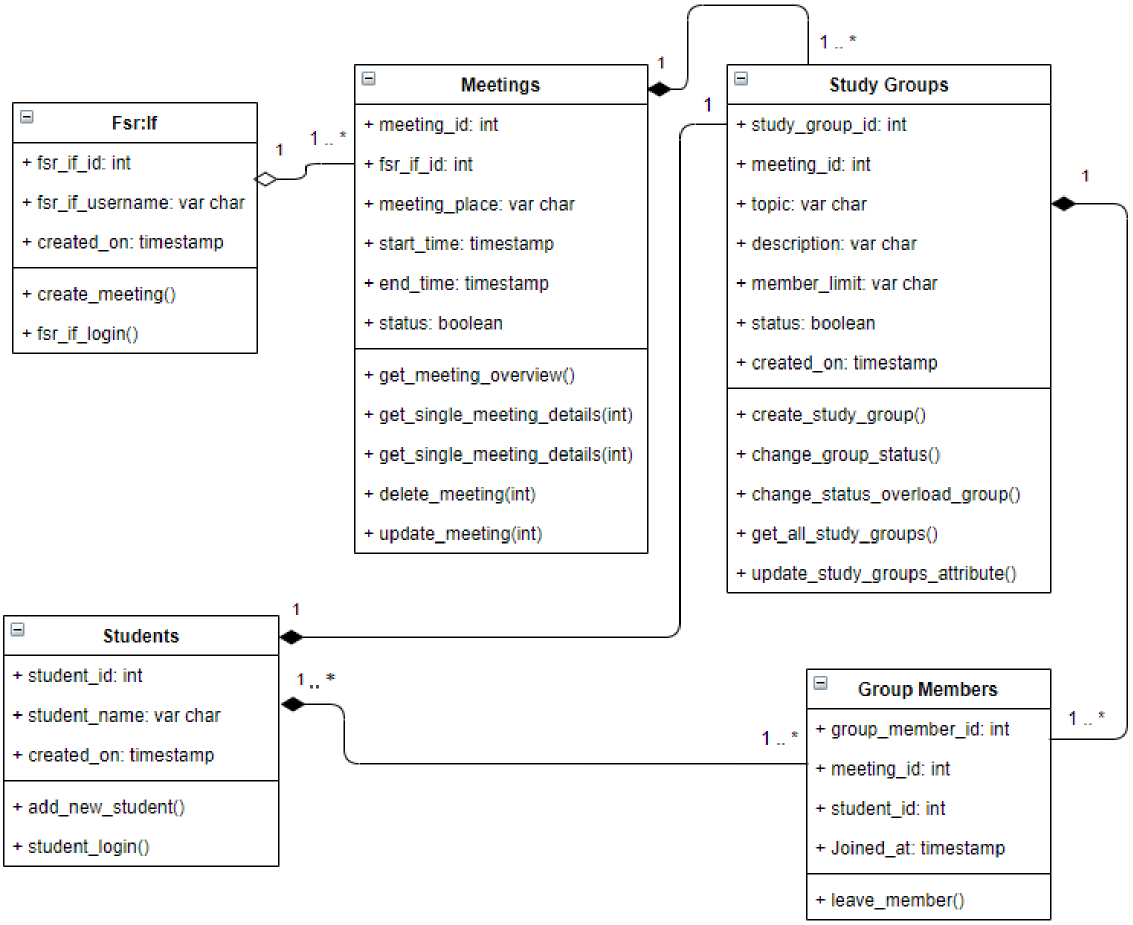


Figure-3: UML Diagram

UML diagram can be very helpful in creating a subtle image of the entire database and its functionalities to someone who is unknown of the project. It shows all the functions and relations among the tables. It is also a detailed representation of relation to understand the project. We created the rough diagram immediately after recognizing all the tables and functions. After that, it became easier to further complete the project.

2.3 Functional requirement Requirements

The system mainly has two users. Both student and admin panel have their separate tasks to handle. They perform different tasks according to their role and requirements.

