

Comp5531/4 Winter/Spring 2022
CrsMgr Group-work Assistant(CGA)

Demos Week of Apr 7, 2022

Your team is required to design a working web application driven by a relational database system; the application needs to be “realistic” and is described below. This is followed by an implementation of the system using CSS//HTML/JavaScript/ MySQL-MariaDB/PHP; it should be deployed and be demonstrated with realistic data and would be used for the group interaction during the realization of this project. The application is a two/three tier system, which uses any popular web browser at the client side and HTTP server with PHP parser and a MySQL/MariaDB database at the server side. The system is expected to support all “representative” queries and operations of the CGA system. Each team member will be responsible for the entire project with each teammate being primarily responsible for an equitable and clearly defined portion of the project, to be agreed on by the team members. You will be provided a 20 minutes time slot to (remotely) demonstrate that your system performs as expected in the week starting 7th Apr. 2022. A schedule for the demos will be posted on CrsMgr (first come/first reserve). The project report as well as tar-gzipped (or zipped) file with the entire system including scripts to install and initialize a virgin system and details of all updates and bug fixes made since the demo must be uploaded to CrsMgr soon after the demo time.

Project Description

Use MySQL/MariaDB Database Management System to develop a **CrsMgr Group-work Assistant(CGA)**. The application includes a collection of tables and a web-based user interface to provide the functions required to support various activities and services of the system. The CGA system enables teammates for a course managed by CrsMgr to share information and ideas via pages and shared files. Each set of discussion pages and files are associated with a group-marked-entity for the course and all comments and a web page gives the trace of who did what when for the marked entity. In addition, the system allows the discussion web pages to be seen by the instructor and TAs of the course to add their suggestions and comments. All additions made to any discussion web page must not change the existing contents! Hence the existing content would be in read-only mode. The access to the group contents is accessible by the team-mates. It is possible that the teammates could change over the term: such changes to a group are made by the instructor of the course; however access to the contents up-to the date of the change should be accessible -read only mode by a departing team-mate as long as this teammate is registered in the course.

The CGA system administrator can insert a new course/section, the instructor, and the TAs, The instructor can insert the set of students, the groups in the course-section, each group marked entity (group assignment, group project)and changes in the group make up over the term.

CGA is to provide services such as:

- Set up a course-section and the instructors for it
- Allow the instructor to set up and update the students, TAs, groups, members for each group and the group marked entities
- Initiating and coordinating discussion, comments and files for the current marked entities for the groups teammates.

- No changes in the content created for a marked entity is allowed after the corresponding deadline/submission date.
- Post and email notices of the group meetings, their agenda, and minutes of these meetings,
- Polling for various options being considered for a possible resolution of issues
- Provide facility for: on-line discussions, plan and share activities
- Internal text system accessible to the teammates only(are stored in CGA) discussions in this text system could be private to the current teammates

Thus only current teammates of a given group marked entity can access material for that group marked entity However, there could be an additional feature of public postings where members of different groups could have access and allow communication among groups for a given course-section.

Requirements Specification

You are required to develop a database system that will store at least the information about the following entities and relationships:

- *Details on students:* ID, password, other personal information such as name, and email address.
- *Details about groups:* Records of information on each group, leader of the group, list of members belonging to the group: include facility to update members (add, remove members, change group leader)
- A student could be part of only one group for a given marked entity but could be moved to another group for another marked entity..
- A group leader must be in the team: the team leader could change
- Detail on file uploaded and the permission: Each file uploaded by a team mate can have a profile which indicates who can do what with it. Content can be classified as view only, download and update (replace) or delete..
- The TAs and instructor have access to the contents created by any group in read mode and are able to add comments and feedback
- The instructor of the course could create the list of students, the groups, its members and any the group modification, assign TAs, etc.

A user with CGA's system administrator privilege has the full access on all services such as creation, deletion, editing of: courses, course sections, term dates, instructors, members and groups. The CGA's administrator could also post notices accessible to all current users of CGA.

A student user of the CGA's privileges to the system for the course-section would end at the end of the term for the course section.

CGA system, by default, has one member with user-name admin and password admin created initially(Both of these must be changed during the first login). Only members with administrator privileges can add course/section/instructors. The instructor of a course-section can upload the list of students and the group details, The instructor and the TAs have access to the discussions and are able to add comments and provide answers. The comments etc. posted by the instructor or the TAs could be accessible to one or multiple/all groups. As a bonus, CGA could have an internal email system. The emails are internal to the system and delivered to a user's internal in-box and copies of internal email sent out are kept in an internal sent email box(as illustrated below).

✉ My Mail				
Mail Folder: Inbox ▼		<input type="button" value="Add/Modify Email"/> <input type="button" value="Send Mail"/>		
15	mail/page	<input type="text"/>	<input type="button" value="Search"/>	Page 1 of 68 1 2 3 Last
	Mail Title	From	To	Date
1	Re: IDEAS2021: In digital library	conf879475	admin21C	2021-11-18 00:51:26
2	Re: IDEAS2021: In digital library	conf851628	admin21C	2021-09-13 09:23:19
3	Re: Re: Re: Final paper upload	conf945213	admin21C	2021-07-22 11:57:32
4	Re: Re: Re: Prelimanry program -again!	conf879475	admin21C	2021-07-17 08:02:12

✉ My Mail	
Date:	2021-11-18 00:51:26
From:	conf879475
To:	admin21C
Subject:	Re: IDEAS2021: In digital library
Content:	<p>Hello,</p> <p>I hope this email finds you well.</p> <p>We haven't received a hard copy yet. Would you please give me an update on the same?</p>

The system provides to the teammates, the instructor and TAs a summary page with the complete list of the files for a given marked entity and for each one the person who uploaded it, downloaded it, replaced it or deleted.

