

How To Design a Formulize Application

July 12, 2009



This document has been prepared by Freeform Solutions. It is licensed under a Creative Commons license (attribution – non-commercial – no derivatives).

Introduction

Formulize is a piece of software that lets you create and configure data management tools. These tools are primarily forms, and you can configure the way those forms are related to each other, who can perform which actions on the forms and on the data submitted through them, and how that data is presented to users.

When you're finished doing that, you end up with, essentially, a new piece of software that people can use to manage whatever data your application is concerned with. This is very similar to the way an extensible content management system for a website lets you create and configure different types of content and features for a website. When you're done setting up your website, you have a new website to manage, rather than the content management software you started with (although it's still there behind the scenes, and you can use it to make changes to your website's structure if needed). When you're done setting up your Formulize application, it's like you have a new piece of software for people to interact with, but the Formulize toolkit is still there operating in the background, making the new software work.

When you set out to create an application with Formulize, you need to do a bit of design work, to figure out what pieces you need to create and how they should fit together. Once you've done that, you should have a tidy recipe for how to build your application.

This document describes the kinds of questions you need to ask, and the issues you need to think about, in order to create that recipe. You don't have to go through this process exactly in the order listed here, and often you will find the information you get at one stage will affect other parts of the design. So take a holistic approach to envisioning your application. This is definitely one of the times where you need to see the forest, and not just the trees.

Who Will Use Your Application?

The people who use your application will each have accounts, ie: a username and password, for logging in. Accounts are collected together into groups. Each account can belong to one or more groups. You have total control over what groupings of accounts you wish to have.

The groups are used to represent the logical collections of users that have some meaningful role in your application. You never refer to a user directly when you are configuring any part of a Formulize system. You only refer to groups. Then, all the users who have accounts that belong to that group inherit the configuration options that the group has received.

For example, if you are creating a system where volunteers log certain information, and then staff people review the logs, you should have a volunteer group and a staff group. If your organization is divided into different physical locations, then you may need to create one volunteer group and one staff group for each location. Whether you do or not depends on whether the fact there are different locations matters to the submission of logs and the review of logs; if all logs are considered together in your organization, and it doesn't matter what site a log comes from, then you probably don't need to have groups for each location.

The groups in Formulize are based on the requirements of the processes and workflows you're trying to model. They don't necessarily correspond to the company organizational chart, though hopefully the entities on the org chart do have some meaningful relationship with the processes and workflows that are actually carried out!

Groups serve three main purposes in Formulize:

You need groups for controlling permissions

The major reason you create groups of users, is so you can assign them different permissions (such as adding entries to forms, viewing entries, deleting entries they have created, or that other people have created, etc).

You need groups for controlling the visibility of information

You also need to have groups so you can control the "scope" of what information that people see.

For example, Formulize has built in controls that let you limit what people can see to only information from the same group(s) that they themselves are part of. In the logging example above, if there are different locations and each one should not have access to logs from the others, then you need to have groups corresponding to each location so that you can enforce the separation of the information according to group.

That doesn't prevent you from having more senior users who can see information from all groups. But if you want to be able to separate the information for some of the users, then you need to setup a group structure that will support separation.

You need groups so you can refer to certain users when configuring your application

Groups are also used to setup various configuration options in the forms that you make, such as which groups can see certain elements in a form and which cannot, or which groups are used to create a list of users that people can choose from, or even which groups should receive an e-mail notification when a new entry is created in a form.

You might need to create a group of users for just this purpose. For example, imagine you have some entries in a form, and a group of managers reviews them, but only certain managers can approve the entries after they have been reviewed. Any manager has permission to view the entries and edit them, but only certain managers can actually mark them as finally approved. In that case, you could create a check box on the form which, when checked, would indicate that the entry was approved. You could create a group of "approvers" and only certain managers would be members of that group, and then you would make that approvers group be the only group that could see the check box. There is a configuration option for each form question that lets you control which groups can see that question.

What Information Will Your Application Manage?

In a Formulize application, the basic way information is collected and stored, is through a form. The forms represent the different kinds of information that you need to manage. Each entry in a form represents an individual piece of information, such as one conference registration, or one survey response.

You manage different kinds of information

Sometimes, what looks like one form on paper is really two forms in Formulize; there might be two different kinds of information represented on one paper form. For example, imagine a loan application form that provided space for listing someone's current and previous jobs. On paper, the whole application might be one page. But electronically, you probably want to divide up that information into two parts: the main application, plus a series of "job" entries, so that you can easily work with the job information as a set. So in this situation you would probably create two forms in Formulize, a main application form and a job history form. You could tie them together into a single user interface later. The important thing is that the abstract kinds of information are handled separately.

Sometimes, there might be some information that could be separated out as its own kind, except you know that you don't ever want to do anything in particular with that information later. For example, you could perform followups after people fill in a survey. Each followup could be recorded in a separate followup form. Or there could just be questions related to the followup at the end of the survey itself. Those questions wouldn't appear when the survey was done the first time, but would be used later when it was time for the followup. If you don't ever need to work with followups by themselves, independent of the surveys that triggered them, then perhaps you don't need a separate form for that information.

Your groups interact with different kinds of information in different ways

Users can be related to different kinds of information in different ways. For example, a profile form is a kind of information where you expect only one entry per user. An activity log is a kind of information where you expect more than one entry per user. You need to identify what are all the ways of interacting with the information, and then select the appropriate options in Formulize when creating the forms and setting the permissions for various groups.

These interactions involve not only the creation of information, but the access rights for information. Some groups of users can see information that belongs to everyone in that same group. Some groups of users can see all information belonging to everyone. These facts about how people relate to the information need to be specified when you create the forms, and when you set the permissions for that form, for each group of users.

