General Workflow Suggestions

Here are some points to consider when designing your application

How can your user edit data?

If, for example, your user enters the wrong village name or misspells a name, how will they correct that case data?

It is best to add an "edit details" form to your application. This can be a very simple process:

1. Make a copy of a registration form (or whichever form has most of the details you want to be able to update)
2. Change the name of the form, delete any questions you don't want to update, and add any additional questions
3. In the case configuration, pre-load all of the questions with the values from the case, and make sure that everything updates the case. By doing this, when the user opens the form all of the fields will be pre-populated with the previous values. They just need to change whichever value they want to correct, and all of the case properties will be updated with the most recent value.

Things to note about this:

- This process involves submitting a separate form to update the case data. So the original registration or other form will still be in the system, only the CASE DATA will be updated. As such, when analyzing data, if you only look at the registration forms but not the registration update forms, you will not be looking at an accurate data set
- If you have several case types in your application you may want an edit form for each case type

How will a user close a case?

In a single application there may be many ways to close a case. For example, from the user's perspective a case may be closed automatically when an event occurs, such as giving birth or recovering from an illness. However, sometimes it is important to account for edge cases where a case may have been registered twice or by accident.

There are a couple of ways to deal with this:

- You can add a "remove name" form to your application to deal with these situations- remember that using language like "close case" may not be clear to the user, as from their perspective you are enabling a way to remove the name from their list.
- You can add an edit form that allows the user to remove the name (i.e. "do you want to remove this user from your phone? Yes/No -> if yes, close the case)
- You can alter a case property which filters the case out of the user's case list. This can be a useful approach when you want someone other than the mobile user to have control over actually closing out a case. This solution is only recommended if you have case sharing enabled and therefore a way to actually close out the cases in a different application. Filtering the cases without a way to close them would lead to an build up of non-accessible cases on the phone over time. You could have a CloudCare app that can see all cases where a case property called case_to_close = "yes" and then the CloudCare user can deal with the case (i.e. determine if it is safe to close it, etc.)

How should I organize all my forms related to a single case type?

If you have a lot of forms related to one case type (i.e. registration, follow-up, home visit, counseling, edit/close, etc.) it may be useful to put them in more than one module with the same case type. Specifically, it is often useful to put registration forms in a separate module. This is because how the presence of a registration form affects the way in which the user accesses the forms.

If your module has a registration form in it, the workflow is: module -> form -> case list (if relevant) -> open form

If you module has NO registration form (all forms update or close a case), the workflow is: module -> case list -> form -> open form

The latter workflow is generally preferred because it allows for effective form filtering.

How can I reference the number of times a given form has been filled out?

You can do some creative work with hidden values to keep track of the number of times a given form has been completed by a specific user. Let's imagine you have a form that is used to take height and weight data every week, but every 10th week you also want to take some additional measurements. Or perhaps you want to show the number of times a form was filled out in the case details screen. Either way you will need hidden values structured like this example:

- prev_visit_count
  - Calculate condition: (blank)
  - Do not save to the case
  - Loads the value of visit_count
- visit_count
  - Calculate condition: if(#case/prev_visit_count = ", 1, #case/prev_visit_count +1)
  - Save to the case as: visit_count
  - Do not load anything into this hidden value

You can then display the value of visit_count or reference it in a display condition.

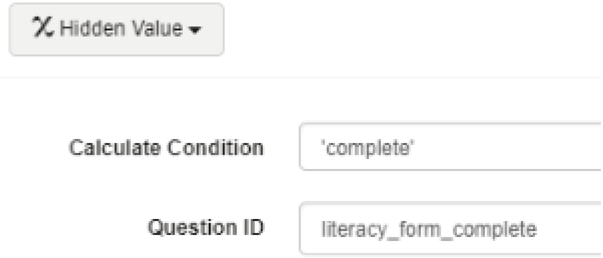
How can I make a form disappear from a form list once it has been filled out, or make forms appear in sequence?

Sometimes you only want a form to be filled out one time for a case. The definitive way to do this is to use case management and form filtering, such that the form is only available until it has been filled out once.