Application	**task**
FSR:IF view	• Login • Create Meeting • Edit/modify meeting • Delete meeting • Change meeting visibility • Check Meeting/Study group activity
Student view	• Login • Select meeting • Create study group • Edit/modify study group • Delete study group • Join study group • Edit student name

Table-2: UI – Functional requirements

2.4 Project operations

This project is mainly consisting of two user interfaces. One for the admin FSR:IF and the other is for student. All the major operation is conducted by linking and fetching data from the database using PL/pgSQL. The FSR:IF admin can create, edit, delete meeting. Also they can see all the activities happening in the system. The students can find, select meetings to create study groups. They can also join other study groups or modify the group activities and so forth.

2.4.1 Home

The home menu is the index of the project. It consists of two login panels for both FSR:IF and Student. The admin login data will lead to FSR:IF admin view. And the student login data will lead to Student view.

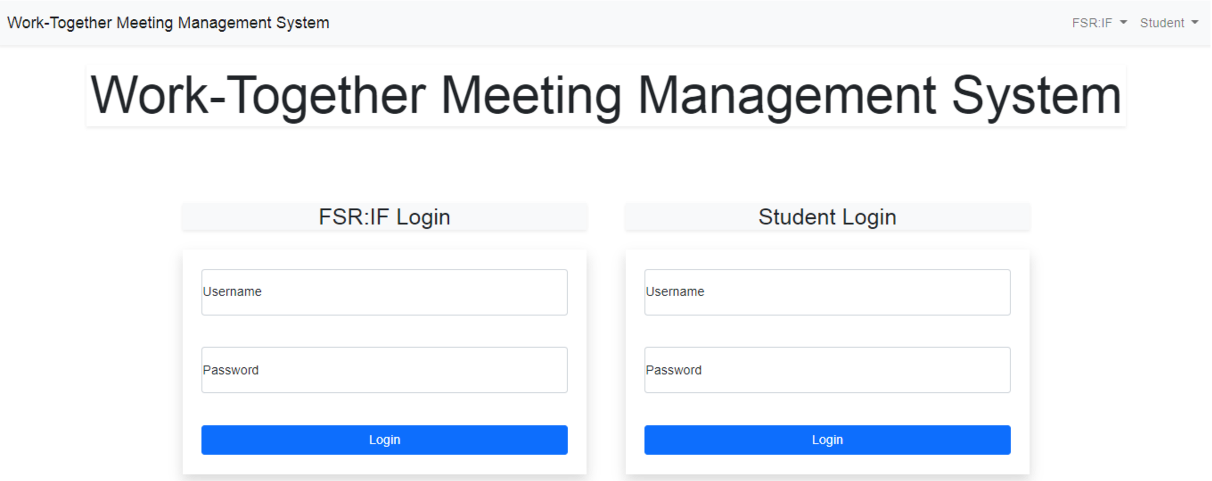


Figure-4: FSR:IF login

2.4.2 FSR:IF

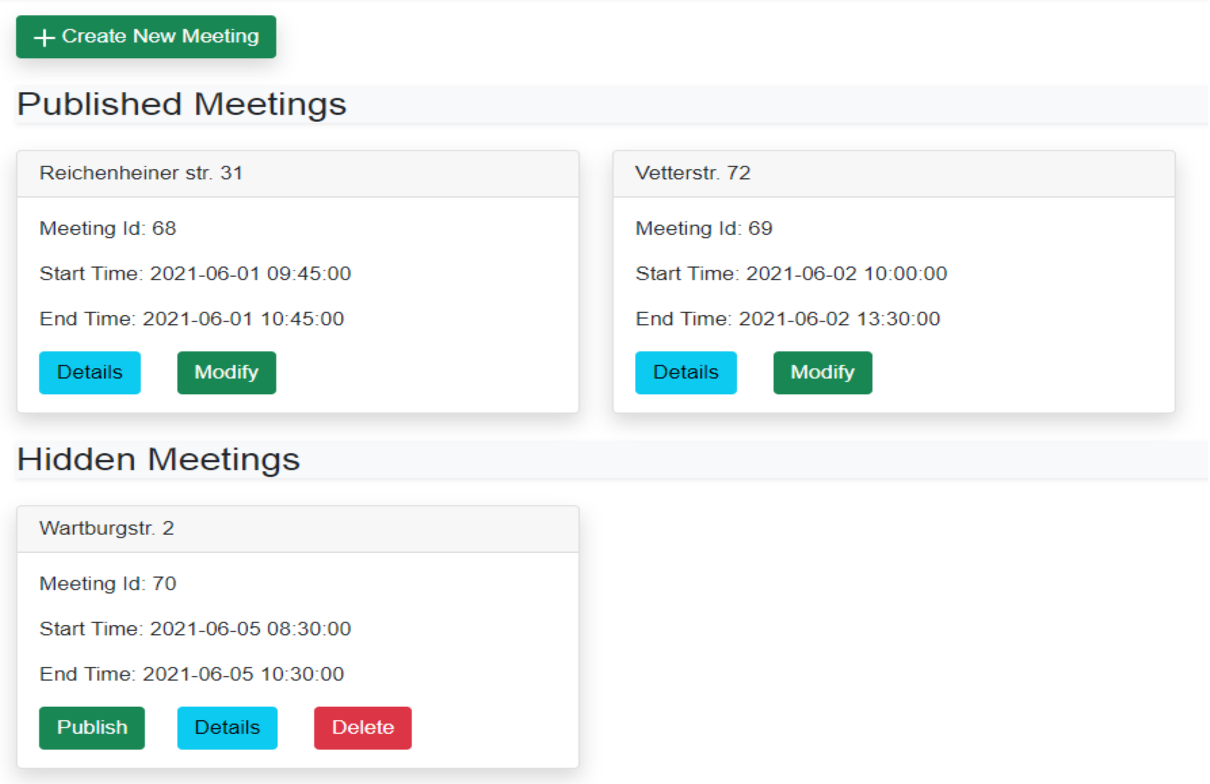
The FSR:IF menu is designed for the admin of the system. Below is a step by step explanation of how this interface works.

