[Client] Partner Program - Solution Submission Form
User Definitions, Use Cases, and Process Flow Notes

User Definitions
- Partner: the person at the partner company responsible for submitting information about their solution.
- Admin: the [client] solutions partner representative.

User Tasks
Following are the common user tasks. Optional tasks are in italics.
- **Partner creates new solution**
  The partner enters the form and initiates the process of submitting a new solution.
- **Partner saves solution**
  The partner interrupts the process of inputting information in the form and saves it to be retrieved and finished later.
- **Partner retrieves solution**
  The partner retrieves a partially completed solution saved earlier (or a rejected solution) and resumes the process of inputting information in the form.
- **Partner submits solution**
  The partner submits a solution, having completed the form.
- **Admin notified of submission**
  The admin receives an email alert that a solution has been submitted.
- **Admin views solution**
  The admin views a submitted solution.
- **Admin approves solution**
  The admin approves a submitted solution for publication via the CMS
- **Admin communicates approval**
  The admin communicates to the partner that the solution is pending publication and relevant information on publication via email.
- **Admin rejects solution**
  The admin rejects a submitted solution.
- **Admin communicates rejection**
  The admin communicates to the partner that the solution requires additional editing via email and relevant information on required changes (via TBD).
- ***Admin deletes solutions?***
  The admin deletes duplicate or old solutions
- ***Admin inputs the solution into the CMS (outside of this system)***

Use Cases
There are two basic use cases, depending on the admin's review. The partner creates, may save and retrieves and submits a solution in both. In the first, the admin approves the solution. In the second, the admin rejects the solution and the process returns to the partner for re-submission and then approval. Optional tasks are in italics.

Use Case 1
The partner:
- creates a new solution
- *saves the uncompleted solution*
- *retrieves the uncompleted solution*
- submits the completed solution

[admin is notified of submission]

The admin:
- approves the solution
- notifies the partner of approval
- inputs the solution into the CMS (outside of this system)

Use Case 2
The partner:
- creates a new solution
- *saves the uncompleted solution*
- *retrieves the uncompleted solution*
- submits the completed solution

[admin is notified of submission]

The admin:
- rejects the solution
- notifies the partner of rejection

The partner:
- retrieves the rejected solution and makes required changes
- re-submits the completed solution

[the previous 4 actions may be repeated as necessary]

The admin:
- approves the solution
- notifies the partner of approval
- inputs the solution into the CMS (outside of this system)

Process Flow Notes

We are proposing two options on how the admin communicates approval or rejection with the partner. The descriptions below accompany the process flow charts and wireframes.

Form Solution (see process flow chart pp. 1-2)
In the first solution, the form includes a comments field at the top of each page so that, in case of rejection, the admin can request changes of the partner via the form on a per page basis. The comments fields would appear as editable fields on the admin's view and as text fields on the partner's view only after rejection.

1. The partner initiates the form to create a solution. The admin's comments field is hidden from the partner (see wireframes 1-5).

2-4. The partner progresses through pages 1-3 of the form. The partner can save the incomplete form at any point by clicking the SAVE button at the bottom of each page (see wireframes 1-5). The state of the saved solution is changed to SAVED. (all normal form error messaging and functionality)

5. The partner can return to an incomplete, saved form via their list of saved forms on their profile page (see wireframes 8-9).

6. When the form is complete, the partner submits the solution by clicking the SUBMIT button at the bottom of the 3rd page (see wireframe 5). The state of the solution is changed to PENDING or PENDING REVIEW (see wireframes 8-11), *depending on whether we're using 3 or 4 states.*

7. The system is triggered to send a notification email to the admin with at a minimum, the solution title, and ideally with a direct link to the solution or the admin's list.

8. The admin views the submitted solution via their list (see wireframes 10-11) or through a link directly to the solution (TBD by tech). The admin can view the partner's form data but cannot edit it. The admin can use the admin's comments field at the top of each page of the form to add comments (see wireframes 12-18).

