

## **Comp353/2 Fall 2020**

### **Condo-association Online Network System -- CON**

#### **DEMOS Week of Dec. 7, 2020**

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 CSS//HTML/JavaScript/ MySQL/ PHP. 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 CON 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 week starting 7<sup>th</sup> Dec. 2020. 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 Condo-association Online Network System (CON). 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 CON system enables members in a privately running server to share information and ideas. In addition, the system has a public web page that could be seen by anyone in read-only mode. Posting could be done by authenticated users of the system.

A condo-association board is a legal entity, made up of a number of elected condo-owners, and set up to oversee the operation and maintenance of a set of condos in a building(or buildings). The building(s) have a number of condos, each owned by an owner(s). An owner could own more than one condo; each condo could have one or more parking spaces and storage space associated with it. Each square meter of space has an associated fee which is determined by the condo-association.

The CON system administrator installs, for each condo-association, a user(and optionally an alternate user) with the role of the condo-association administrator. The condo-owners, for a condo-association, are ‘signed-up’ as members of the CON system by the condo-administrator. Each owner of a condo unit has a ‘percent share’ in various components of the condo unit some of which are private(the condo itself and any space such as parking and storage) and the condo-buildings public space (the hall ways and passages, the external structures, and facilities such as elevators, sports centre etc.). The CON system could manage one or more distinct condo associations.

CON provides services such as:

- Posting the current financial status of the condo association, budgets, ownership percent and current fees for the various condo-units and the historical record of the fees paid as well as special contributions(for major repairs etc.), maintenance work (rationales, dates, contractor, cost)
- Posting of all call for contracts and posting of all submissions to these calls
- Posting of contracts awarded and the status

- post notices of the administrator meetings, their agenda, and minutes of these meetings,
- notices of forth coming general meetings, agenda and resolution to be voted on
- polling for various resolutions
- election of members of the Condo-association board
- expressing condo owners concerns and discussions about them,
- provide facility for: on-line discussions, plan and share activities
- internal email system (all emails are stored in CONS)
- post 'classified' ads limited to the condo owners of a single condo-classification or public(accessible by members of other condo-associations or the general public)

Any on line discussions could be private to a limited number of condo-unit owners or accessible by all condo-unit owners from the same condo-association.

Thus only members of a given condo association can access material for that condo association. However, there could be an additional feature of public postings where members of different condo associations could have access: such as classified ads for items for sale, service availability or exchange. The CON system allows its members to create a profile, to create a list of other members with whom to share contents, and to view and add comments and contents within the system. Also, it allow members to hold discussions in forums.

## 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 members:* ID, password, other personal information such as name, address, internal email address. Members have family, friends and colleagues, privilege and status. A member can specify what part of his/her posting is public and what part is accessible to which members of his/her group or is private.

The administrator of a condo-association is responsible for keeping the list of condo-unit owners up to date. Any condo-unit owner who sells his condo is considered in-active but his information is maintained for a period of time. Any condo-unit owner can manage his/her information and delete any of it while being in good standing: except the financial obligations and the record of all transactions. The Condo administrator enters the details of a new condo-owner..

A member with a system administrator privilege has the full power on all services such as creation, deletion, editing of all members and groups. The CON administrator could also post notices to the condo-owners or public items (accessible to all users of CON and/or the general public). A member can post new items and specify which of his/her groups can access the post and who in each group can comment on it or add content to the post: such items could also be accessible to a wider audience.

CON system, by default, has one member with user-name admin and password admin created initially(Both of these must be changed after the first login). Only members with administrator privileges can add Condo-administrators. 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 a sent email box.

- ⊃ *Details about groups:* Records of information on each group, owner of the group, list of members belonging to the group. Owner of the group can add new members or remove members
- ⊃ Each user has a home page which has an index of his/her contents as well as the contents of each of the groups where he is an owner or member. The owner has a feature to view and manage permission to any content; permission can be changed only by the owner.
- ⊃ Detail on contents and the permission: 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.

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.

## **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 users.
- 4 User's request to join a group.
- 5 User's ability to withdraw from a group.
- 6 User's ability to post texts, images as well as to view posts by other members and comment on them.
- 7 Users can't post or view posts of only groups that they belong to.
- 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 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

Note: The system would be run by an administrator on a private server (could be on a cloud). There must be no facility to download any contents except through screen capture. Each team is to implement the CON 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 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 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 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