Login:

- Admin have their predefined username and password to initiate the process.
- For the requirement of this project, no complex login management system was created. The login system is designed for aesthetic user experience and demonstration purpose only.

Meeting view:

- The first view for the admin is a list of all the available and created meetings by them. Also they can see available unpublished meetings which they can publish.
- The meetings will only be visible to students when they are published.
- Admin can delete a meeting. Meetings can also be edited.
- All the attributes of any meeting can be completely changed by the admin.
- There is also a button in this view for the admin to create a new meeting.



Published Meetings

Reichenheiner str. 31

Meeting Id: 68

Start Time: 2021-06-01 09:45:00

End Time: 2021-06-01 10:45:00

Vetterstr. 72

Meeting Id: 69

Start Time: 2021-06-02 10:00:00

End Time: 2021-06-02 13:30:00

Hidden Meetings

Wartburgstr. 2

Meeting Id: 70

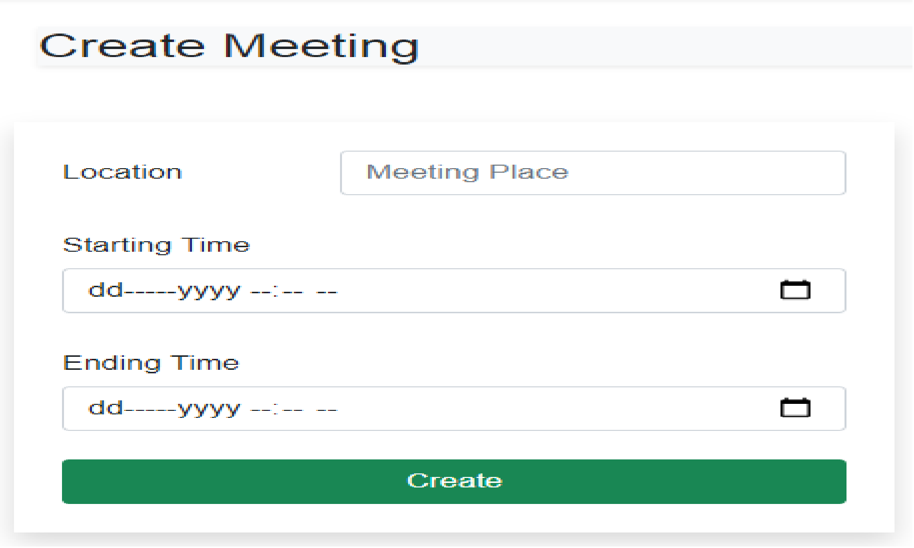
Start Time: 2021-06-05 08:30:00

End Time: 2021-06-05 10:30:00

Figure-5: FSR:IF home view

Creating a meeting:

- An admin can create a new meeting.
- Meeting is created by entering several attributes as input.
- Admin has to select a location, start and aend time for a meeting.
- Initially a newly created meeting is unpublished or hidden from the students. It only appears on the student view when admin publishes it.


