**IDEA - DETAILED DESIGN**

To do last as it may change: Need to create tables useridea\_hist, ideafinal\_hist & ideahist in Create…sql.

Read through entirely at least twice.

1. References:
   1. Current version of the Life cycle of Ideas & grant projects yymmdd.docx’.
   2. Current version of ‘Register and login detailed design yymmdd.docx’
   3. Current version of ‘Website standards yymmdd.docx’
2. **Task summary**
   1. Strengthening Future Families’ core business is making research grants. That is the primary purpose of the website. Researchers interested in receiving grants submit their research Ideas in a two-step process: preliminary Idea submissions and (for those selected) final Idea submissions.
   2. This task is to create two Flask Forms: “Share My Idea” for initial Ideas and “Complete My Idea” for final Ideas. In addition, at the bottom of both forms is a second input form named ‘addingusers.html’ containing a set of input boxes to associate users with the above Idea. Based on data entered in these boxes, five data elements populate tables ‘useridea’ or ‘userideafinal’ for each user associated with the project.
   3. An Idea progresses through a series of steps covering its creation, submission and review and is controlled by an Idea Status Code maintained on table ‘Idea’. Ultimately, there will be 19 codes covering the full life cycle of an Idea. All except the first three, however, are outside the scope of this project. Six tables contain Idea data at various times during the Idea’s life cycle. For more about these tables and their relationships, see the current versions of “Tables yymmdd.docx” and decision number six in “Design decisions yymmdd.docx”.
   4. The first three codes cover initial Idea creation and submission. These three codes and the two input forms are the subject of this task and are described below:
   5. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WORKING HERE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
3. **Creating and updating Ideas.**
   1. The “Share My Idea” data entry form (sharemyidea.html) is used to create and update Ideas. It is accessed at the bottom of login.html by clicking either of two types of links or, below them, a button. (See near the end of the last section of Ref 1.b. for a description of how the links are created).
      1. Types of links:
         1. Zero or more ‘ideanum’ links based on tables ‘userideatemp’ and ‘useridea in the first row,
         2. Zero or more ‘ideanum’ links based on table ‘userideafinal’ in the second row, and
         3. The ‘Create a New Idea’ button.
            1. Note: any logged in user can click this button to create an Idea.
      2. When any these links is clicked the user will already be authenticated using <emailaddress> as the user ID.
   2. Even though many columns on tables ‘ideatemp’ and ‘userideatemp’ are nullable, the edits requiring these data elements should be the same as for tables ‘idea’ and ‘useridea’ except the edits are warnings rather than fatal.
   3. A trigger is added to table ‘useridea’ requiring ‘anticipated\_contrib’ to be not null if ‘projrolecode’ <> (‘PI’ or ‘PI1’).
      1. A “Share My Idea” form edit constraint also enforces this so no records are rejected on insertion into the database.
   4. A trigger is added to table ‘userideatemp’ as a non-fatal warning requiring ‘anticipated\_contrib’ to be not null if ‘projrolecode’ <> (‘PI’ or ‘PI1’) if possible.
      1. A “Share My Idea” form edit constraint also enforces this so no records are rejected on insertion into the database.
   5. Two values are used to determine if this user is allowed to update this Idea:
      1. ‘open\_locked\_done’ indicates if SFF allows updates to this Idea or has closed it because SFF is reviewing or already has reviewed the Idea.
      2. ‘ideaupdate\_TF’ indicates if the Idea’s project team has allowed this specific user to update this Idea.
   6. At the bottom of the login.html page users may click any one of the following types of links to work on their Ideas:
      1. Note: if displayed, the user may also click role based links but that is discussed in a separate process design.
      2. If the user clicks the bottom ‘Create a New Idea’ button OR If the user clicks one of the ‘ideanum’ links in the top row above the button:
         1. If the user clicks the bottom ‘Create a New Idea’ button:
            1. The user’s <emailaddress> is saved to table ‘Ideafirst’. This creates a number for the Idea since by default, if ‘ideanum’ is null on insert, the database assigns the next available number to it.
            2. The number is permanently assigned and never reused.
            3. The ‘ideastatuscode’ is set to ‘ST’ – Started.
            4. Because the Idea is newly created:

‘open\_locked\_done’ = ‘open’

‘ideaupdate\_TF’ = ‘True’

* + - * 1. User <emailaddress> has both read and update authority since the user is the creator of the Idea.
        2. **ST** The Idea’s ‘Ideastatuscode’ is initialized as ‘ST’ meaning ‘Started’.
      1. If the user clicks one of the ‘ideanum’ links in the first row:
         1. The user’s access authority is determined:

Table ‘userideatemp’ is queried using <emailaddress> = <the currently logged in user’s ID> and ideanum = <the ideanum that was clicked> to return the data including ‘ideaupdate\_TF’.

