COMP 353 DATABASES Fall, 2019

Bipin C. Desai

Main Project

Design, Implementation and On-line Deployment of a Share, Contribute and Comment(SCC) System

Due: Draft December 4, 2019 DEMOs: Dec. 5/6, 2019 Final: Dec 9. 2019

The Project

Your team is required to design a relational database system for a "realistic" application described below. This is followed by an implementation of the system using MySQL(MariaDB), HTML, PhP, JavaScript. The application is a multi-tier system, with any popular web browser at the client side and HTTP server with PhP parser and bridge to the back-end MySQL(MariaDB) database at the server side. The system is expected to support all "representative" queries and operations outlined in the following description of the SCC 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 minute time slot to demonstrate that your system performs as expected on Dec 5th or 6th 2019. 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 'clean' 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 Share, Contribute and Comment System (SCC). The application includes a database made up of a collection of normalized(at least 3NF) tables and web-based user interfaces to provide the various functionalities and contents provided by SCC. The SCC system enables members of an event such as an academic or community meeting, a social gathering(anniversary, wedding, etc.) to share information(text, photos, videos) and manage get-togethers and keep in touch. The SCC system allows participants in an event in addition to the main happening to create one or more groups: each being made up of a list of other participants with whom to share contents; and to view and add comments and content (if enabled by the curator of the group) within the system. The default group is all participants who have access to the main content and could post and comment on it. It also allows event participants (or group members)to interact among each other via 'simulated' instant messaging. Furthermore, SCC allows participants to hold on-line discussions.

Requirements Specification[1]

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

The people involved in the system are the following:

- System Administrator: This is the person who configures, installs, deploys and maintains SCC; adds events and assigns event managers to the new events. Events for non-profit organizations are complimentary and have no charges; others are charged as determined by the current fees.
- Controller: This is the person who sets the charge rate for usage of the system. The charges are for events and the resources used. The basic fee for an event includes an established storage and bandwidth limits which could be changed from time to time. Recurring events have a discount. Resources used over the thresholds have a rate which could change over time and are for the amount of extra storage used and the extra bandwidth used by each event. An event and all its contents are maintained for a pr-established time period(event life-time); the event manager could extend this period for extra charge. Once this time period is over, the event is considered archived and its contents are no longer accessible. However, the details of the event are accessible to the event administrator and all the participants for a period of seven years.
- Event manager: This is the person who administers an event. The event manager is required to provide details such as name, address, phone number, event(s), if it is for a non-profit organization or private, debit details(bank information, on-line payment system etc. would have details such as: name, address, account code, etc.), register an event and manage it. A person could be an event manager for many events and the type of events could be for non-profit organizations, other types of organizations, family, or community. The even manager would set up a home page for the event using a template provided by SCC
- Participants are people who have been associated with one or more events; Details of the participants include: Name, email address. The list of participants is supplied by the event manager and if the person has not already been a participant in a previous event, is registered as a user and becomes a participant for the event and is sent an email

[1] To be worked out and ambiguities resolved in tutorials/labs

with an URL and a one time code. Persons who have other roles in SCC could also be participants in one or more events.

- Details on participants: ID, Name, password, event, other information such as organization, etc.

- The event admin is the only person who can manage and update the event and add or update the participants. The details about the events and the participants are loaded by providing an appropriate CSV file as in the warm up project. A user with the administrator privilege has the full capability on all services such as creation, deletion, editing of members and groups. The administrator could also post any contents.

- A participant can create a new group and hence become the manager of the group, invite other participants of the events to join the group and admit those participants who apply to join the group; post new items for the group. The group pages are not accessible to anyone other than the group members; however any participant of the main event can request to join a group. Membership of a group is visible to all members of the group and they can send private communication or comments to another member in the group. NO access is allowed to the group contents to participants not members of the group.

- All event posting not intended for a group is accessible to all participants of the event and they can add contents and comment on any contents.

- Status of a an event is active or archived. An active event allows the participants to share, post and comment. An archived event only allows access to the basic details about the events.

- Only the event manager (and of course the system administrator) can change the status of a participant or add or remove participants for an event.

- When installed on a 'real' server, SCC has an email system that can send out email messages to indicate new contents to the participants of the event; such messages are also copied to a SCC's internal in-box of the participant. However, there is a restriction of sending emails by AITS: no email messages are allowed to be sent out of the system. Hence when run on AITS system, no external 'real' emails would be sent out by the system; email messages have to be simulated by delivery only to the recipients in-box and 'sent email' box of the sender. Any user can consult these internal boxes at any time. In a 'production' system on a server, emails should be sent out as well as recorded in the internal messaging system.

- Details about groups: Records of information on each group, owner of the group, list of participants belonging to the group(group members). Owner of the group can add new members or remove members from the group. Members of a group can share a forum of information such as photos, videos and posts. Adding any member into a group requires the knowledge of the person's SCC ID, email address, first name and DOB.
- Each SCC user has a home page which has an email facility, list of of his/her events and access to the events contents as well as the contents of each of the groups where he is an 'owner' or member. The user can switch to one of these events by selecting it and hence assumes the role(s) s/he has in that event. The functionality accessible would depend on the role in the event. The group owner has a feature to view and manage the permissions to any content; the permissions can be changed only by the owner.
- Detail on contents and the permissions: Each content added by a member can have a profile which indicates who can do what with it. Content can be classified as view only, view and comment or view and add or link to other contents.