Create Meeting

Location: Meeting Place

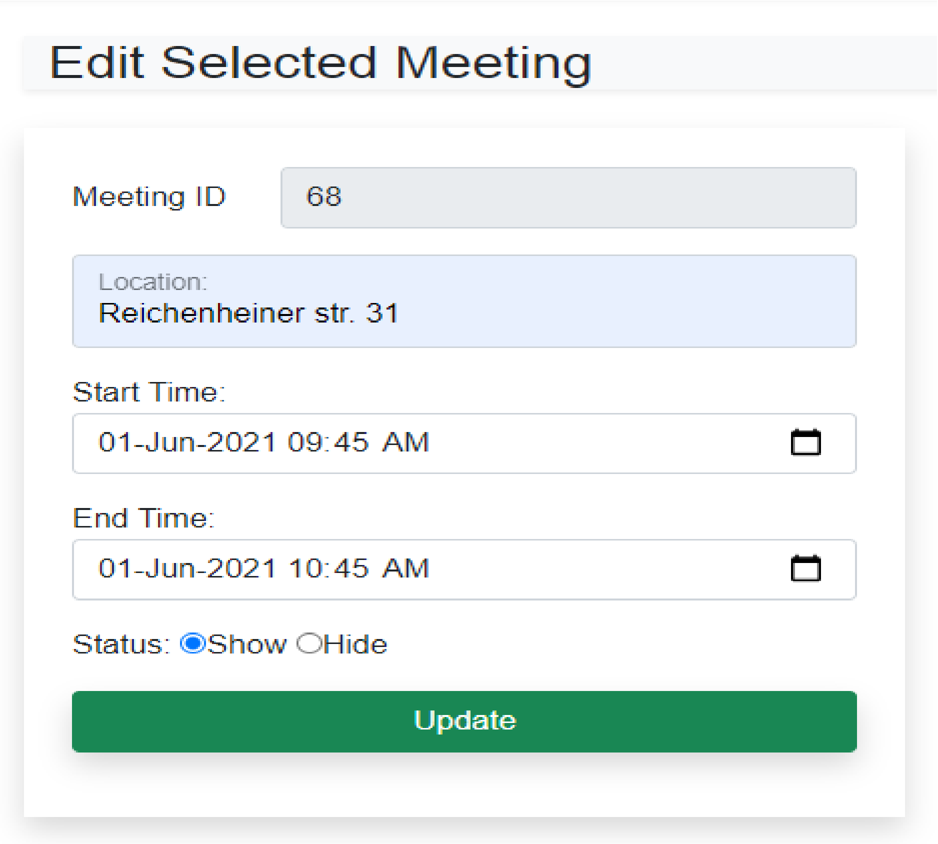
Starting Time: dd-----yyyy --:-- --

Ending Time: dd-----yyyy --:-- --

Figure-6: Creating meeting

Modifying meeting attributes:

- Admin has full control over meeting attributes.
- They can change location, times and visibility of a meeting.
- Modification is synchronized properly with the student view.
- A meeting can also be deleted before publishing it.