If an Idea is found then the ‘open\_locked\_done’ value is ‘open’ since all indeas on this table have this ‘open\_locked\_done’ value.

If no Idea is found for this user and ‘ídeanum’ combination then the Idea must exist on table ‘Idea’ and the required two values are found as follows:

Table ‘useridea’ is queried to obtain all data including ‘ideaupdate\_TF’.

‘Ideastatuscode’ from table ‘Idea’ is used to access ‘**idea\_**open\_locked\_done’ on table ‘ideastatus’.

‘open\_locked\_done’ is set as the ‘**idea\_**open\_locked\_done’ value returned.

Table ‘idea’ is queried to return the data for the Idea.

The ‘ideaupdate\_TF’ and ‘open\_locked\_done’ values are interpreted as follows (same process as below so no need to recode. Instead can create a callable function for this):

If ‘ideaupdate\_TF’ is True, and

If the ‘open\_locked\_done’ value is 1 (open):

Then user <emailaddress> has both read and update authority.

No message is displayed.

Otherwise

The user has read only authority.

A message is displayed at the top of the screen, “The Idea cannot be updated because it is in processing or has been processed.”

If ‘ideaupdate\_TF’ on table ‘useridea’ is False, then

The user has read only authority.

No message is displayed.

* + - 1. The “Share My Idea” input form is opened and populated with either the returned ‘ideatemp’ and ‘userideatemp’ data elements or the ‘idea’ and ‘useridea’ data elements, whichever were returned.
      2. After any data element is updated on the “Share My Idea” input form,
         1. If the ‘ideastatus’ of the Idea on table ‘idea’ is ‘ST’ and no errors are detected in either the “Share My Idea” or ‘addingusers.html’ input forms then:

**UP** The system changes the ‘Ideastatus’ to ‘UP’ – ‘Updated’.

The Idea is saved to table ‘Idea’.

The Idea is deleted from table ‘ideatemp’.

* + 1. If the user clicks one of the ‘ideanum’ links in the second row:
       1. Using the user’s logged in <emailaddress> and the ‘Ideanum’ from the clicked link, the user’s access authority is determined:
          1. Since the Idea must exist on table ‘Idea’, the required two values are found as follows:

Table ‘useridea’ is queried to obtain all data including ‘ideaupdate\_TF’.

Table ‘idea’ is queried to obtain all data including ‘Ideastatuscode’.

‘Ideastatuscode’ is used to access table ‘ideastatus’ and return ‘**ideafinal\_**open\_locked\_done’.

‘open\_locked\_done’ is set as the ‘**ideafinal\_**open\_locked\_done’ value returned.

* + - * 1. The ‘ideaupdate\_TF’ and ‘open\_locked\_done’ values are interpreted using the same process as above so no need to recode. Instead can create a callable function for this
      1. The “Complete My Idea” input form is opened and populated with the data already retrieved from tables ‘ideafinal’ and ‘useridea’.
    1. Below both the “Share My Idea” and “Complete My Idea” input forms appears the ‘addingusers.html’ user input form via an includes statement. (See the Adding and Changing Project Participants section below).