9. While reviewing, the admin may add comments in the per page fields (if approved – TBD what happens to comments based on how data dealt with, if rejected – comments appear on per page basis on Partner's display)

10. The admin decides whether to approve or reject the submitted solution (see wireframe 17).

11. If the admin **approves** the solution, the state of the solution is changed to APPROVED (see wireframes 8-11).

- The system is triggered to send a notification email to the partner with information about approval and pending publication.
- The admin inputs the solution into the CMS (TBD if this is cut and paste from this view or the solution info is exported into some other location/format)

12. If the admin **rejects** the solution, the state of the solution either remains PENDING or is changed from PENDING REVIEW to PENDING EDITS (see wireframes 8-11), depending on whether we're using 3 or 4 states.

-The system is triggered to send a notification email to the partner with a request to return to the solution via their list of saved solutions or through a link directly to the solution (TBD by tech).

13. When the partner returns to the rejected solution, the admin's comments will be visible (see wireframes 19-21). They can make changes and re-submit (see #6 above)

Note on state (aka status)

In this approach, the pending state is complicated because a submitted solution is pending review by the admin and a rejected solution is pending edits by the partner. While the partner must be able to edit a rejected solution so they can re-submit it, they should not be able to edit one they have just submitted that is pending review (versioning issue).

The simplest approach is to have 3 states (see wireframes 9 & 11):

- **SAVED**
  A partially completed solution which the partner has saved and has not yet submitted. It is displayed in the list of saved solutions as a link to the form.
- **PENDING**
  A submitted solution not yet approved, pending review from the admin OR a rejected solution pending changes by partner. In both cases, it is displayed in the list of saved solutions as a link to the form, but partners could be instructed not to edit

solutions pending review, unless they have received a “rejection” email with instructions on what to change.

- **APPROVED**
  An approved solution, published or not. It is displayed in the list of saved solutions, perhaps for a predetermined timespan, but does not link to anything.

A more nuanced approach is to have 4 states (see wireframes 8 & 10):

- **SAVED**
  A partially completed solution which partner has saved and will retrieve and complete later. It is displayed in the list of saved solutions as a link to the form.
- **PENDING REVIEW**
  A submitted solution pending review from the admin. It is displayed in the list of saved solutions but does not link to anything.
- **PENDING EDITS**
  A rejected solution pending changes by the partner. It is displayed in the list of saved solutions as a link to the form.
- **APPROVED**
  An approved solution, published or not. It is displayed in the list of saved solutions, perhaps for a predetermined timespan, but does not link to anything.

Manual Solution (see process flow chart pp. 3-4)
In the second solution, the admin simply uses email to contact the partner in case of a rejected solution. This approach uses the 3 states (SAVED, PENDING, and APPROVED) described above.

1. The partner initiates the form to create a solution (see wireframes 1-5).

2-4. The partner progresses through pages 1-3 of the form. The partner can save the incomplete form at any point by clicking the SAVE button at the bottom of each page (see wireframes 1-5). The state of the saved solution is changed to SAVED.

5. The partner can return to an incomplete, saved form via their list of saved forms on their profile page (see wireframe 9).

6. When the form is complete, the partner submits the solution by clicking the SUBMIT button at the bottom of the 3rd page (see wireframe 5). The state of the solution is changed to PENDING (see wireframes 9 & 11). The system is triggered to send a notification email to the admin.

7. The admin views the submitted solution via their list (see wireframe 11). The admin can view the partner's form data but cannot edit it (see wireframe 18 for example).

8. The admin decides whether to approve the submitted solution or not (see wireframe 18).

9. If the admin approves the solution. The state of the solution is changed to APPROVED (see wireframes 9 & 11).

10. The system is triggered to send a notification email to the partner with information about approval and pending publication.

11. The admin inputs the solution into the CMS.

12. If the admin requires changes to the solution, s/he manually contacts the partner via email or phone to request additions or changes to the solution. The state of the solution remains PENDING until it is subsequently approved (see wireframes 9 & 11).