Edit Selected Meeting

Meeting ID: 68

Location:
Reichenheiner str. 31

Start Time:
01-Jun-2021 09:45 AM

End Time:
01-Jun-2021 10:45 AM

Status: ● Show ○ Hide

Figure-7: Editing meetings

2.4.3 Student

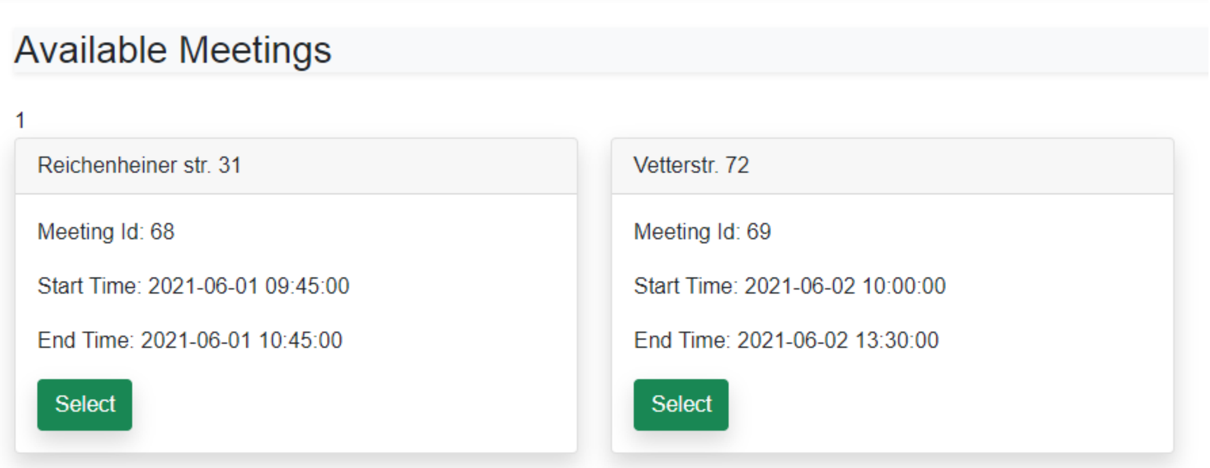
Student view is the user view of the system. Here students can login to access the system and see meetings. They can easily join existing study groups under individual meeting or create a new one. Students also have full access to their created study groups where they can edit some attributes. Below we will see some more functionalities in details.

Login:

- Students have their predefined username and password to initiate the process.
- For the requirement of this project, no complex login management system was created. The login system is designed for aesthetic user experience and demonstration purpose only.

Meeting selection:

- The student view starts with a selection of all the available meetings created by FSR:IF admin.
- Students can join any one of them.
- It is required for a student to select a meeting to go further and create or join a study group under that meeting.



Available Meetings

1

Reichenheiner str. 31

Meeting Id: 68

Start Time: 2021-06-01 09:45:00

End Time: 2021-06-01 10:45:00

Vetterstr. 72

Meeting Id: 69

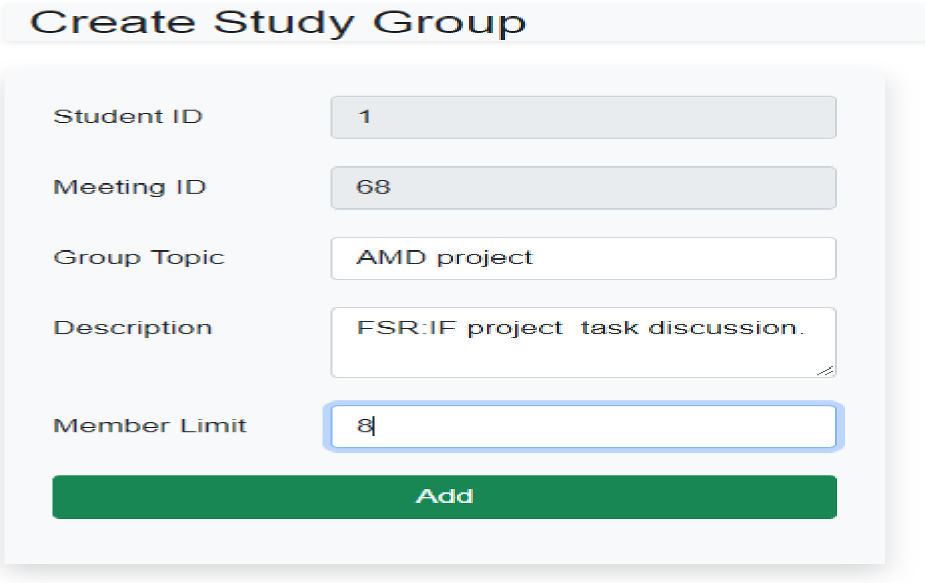
Start Time: 2021-06-02 10:00:00

End Time: 2021-06-02 13:30:00

Figure-8: Selecting meeting

Creating a study group:

- Students can create a study group of there own under any meeting.
- One student can join only one group at a time.
- They can not be on different places at the same time. So They are automatically removed by the system if the join a new study group.
- Students have to choose a topic, add some description and select a member limit for a new study group.
- The minimum limit has to be 2. Other wise the group will not be created.
- Creator student is automatically assigned as owner of the study group.