1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WORKING HERE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. **Saving Ideas**
   1. The user may save an open Idea at any time.
   2. Preliminary Ideas, when saved, may be saved to any one of three tables:
      1. ‘Ideafirst’ if it does not yet have an ‘Ideanum’ to uniquely identify it.
      2. ‘ideatemp’ if it has an ‘ideanum’ and its ‘ideastatuscode’ is ‘ST’ meaning started.
      3. ‘idea’ in every other case.
      4. The following explains this in more detail.
   3. When saving the Idea:
      1. At creation it is automatically saved to table ‘ideafirst’.
         1. This table assigns it the next sequential ‘ideanum’.
      2. After that it is saved with ‘ideastatuscode’ = ‘ST’ (for started) to able ‘ideatemp’.
         1. The Idea is saved with any data elements the user may have populated, and
         2. Any data elements for users associated with the idea are saved to table ‘userideatemp’. (See the “Adding Project Participants” section below.)
         3. Both tables have fewer ‘not null’ constraints so the Idea can be saved precisely as the user created it, including with some but not all errors.
         4. When saving an Idea, the system verifies there is at least one person with update authority associated with it on table userideatemp when saved.
            1. (This assures the user did not revoke his/her own authority during the session).
            2. If there is no such user the Idea is not saved and an error is reported.
      3. For all other Ideas:
         1. If any errors are present for either table ‘idea’ or table ‘useridea’:
            1. An additional error is reported that the errors must be corrected before saving the Idea.
         2. If no errors are present for either table ‘idea’ and ‘useridea’, the Idea is saved to both tables.
   4. Additional notes on accessing and updating the Idea after creation:
      1. Whenever table ‘Idea’ records are inserted, *accessed*, or updated, now() is saved to ‘lastaccessed’ for the ‘idea’.
         1. Note that accessing an ‘Idea’, even if not updating it, is sufficient for purposes of updating ‘lastaccessed’.
      2. After an Idea has been saved once, any user associated with the Idea on table ‘useridea’ may read the idea.
3. **Deleting Ideas:**
   1. A “Delete Idea” button at the bottom of the “Share My Idea” data entry form, but above the Adding Project Participants section (see below) is provided for the user to delete the currently loaded Idea.
      1. When clicked, the system will respond with a Yes/No request to confirm the Idea should be deleted.
      2. If Yes is clicked, the Idea is deleted.
   2. In a future enhancement:
      1. Ideas will be deleted when they have not been accessed for at least (tentatively) two years.
      2. We may send an email warning that it will be deleted in (tentatively) 60 days.
      3. If so, this will create an additional Idea status code for an Idea where a warning has been sent.
   3. **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WORKING HERE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***
4. **Submitting Ideas:**
   1. A ‘Submit My Idea’ button will be displayed at the bottom of both the ‘Share My Idea’ (‘sharemyidea.html’) input page and the ideafinal.html input pages.
      1. The buttons submit the initial and final Ideas respectively.
      2. Before submission, initial Ideas will have an ‘Ideastatuscode’ = ‘UP’ and finals will have ‘Ideastatuscode’ = ‘C2’.
   2. Clicking the ‘Submit My Idea’ button.
      1. When the user is satisfied the initial or final Idea is finished, the user clicks the ‘Submit My Idea’ button.
      2. The system asks, “Are you certain you would like to submit this Idea?”
         1. ‘Yes’ and ‘No’ buttons are displayed immediately below the question.
         2. If ‘No’ is clicked, the user is returned to the open form.
         3. If ‘Yes’ is clicked, processing continues.
      3. The system verifies there are no errors present in the open form.
         1. If one or more errors are present:
            1. One or more error messages will already be displayed on the form.
            2. The system additionally responds, ‘One or more errors remain in the form. Please revise and resubmit.”
            3. The user is returned to the open form.
      4. The user’s login email address and the ideanum are used to verify the user has a ‘PI1’ ‘projrolecode’ on table ‘useridea’.
         1. If not:
            1. The user is returned to the open form, and
            2. The following error is displayed: “To submit an idea the user must be a Principal Investigator with submit authority.”
      5. If any user record associated with the Idea, has a lastaccessed timestamp > 2 1/2 years old:
         1. The user is returned to the open form.
         2. An error is reported indicating that to submit an idea, all associated user’s must have reviewed their personal data at least once in the past 2 ½ years and then listing the <emailaddress> of each such user.
      6. If ‘Ideastatuscode’ = <the value indicated> then the described process occurs:
         1. ‘ST’
            1. The user is returned to the open form.
            2. The following message id displayed: “The Idea contains one or more errors. Please revise and retry.”
         2. ‘S1’ or ‘S2’
            1. The user is returned to the open form.
            2. The following message id displayed: “This Idea has already been submitted.”
         3. ‘Ideastatuscode’ = <any other value except ‘UP’ or ‘C2’>
            1. The user is returned to the open form.
            2. The following message id displayed: “The Idea cannot be submitted because it is in processing or has been processed.”
         4. ‘UP’ or ‘C2’
            1. Now() is saved to variable ‘current\_timestamp’
            2. THE FOLLOWING ARE PROCESSED AS A SINGLE TRANSACTION:

‘Ideastatuscode’

**S1** If the Idea’s ‘Ideastatuscode’ is ‘UP’ it is updated to ‘S1’.

**S2** If the Idea’s ‘Ideastatuscode’ is ‘C2’ it is updated to ‘S2’.

‘current\_timestamp’ is saved to ‘S1\_initsub\_time’ or ‘S2\_initsub\_time’ on table ‘idea’.