  These are the minimum requirements in your database. It is expected you will discuss this application among yourselves as well as talk to people knowledgeable in the domain to enrich the application and make it more realistic.
- BONUS[2] Since each member may belong to many groups(family, extended family, colleagues, friends, etc.) s/he may want to post some contents in more than one event/group. Create mechanism and interface for this function.
- BONUS Allow use of a facility to ensure that the contents s/he added (regardless of where) is deleted after a preset period which is before the expiry of the event's life time

Implementation Details: Functions and Reports

Design and implement the user interface using HTML and PHP for the required operations described above. Details on how to use PHP and HTML with MySQL will be provided by the lab instructors. Populate your tables with enough data to show the functionality of your system. The system should support at least the following functionalities(in the spirit of the requirements specifications) through its interface:

1. Administrative features: management of system, users and contents
2. Create/Delete/Edit/Display a user/member/participant.

[2] The bonus items are applicable only if all the above specs have been satisfactorily implemented

3. Create/Delete/Edit/Display a group.
4. Create/Delete/Edit/Display list of members in a group.
5. AN event's particiapnt request to join a group in the event.
6. Member's ability to withdraw from a group.
7. Member's ability to post texts, images or videos as well as to view posts by other members and comment on them.
8. Members can either post or view posts of only groups that they belong to.
9. Member's main page shows the latest posts from their groups.
10. Members can send a private message to one or more members of a group where s/he is a member.
11. Report of groups or members by specific category such as interest, age, profession, region, etc.
12. Ability to organize an event for the group by voting on date/time/place from a set posted and/or alternates suggested by one of the group members
13. Registry and/or Gift exchange (secret) ideas among a family (secret) or a group.

Note: Each team is to implement the SCC 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. In addition to the implementation, the team is required to prepare a report documenting their project which must include the E-R model, the DB design and its normalization. This eport is due before the demo and submitted to the lab instructors. The implementation of the system including the database system must be demonstrated by each team at a pre-designated 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 (a.k.a user friendly).

1. Develop an E/R diagram for the entity sets and relationships for SCC. 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.

The tasks

NOTE: If you are working in a team[3], it is recommended to proceed as follows. If you are doing it solo, you still have to meet all the deadlines and your system must meet all the requirements and expectations.

Each team is to implement the system. The team must have a designated leader who co:ordinates the work. It is expected that the team is to meet regularly during lab hours and each meeting is logged with the members participating.

Suggested protocol

- Get to know your team mates: make sure you respond to communications (emails text mail, phone calls, voice mail, ...) for meeting etc. Ignoring of such communications could be counted as a reduction in amount of participation and hence in the peer evaluation score!
- Make sure each member of the team is cognizant of his or her tasks.
- Do your share of the work
- Make alternate arrangements in case some of your team mates are "lost". You should be aware of circumstances where a team mate cannot participate fully due to crisis etc. However, there are those who are simply “free loaders”: making the distinction requires the wisdom of Solomon!
- Any ambiguities in this project description would have to be resolved. Some of these could be done via discussions with other teams and in the lab sessions. Many factors of the operating environment could influence the design. Some form of control and audit features have to be built into the system.

[3] Read the document “Working in groups”: see link in the FAQ section of the course page in CrsMgr

- Flesh out the applications; try to find details by talking with knowledgeable people. The web is a good resource: look at existing systems for inspiration -but do not 'pirate' these systems! Give all acknowledgements and references.
- Find the processing steps for the applications and the data required for such steps.
- Pin point the data generated by each such step and the application to process this data.
- Design the E-R model of your system and note the entities and relationships and their attributes.
- Convert the E-R model to the relational model.
- Identify the primary key of each relation and note the functional dependencies among its attributes.
- If necessary, convert the relation schemes to at least 3NF. Show this clearly in your report.
- Make sure the data in your tables is reasonably large and representative (as few as 5 to 10 tuples per table may be sufficient to cover most situations)
- Document all steps including an on-line log of work done(who, when, what, why).
- Document all communications. All emails must go to all members of the team and must be logged and included in the final report.

Report & submissions

Your final 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 Mbytes which means you need to upload a tar-gzipped ball (or rar, zip, etc.). This tar-gzipped ball must also include, LaTex or 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 (also in the README text file).
- 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 a 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 "README" 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 README 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 along with all updates to it should be uploaded with your report at demo time as part of your project report.

Note 2: The project is a group project and requires that each member of the group participate fully in its development, implementation, testing, preparing the report and demonstrating. It is also required that each member of the team evaluates the contribution of the other members of the team. If you do not do a peer evaluation, YOU get no credit for this part of the course! The draft version of the report must be given to the lab instructors before the demo date. There is a heavy penalty for late submission of the final report! No submission would be accepted once the penalty reaches or exceeds 100%.

The demonstrations of the project would as indicated above. See CrsMgr for the demo schedule to reserve a slot. Your draft report must be completed along with the project must before your demonstration day and time.

DO NOT FORGET TO DO A PEER EVALUATION before the deadline[4].

[4] REPEAT: If you do not do a peer evaluation, you get 0 for the project!