STL SITE

Project Specification & Database Design

Group Members:

1. 412855412 加梅莉
2. 412855420 陳麗娟
3. 412855511尹路加

Course : Open-Source Practice

Instructor : Feng-Cheng Chang

Date : 21 May 2025

Table of contents:

1. Demo Scenario Overview.

Final Demo Description:

The end demo will display a web-based club management system (title: STL Site) with fundamental modules facilitating member and admin functionality. It will demonstrate the system's ability to handle different member applications, attendance tracking, and documentation announcements, with variable user access.

Specific Feature to Present (High-Level Descriptions):

Role Based User Differentiation:

- Access by Admin (User Manager) and Interactive User
- Role-based dashboard visibility

Membership Application Module

- New member's web form
- Backend data storage and validation

Attendance System

- QR code generation for meetings
- Attendance logging with history view

Post Announcements/Document Management

- Upload interface of admin-only file and project folder

User Action(Triggers of Functional Flows):

Interactive User:

- Clicks on "Apply Now" on the homepage to trigger the membership application flow
- Submits the application form

User Manager:

- Logs in to continue with admin features
- Clicks on "Generate QR Code" to trigger a new meeting
- Views attendance record by selecting an old meeting

Functional Components of the Application (Partial/Pseudo Implementation):

Working Fully or with Real Backend:

- Submission of membership form:
    1. PHP validates proper input and stores in MariaDB
    2. Feedback on success/failure is shown

Tracking of attendance:

- PHP generates QR code using library(for example, PHP QR Code)
- Attendance data stored and retrieved from database
- Admin views past attendance records

Functional with Pseudo Implementation or UI Mockups:

- Role-based dashboard(if not implemented yet, can be displayed through static content or conditionally displayed views)

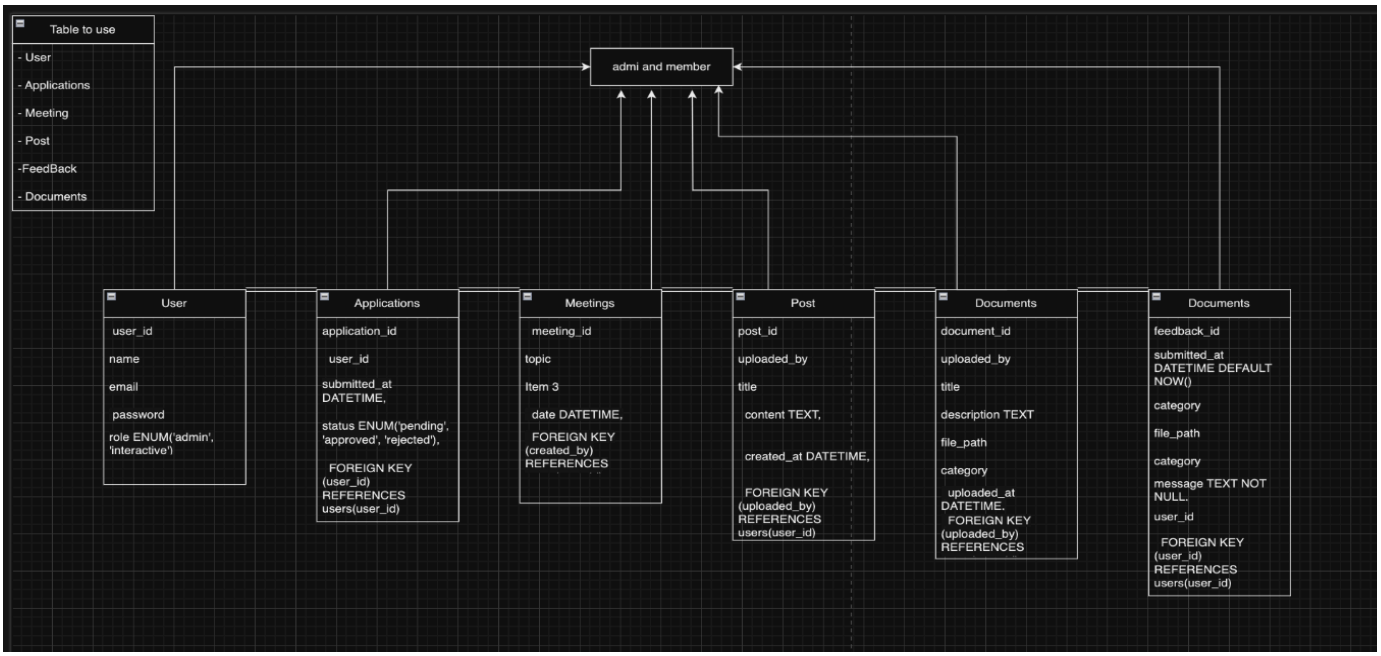
File/document upload module (can be stubbed to show upload interface without actually storing anything)

2. Planned URL Endpoints.

Number	URL Path	HTTP Method	HTTP Variables	Session Variables	Database Operations
1	login.php	GET, POST	username password	user_id, username	SELECT
2	about.php	GET	none	user_id (optional)	none
3	apply.php	GET, POST	application form data	user_id	INSERT
4	contactus.php	GET, POST	name, email, message	user_id (optional)	INSERT
5	feedback.php	GET, POST	rating, comments	user_id	INSERT
6	home.php	GET	none	User_id, username	SELECT
7	post.php	GET, POST	post_id, content	user_id	SELECT, INSERT, UPDATE, DELETE
8	register.php	GET, POST	Username, password, email	none	INSERT

3. Database Design.

a. Entity-Relationship Diagram (ERD)



b. Relational Model

Table	Primary Key	Foreign Keys
`users`	`user_id`	—
`applications`	`application_id`	—
`meetings`	`meeting_id`	—
`attendance`	`attendance_id`	`meeting_id`, `user_id`
`posts`	`post_id`	`uploaded_by` → `users(user_id)`

documents	document_id	uploaded_by → users(user_id)
feedback	feedback_id	user_id → users(user_id)

c. Normalization (3NF)
How the model meets Third Normal Form (3NF):

- **1NF (First Normal Form):** All columns contain atomic values; each table has a primary key.
- **2NF (Second Normal Form):** All non-key columns are fully dependent on the entire primary key.
- **3NF (Third Normal Form):** There are no transitive dependencies; each non-key column depends only on the primary key.

Design decisions to reduce redundancy and ensure integrity:

- **Separated posts and documents** into two different tables to avoid null values and reduce semantic confusion between text-based announcements and file uploads.
- **Used foreign keys** (e.g., linking attendance, feedback, and uploads to users) to maintain data consistency and avoid duplication.
- **Used ENUM types** where applicable (e.g., role, category) to enforce controlled values and reduce user input errors.
- **Applied unique constraints** on emails and QR code tokens to prevent duplicates and ensure correct login/meeting behavior.