Create Study Group

Student ID: 1

Meeting ID: 68

Group Topic: AMD project

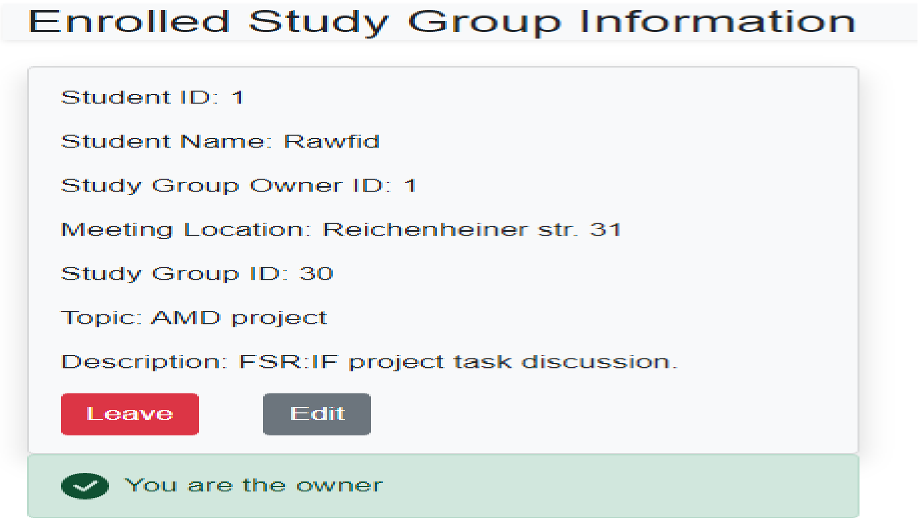
Description: FSR:IF project task discussion.

Member Limit: 8

Figure-9: Creating study group

Group activity view:

- Here one student can see the activity of a group that they have joined in.
- The view shows all the information including all the IDs, meeting place and times, group details and owner, number of student that can join the meeting and so on.
- A student is owner of the joined group if they have created it.
- If the owner of the group leaves the group, then the next joined student is automatically assigned as owner of the group.
- Only the owner of a study group is allowed to edit or delete a group. Others can only join or leave.



Enrolled Study Group Information

Student ID: 1

Student Name: Rawfid

Study Group Owner ID: 1

Meeting Location: Reichenheiner str. 31

Study Group ID: 30

Topic: AMD project

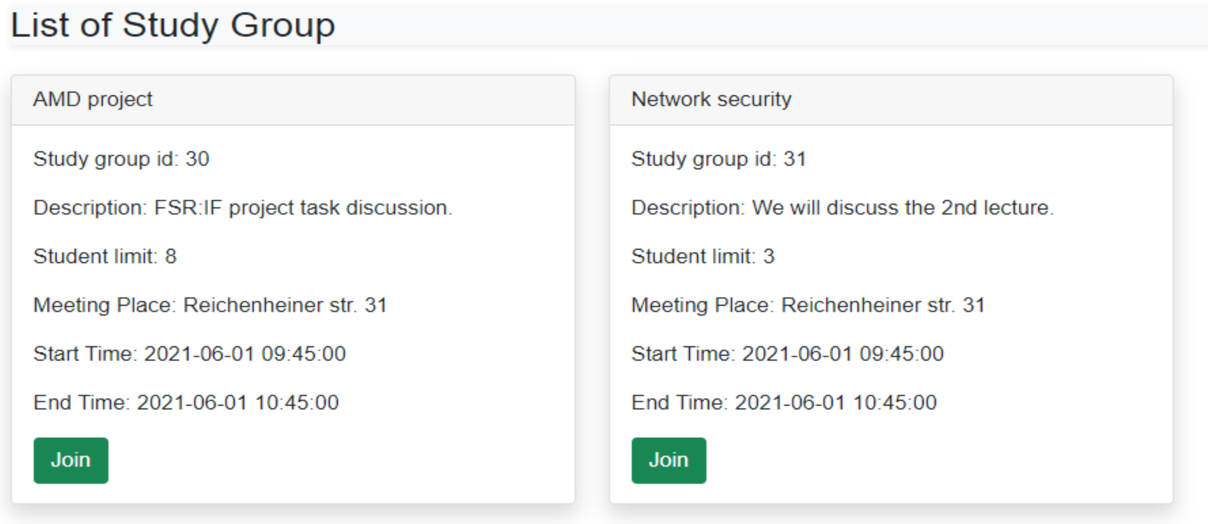
Description: FSR:IF project task discussion.