Your different kinds of information might be related to each other

Formulize allows you to relate forms to each other, for example, in the loan application mentioned above, there would be a "one to many" relationship between the loan application form and the job history form. The basis of the relationship would be that the application and the job entries would all have been created by the same person. You need to think about these relationships and you can specify them as part of setting up your application, using the "Frameworks" feature.

When you specify a relationship between forms, this allows you to do two important things later on, when people work with the forms and the information that has been submitted through them: first, relationships let you embed one form inside another form, ie: in the loan application and job history example above, the job history form could appear as if it were part of the loan application form, except the job history information would be managed separately by Formulize behind the scenes; second, relationships let you review related information as a single set of data when looking at lists of entries, ie: you could look at a list of loan applications, plus all their related job history information, as a single set of data, instead of having to look at two different lists.

You need to think about whether these are things you want to do with your information later on, and if so, then you should specify the relationships between the forms.

What Information Will Be Part Of Each Form?

Once you have identified all the kinds of information you will manage, and the way the users interact with the information, and how each kind is related to others, you need to focus on the forms. The forms themselves usually contain two or three different kinds of information:

Information for your main users to fill in

Most of the questions on the form will be filled in by the "main users", the primary audience for the form, such as employees who fill in an expense report form. So you need to identify what information you expect them to fill in. You can choose from all the form elements normally available on the web, from textboxes to dropdown lists to checkboxes. In general, the element types like checkboxes or dropdown lists are useful when you want to standardize the answers people give you, which makes statistical reporting possible later. Open ended elements like textboxes are good for custom information that you want from users.

If you are working with numeric information, make sure you set the data type appropriately, so Formulize knows that piece of information should be managed as numbers.

Information other users might need to fill in

Forms can contain lots of questions, and there might be two or more distinct groups of users who need to work with the form in different ways. For example, managers receiving the expense reports might have to check off an "approved" checkbox on the form. The main users of the form would obviously not have access to this checkbox!

These kinds of "office use only" questions need to be part of the form as well. You can set them to display only to the groups of users who should fill them in.

Information that reflects the "state" of an entry in a form

You may need some elements on the form that don't really *ask* for information at all. Instead, they might capture the state of this entry. For example, the "approved" checkbox on the expense report tells you something about what state that particular expense report is in. You might want to record even more detailed "state information", such as due dates or a rating that is calculated based on the information the main users submitted.

You can use the "derived value" feature to add dynamically calculated information to the form. You can also create elements in a form that you don't intend any user to interact with, and instead, Formulize itself will set the values of these elements as users work with your application.

For example, if you have a to-do list form, you might have a checkbox on the form that indicates if the task is done. But rather than letting users check off the box, it might be more effective if they can see a list of tasks, and simply click a "done" button beside each task when it is complete. Formulize can be configured to fill in the "done" checkbox automatically when the "done" button is clicked. If you want to use these kinds of automatic behaviours, your forms will need elements in them that Formulize can interact with.

How Will Users Interact With The Forms?

Formulize provides a default interface for working with lists of entries in forms, and for working with the forms themselves. For many simple applications, that is enough, and your application is now finished.

In other cases, you might want to create custom versions of the default interface, and Formulize provides a way for you to do that: the "screen" system. It lets you create as many variations of the default screens as you wish. Your users might expect your application to work in a different way from the default interface. They might want streamlined lists for certain purposes, rather than a generic list of entries. They might even need custom buttons and other special interface features.

You need different screens so users can perform different actions

Deciding what screens you need usually involves thinking about the activities the users are going to perform. Each discrete activity may require its own screen. Sometimes related activities could be handled on the same screen, as long as you provide multiple views of the entries. For example, in an equipment signout system, there are many activities that an administrative user would need to perform, such as seeing which pieces of equipment have been signed out and are awaiting pickup, and which pieces have been picked up and need to be returned. Those two activities could be carried out on two different screens, one that listed the items awaiting pickup, and one that listed items that have been picked up. Or those two activities could be carried out on one screen, but with two different "views" available from the "Current view" dropdown list.

It's up to you to decide which approach makes sense in your application. The decision often boils down to how similar the activities are that users will perform on each screen. Experiment with the "Save current view" feature on the default list of entries interface, since specific saved and published views can be used as the basis for a custom list of entries screen.

You need to think about how users will actually perform the activities related to each screen. For example, in the equipment signout system, when users are choosing which equipment they want to signout, they will presumably select items from a list. Do they do this by clicking the piece of equipment and selecting their name from a list of potential signers-outers? Or do they select the equipment in the list somehow, usually through the checkbox to the left side, and then click a "sign out" button? Think about how you want things to work, from the user's point of view.

Checklist

Here is an overview of the things you need to think about when designing your Formulize application:

Groups of users

- What is the purpose of each group?

Forms (kinds of information)

- How will each group relate to the forms?
- What permissions does each group need?
- How will the forms relate to each other?

Questions on the Forms

- What are the questions for users on the form?
- What are the "office use only" questions on the form?
- What are the "state information" elements in the form?

Screens (the interface for users)

- What are the different activities that users need to perform?
- What screens and saved views do you need to support these activities?
- Which activities will be part of which screens?
- What will users do on each screen to carry out each activity?

Getting More Help And Advice

You can post messages on the Formulize support forums on the Freeform Solutions website: www.freeformsolutions.ca/formulize. Other users, and Freeform Solutions staff, post there regularly with all kinds of questions and answers, from the very technical, to the very abstract, and everything in between. Come and share your experience with this growing open source community!