First, you have to make sure that the form is in a module that only has forms that require a case. This means that there cannot be any registration forms in the module. That way the case list will display before the form list. Let's take an example where you have a Literacy Test form that you only want to fill out once for each of the pregnant women that you are monitoring in your app.

Set up the form:

- In the Literacy Test form you will create a hidden value called something like literacy_form_complete
- Set the calculate condition to be contain some text or number- for example, we could set the calculate condition to "complete"
- In the form case management, save literacy_form_complete as a case property



Calculate Condition 'complete'

Question ID literacy_form_complete

Now you will need to set up a Form Display Condition for the Literacy Test form. Set the display condition so that the form only is displayed if literacy_form_complete DOES NOT equal complete. It would look like

./literacy_form_complete != 'complete'

For each new case the literacy_form_complete property will be empty, so the form will be displayed. But whenever the Literacy Form is filled out for a case, the case property value will change to "complete" and the form's display condition will never be met.

How can I make forms appear in sequence?

Sometimes there are a number of different forms for a case and you want only one to be filled out at a time. You can set up an application so that at first it only shows "Home Visit 1" and then after that so it only shows "Home Visit 2," etc.

The basic structure is very similar to the instructions above for making a form disappear, and again this will only work in a module where all forms require a case (this allows the case list to show up before the form list)

Set up your forms:

- In each form create a case property called next_form
- Set the calculate condition in each form to be the name of the subsequent form you want to display, or a number for that form. For example, in Home Visit 1 the value of next_form will be "2" and in Home Visit 2 the value of next_form will be "3"
- In the form case management make sure to save next_form as a case property in all forms

Set a Form Display Conditions for each form that references the value of next_form. The display condition for Home Visit 1 would be *./next_form = " or ./next_form = '1'* (at first the case property next_form will be empty, and that is when we want to show Home Visit 1) and for Home Visit 2 it would be ./next_form = '2'.

To complicate this workflow a bit, suppose you have a form in the list that should only appear if certain conditions are met. In that case, you can update the hidden value which updates the case property *next form* based on which form should appear next. For example, let's say you only want display home visit 2 if the home being visited has a front yard, if not you'll want to skip firstly to the form Home Visit 3. Your question (#form/front_yard) would be:

Does this home you're visiting have a front yard

a) 'yes'

b) 'no'

Then the hidden value which updates the case properties would then calculate to - #form/next_form = if(#case/front_yard = 'yes, '2', '3').

One thing to consider about these workflows is that they are very inflexible. So if for some reason a form needs to be filled out again, it is not going to be possible without having forms able to affect the next_form property in more complicated ways.

A solution to this is to have a 'reset forms' form which is always visible within the module. This form can contain just one single-select question and each choice should correspond to a form in the list. The value of each choice should correspond to the number of the form represented.

Eg - Which form would you like to reset to?

a) Home Visit 1

b) Home Visit 2

The value for choice *a* would be '1' and the value for choice *b* would be '2'. The answer to this question should then update the case property *next_form.*

How can I prevent a case property like ID from being the same for two cases?

Sometimes you have a field like "ID" which you want to prevent users from giving to two different cases. This is advanced and requires minor editing of the xform.

Step 1: Add the following instance to your form (NOTE: you only need to do this if your form is registering a new case, and do not need to do this if your form is modifying an existing case): <instance id="casedb" src="jr://instance/casedb"></instance>

Step 2: Use the following as an example of a validation condition: count(instance('casedb')/casedb/case[@case_type='MYCASETYPE'][case_property_for_id=#form/my_question_with_participant_id]) = 0

- For example, if your case type is "participant" and both the question ID and the property you want to check it against is "participant_id" the expression would be:

  count(instance('casedb')/casedb/case[@case_type='participant'][participant_id=#form/participant_id]) = 0

Notes:

- This will only check against open cases on the user's phone. If the user has not synced recently this validation may fail to prevent a duplicate. Moreover, this method will only effectively query for cases already loaded to the user's phone, and so will not query cases that are not assigned to that user or that user's case sharing group.