✓ You are the owner

Figure-10: Ingredient Settings Screenshot

Suggested group:

- Student view also suggests currently available groups under a meeting.
- The suggestion is customized so that student can only see the study groups of the time and location they have chosen.
- Currently joined group is not shown in the suggestion strip.



List of Study Group

AMD project

Study group id: 30

Description: FSR:IF project task discussion.

Student limit: 8

Meeting Place: Reichenheiner str. 31

Start Time: 2021-06-01 09:45:00

End Time: 2021-06-01 10:45:00

Network security

Study group id: 31

Description: We will discuss the 2nd lecture.

Student limit: 3

Meeting Place: Reichenheiner str. 31

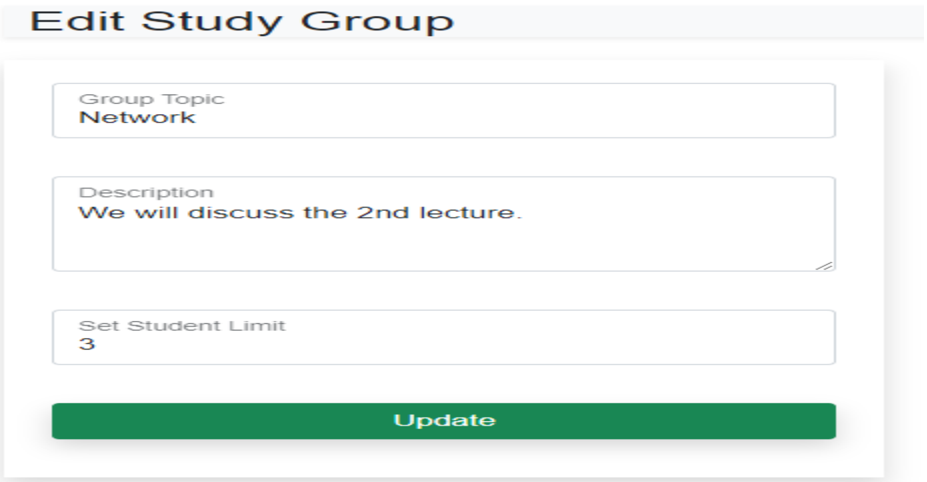
Start Time: 2021-06-01 09:45:00

End Time: 2021-06-01 10:45:00

Figure-11: List of study groups

Editing existing groups:

- A group owner can modify his group attributes.
- Most important of which is to increase or decrease the limit of members.
- If the limit is decreased after the limit was reached or current number of students is higher than currently chosen limit, the system will not force remove any existing student unless they leave themselves.



Edit Study Group

Group Topic
Network

Description
We will discuss the 2nd lecture.

Set Student Limit
3

Figure-12: Editing study groups

2.4.4 Functionalities of DB functions

Functionality	**DB Functions**	**Related Table**
Creating a meeting	add_meeting()	meetings

Creating a study group	add_study_group()
	group_owner()
		Study_groups
		Group_members

Show all the groups	already_joined_groups()
	get_all_study_groups()
		Study_groups
		meetings

Group limit reached	change_group_status()
	change_status_overload_group()
		study_groups
		group_members

Edit study groups	edit_group()
	leave_member()
	update_group()
		study_groups
		group_members

Edit meeting	edit_meeting()
	get_meeting_overview()
	insert_update_meeting()
	update_meeting()
	remove_meeting()
		meetings

Change student name	edit_student()	students
FSR:IF login	fsr_login()	fsr_if
Show all the meetings	get_meeting_details()	meetings study_groups
Joining study group	join_study_group()	study_groups

		group_members
Publishable meetings	only_hidden_meeting_list()	meetings
Student details view	student_details_all()	meetings group_members students study_groups
Student login	student_login()	students

Table-3: Database Functionalities

2.4.5 Prerequisites

The zip file needs to extract at the server's htdocs directory (for local server) or public_html directory (for remote server).

APPS	REQUIRED PACKAGES
App	PHP==7 or above [3] Bootstrap==5 [4] Local Server==XAMPP (Windows) or MAMP (Mac)
Database	PostgreSQL = 13 [1] Local postgreSQL client: pgAdmin4 v5 [2]

Table-4: Required Package for Installation.