The above are the minimum requirements: it is expected that each group would discuss this application as well as talk to people knowledgeable in the domain to enrich the application and make it more realistic. The implementation should have an CrsMgr-like interface!

Implementation Details: Functions and Reports

Design and implement the user interface using HTML, CSS, PHP, JavaScript for the required operations described above. Details on how to use these will be provided by the lab instructors. You also have access to a text book on the topic. Populate your tables with enough data to show the functionality of your system. The system should support at least the following functionalities through its interface:

1. Create/Delete/Edit/Display a user.
2. Create/Delete/Edit/Display a group.
3. Create/Delete/Edit/Display list of groups and their teammates.
4. Teammate's ability to post texts, files as well as to view posts by other members and comment on them; download files and replace them.
5. Teammates can re-post or view posts of only the group that they currently belong to.
6. Discussions etc, and files would be associated with each marked entity in a given course
8. Users home main page shows the best and latest posts from their groups and friends.

9. Users can send a private message to their group(s).
10. Each user could be associated with one or more course in the same term

Note: The system would be run by an administrator on a private server (could be on a cloud). . Each team is to implement the CGA system. The team must have a designated leader (DBA) who coordinates the work. It is expected that the team is to meet regularly during lab hours and each meeting is logged with the names of group members participating. Such logs etc. should be implemented in CGA In addition to the implementation, the team is required to prepare a machine printed preliminary report documenting their project which must include the E-R model, the DB design and its normalization. This preliminary report is due before the demo. The implementation of the system including the database system must be demonstrated by each team at a predesignated time. Remember that the database is the foundation of your system and if it is not sound, it will bring the whole structure down. Just like the building foundation, the database is not visible; what is visible is the structure above ground (in this case, the interface and its functionality). It must be attractive, non-intrusive, non-intimidating and functional (AKA user friendly).

1. Develop an E/R diagram for the entity sets and relationships described above. Determine the attributes and the multiplicity of the relationships. The design should be as compact as possible without sacrificing the required objectives. Make sure you state clearly any reasonable assumption made in your design, which is not specified in the requirements specified above.
2. Convert your E/R diagram into a relational database schema. Make necessary refinements to the schema, if possible. Identify various integrity constraints such as primary keys, foreign keys, functional dependencies, and referential constraints. Make sure that your database schema is at least in 3NF.

Report & submissions

Your report should include: project description, the assumptions, the design decisions made and give rational explanations for all assumptions, the limitations, the applications supported, E-R diagrams and relational database design; responsibility of each member of the team and the joint responsibility; the interface design rational, the script to create the database (clearing any old tables and data that may be hanging around) and populate the tables. Include also the contents of the tables, the queries, and their responses. In addition, the source code of any procedures, triggers, programs, the dump of the database, etc., must also be submitted. Include the screen shots of the interfaces for various members and a log of the entire session. Also, submit the on-line log of work done (who, when, what, why) mentioned above.

All programs/code/scripts must have adequate internal and external documentation. The project submission would also include all codes etc. in a single directory with appropriate sub-directories. This directory could be compressed to a single file with its size limited to 15 M bytes which means you need to upload a tar-gzipped ball (or rar, zip, etc.). This tar-gzipped ball must also include a report in either LaTeX or OO word processor form. Your project report documenting your project and must include details on:

- The design of the DB using an E/R data model.
- Its conversion into a relational model satisfying at least 3NF.

- The user-interface for each supported application and reports.
- A sample session for each application (user guide).
- An installation guide.
- All codes and scripts.(list only the directory structures and the names of all modules- not the actual code: the code must be uploaded in the tarball to CrsMgr)
- List of team members' contribution as its last part. The title of this section should be "Contributions", indicating who did what in the project. It is wise to be realistic since the lab instructors will also evaluate each team member's contributions.

Also include a "READ-ME" text file detailing the team (group) ID, the names, student IDs and ENCS accounts of each member of the team, the group account, PW, URL for the project and list of files included in the submission. Also include: the user IDs and passwords of all users for testing the system and any other information required to install, run and test the system during the final grading by the course professor on his own server.

For the demo, your system must be installed and running on a designated ENCS server.

If you have made changes, enhancements, fixed bugs, etc. since your demo then highlight these in the READ-ME file and make sure it is also reflected in your final report. A working version of the project should be presented by the group to the lab instructors during the presentation. Every member of the group **MUST** be present during their demo.

Note 1: The document report should be included in the upload. The source of the code you demonstrate at demo time should be uploaded with your report at demo time as part of your project report.

Note 2: Your project report must include official names of the team members, student ID's clearly appearing on the cover. Inappropriate submission will be penalized



Bipin C. DESAI