

# MCC Website Design

---

MCC is doing a long overdue renovation to the website -- pretty much starting from scratch, but we have been collecting content. As we discuss what functionality the site should have, it seems obvious to me that we need some CMS solutions – Drupal being the one that I am most familiar with.

Here is what we would need from our web designer:

1. **Site design.** We will soon have a new logo to base the design on. We need to integrate that color scheme and have a site that fits the goals of MCC to provide affordable housing, do community outreach, etc.
2. **Build site hierarchy based on CMS.** I think Drupal would work well, but you may recommend another. We have worked out the tabs and overall hierarchy, so we can give that to you.
3. **Set up a searchable database for documents.** I know there are a lot of free solutions out there. We need the ability to load various documents, such as organizational documents, meeting minutes, agendas, and such, that will come from different houses and committees. We want people to be able to upload and access these documents before, during, and after meetings. It might be nice to be able to load the documents on pages of particular committees and on announcement posts on the front page. We also need to have the ability to limit the availability of some documents (e.g. budgets and internal memos) to (certain) MCC users.
4. **Develop a database for houses and rooms.** One of the main purposes of the site is to let people know about room openings in houses. We need a database that would keep information (dimensions, price, photos, neighborhood, food policies, etc.) about each room of each house. Houses that have rooms available could post an opening that would link to the details of the room in the database, including when it would be available. People interested in joining a co-op could search for available rooms based on their preferences. The output of this database could be to have a widget that would show how many rooms are available for each co-op on the individual co-op pages (e.g. madisoncommunity.coop/avalon) and link to a page that gives the details of the room.
5. [still under consideration] **User IDs and listserv.** Establish a system so that all MCC members would have a personal login that would allow different users different permissions. This database would also be used for e-mail blasts. We may want the ability to add certain people to different committees with checkboxes that associate with different site permissions.
6. [optional] **Implement a bidding room for construction/repair projects.** Houses often need repairs, and this would allow us to solicit bids from contractors in town. There are a lot of solutions out there for silent auctions and other similar things. This is not an immediate needs, but if it were cheap and simple to set up, then it would be worth it to implement now. Here is an example of something that could work: <http://ruk.ca/content/how-run-a-silent-auction-using-drupal>
7. **\*\*Important:** There is a lot of turnover in the organization, so any website that is built needs to be very user friendly and accessible to people with limited experience with web development and with the organization.

## SQL Design [for geek eyes only]

*This is from an e-mail that James Littiebrandt sent about how to build the SQL for the room database/widget:*

The room database will be a single database with tables set up for each house or one large table, it doesn't matter which.

Main content will include the house, neighborhood, address, room size both dimensions and square footage, room rent, approximate utility cost, food cost, and open status.

The conversion to an sql database should be pretty easy given that we have an excel worksheet. But we'll want them to do it if possible so we don't have to play with a converter or manage csv files that might have data type mismatches during the conversion.

Front end widget.

Main Page/House page:

The Main page widget will display 1 small, 1 medium small, 1 medium, and 1 large room available from a random assignment of houses. Displayed information will be total cost of rent and approximate utilities, along with neighborhood and address. The neighborhood should be linked to search for other houses in that area.

I was thinking about this last night and we want to keep the main page clean, so I thought that 4 rooms would be good to give people a decent idea of the different costs associated with MCC. But this can be changed as seen fit.

The House page widget will act in a similar manner, but display 4 rooms according to size and availability. Also, if there is no medium small in the house, we do not want any returned value. There should also be a link in the widget to "list all available rooms" which will bring up a spreadsheet on a new page that displays all the database information.

Modification of the data

There will need to be a login page that accesses the update function. Updates from this page should be checklist for open rooms. There also needs to be a mass update function. Where one can select the type of data to modify, say small rooms, and then a secondary selection for the type of data to modify, like rent.

In addition to this, there needs to be a single room update function where a room can be selected and all information can be modified through text boxes. This should be separated from the main page to prevent accidents in modifying the wrong data. There is a single room, so only that room can be changed.

Searchability of the database

Search functions should include square footage, room rent, total rent, total rent plus food, and neighborhood. The first four should be ranges while the neighborhood should be a multiple selection.

This is all pretty easy stuff to handle in SQL, so there shouldn't be much difficulty. Everything here is simply a SELECT \* FROM function with AND/OR and UPDATE. So the biggest difficulty is in the conversion to get the data types right. This might be something that will have to be checked since numbers can be handled as strings as well as numbers if I remember correctly. We do not want to do that since it will screw up all our search functions. But that might also be my ruby training interfering so I'm not positive on that.