2.4.6 How to Run

After starting the server (XAMPP/MAMP), we just need to hit the URL (directory of our app at the server) at the browser. In our environment the URL was like:

```
http://localhost:8888/WMMS
```

Now the application will run and is ready for showing the functionalities of our Student Meeting Management System (WMMS).

3 Technology Overview

3.1 PHP

PHP is a general-purpose scripting language especially suited to web development. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. [6]

The main interface of this project is entirely created using PHP. All the database logic are implemented using PL/pgSQL. The executable data were shown in the application using PHP. Working with PHP is really easy and the development environment is also very user friendly. So we used PHP as our main language for user interface interaction.

3.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

3.2.1 Bootstrap

Bootstrap is a free front-end - CSS framework. It contains HTML and CSS- based design templates for typography, forms, buttons, tables, grid systems, navigation and other interface design elements as well as additional, optional JavaScript extensions.

Initially we did not use this language because it is mainly used to stylize the project. We almost entirely coded our project interface on PHP. After implementing all the logics and running a number of tests to verify all the required objectives and tasks, we finally used CSS to give the project a visuall aesthetic look.

3.3 Javascript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. [5]

We used javascript for some important functionalities like exception handling warnings, page transformation functions and debugging.

3.4 Github

Github is a provider of internet hosting for software development and version control. It is very popular platform among developers and young enthusiasts. The company is owned by tech giant Microsoft since 2018.

Github is a great platform for shared project works. Since we are two person working in this project, the platform has helped us immensely in this pandemic situation where working together and keeping distance is very challenging. We could easily share ideas and implement them while staying in our own rooms.

4 Distributed database

Distributed database is kind of a database where two or more copies of files are created and located in either same or different networks for a site.

4.1 Distributed database with this project

This WMMS is not a big project or database to be used as a distributed database. But a bigger picture can also be seen where group of university's working together to create a platform for students. A platform where students from different universities can come online and share their knowledge together. For a project something like that, a bigger and spread network is required where different universities will have their own shared databases to create a combination of distributed database system.

4.2 Distributed database with this project

Distribution of a database can be done in some ways like,

Data fragmentation:

It is the process of dividing a table into smaller tables. This fragmentation has to be done in a way that from the fragments, the original table can be reconstructed. The fragmentation technique,

- Increases efficiency because irrelevant data is not available in all sites.
- Increases efficiency because data is stored close to the site of usage.
- Local query optimization technique is also becomes efficient.

Data Replication:

It is the process of storing copies of database in different sites. If we store the database of our system in different sites,

- Even if one site is down, the system will not collapse since there is another copy is existed in another site which increases reliability.
- Query processing can be done with less bandwidth usage because of local database.
- Local database ensures quicker response which increases speed.

5 Conclusion

This project "Work-Together Meeting Management System" is a PostgreSQL based project where most of the functional logic was implemented inside a database. Working on this project was entirely a new experience for both of us. Implementing logics directly on the database and fetching them to cast on the interface was a bit challenging at start. But day by day, we learned and things got easier. Under the situations of COVID-19 pandemic, this project was cordially created to help the great cause to support the FSR:IF team to create an online environment for already a great initiative of arranging meetings for students.

References

[1] "PostgreSQL 13 Released!" PostgreSQL News, 24 Sept. 2020, https://jpostgresql.org/about/news/postgresql-13-released-2077/

[2] Page, Dave. "PgAdmin - PostgreSQL Tools." PgAdmin4 v5 Release, https://pgadmin.org/. Accessed 31 May 2021.

[3] "PHP: News Archive - 2021." PHP 8.0.6 Released, 6 May 2021, https://php.net/archive/2021.php#2021-05-06-2/

[4] "Bootstrap 5.0.1." Bootstrap Blog, 13 May 2021, https://blog.getbootstrap.com/2021/05/13/bootstrap-5-0-1/

[5] Wikipedia contributors. "JavaScript." Wikipedia, 19 Nov. 2001, https://wikipedia.org/wiki/JavaScript

[6] "PHP." Wikipedia, 27 May 2021, https://wikipedia.org/wiki/PHP

[7] Daniel Richter (2021. April 26) "AMD Project Task" [PDF File]. Retrieved from https://www.tu-chemnitz.de/informatik/DVS/lehre/AMD/Project/AMD%20-%20Project2.pdf