A copy of the row on table ‘Idea’ is copied to table ‘Ideahist’. ‘current\_timestamp’ is copied to ‘savedtime’ in table ideahist’s primary key.

All the rows on table ‘useridea’ for this Idea are read. For each row the user’s row on table ‘user’ is copied to table ‘userhist’ and ‘current\_timestamp’ is copied to ‘savedtime’ in table userhist’s primary key. It is essential that ‘current\_timestamp’ is recorded so the record’s key successfully matches future searches.

Each row on table ‘useridea’ for this idea is saved to table ‘useridea\_hist’ and ‘current\_timestamp’ is used to update ‘savedtime’ on table ‘useridea\_hist’.

If ‘Ideastatuscode’ at the start of the transaction was ‘C2’, a copy of the row on table ‘Ideafinal’ is copied to table ‘Ideafinalhist’. ‘current\_timestamp’ is copied to ‘savedtime’ in table ideafinalhist’s primary key.

END OF THE TRANSACTION.

* + 1. The system responds, “Your Idea has been received. Thank you.”
    2. An automated message is created and inserted into table ‘message’:
       1. ‘msgdate’ will be ‘current\_timestamp’.
       2. ‘subjectcode’ will be ‘grantirr’.
       3. ‘visname’ will be ‘SFF System’.
       4. ‘visemail’ will be ‘system@strongfam.org’
       5. ‘message’ will be ‘A grant request first (or final) submission was received <msgdate>’.
  1. **E1** The E1 confirmation email is sent thanking the PI1s for submitting the idea. Currently done manually - future enhancement.

1. **Adding and Changing Project Participants**
   1. Need to extend this process to include both table useridea & ideatemp.
   2. At the bottom of the Share My Idea input form appears ‘addingusers.html’ via an includes statement. ‘Addingusers.html’ looks as follows:
      1. It will adhere to SFF website requirements as described in Ref 1.c.
      2. Title: “Adding and Changing Project Participants”.
      3. Adding one or more persons can improve an Idea, if the persons bring useful experiences, abilities, or resources. This is particularly true for participants that may help connect with state departments of education or state or local boards of education.
      4. To include an individual in an Idea, those persons must first register on this website. Once registered, participants can be added to the Idea here. During registration, participants can enter up to 250 words about themselves, provide a link to their CV, or both.
      5. Participants may be added to Ideas once the Idea has been successfully saved one time with no errors.
   3. Project Participant boxes. Under the above instructions are input boxes.
      1. The boxes perform two functions:
         1. To list the participants already associated with the open Idea, and
         2. To Add or Delete participants.
      2. Editing
         1. All website standards are followed for these input boxes as described in Ref 1.c. above including:
            1. Each box that contains an error will display a red border.
            2. Box contents are fully edited immediately upon entry.
            3. Errors are displayed for the boxes one at a time in red letters below each box.
      3. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WORKING HERE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
      4. The boxes will be displayed in two sections, one above the other.
         1. First Section.
            1. Subtitle above: Current Project Members. Aligned with the left edge of the left box.
            2. The boxes will be in the form of a matrix 4 columns wide by as many down the page as needed to display the information.
            3. The first section will display the current contents of table ‘useridea’ for all users listed for this Idea.
            4. Each form row will display one row from the table.
            5. Each row will contain 4 data elements (see below for more about each of these data elements):

Email address

Project role code

Idea update

Anticipated contribution.

* + - 1. The two sections will be separated by a little extra whitespace.
      2. Second Section
         1. Subtitle: Add or Delete Aligned with the heading for the First Section.
         2. Is used to Add and Delete users from table ‘useridea’.
         3. Is displayed below the first section.
         4. Will contain only one row.
         5. Will be five boxes wide.
         6. Each column in the second section will be precisely aligned below the similarly named column in the first section.
         7. An additional “Add/Del” box will be provided a little to the left of the four columns that are aligned with the First Section.

The user may enter either ‘A’ or ‘D’ (not case sensitive) to add or delete a user.

* + 1. Processing adds and deletes.
       1. If a ‘D’ is entered
          1. The user enters the <emailaddress> of the user to be deleted.
          2. the three remaining boxes for this user are grayed out.
          3. Instructions above this row tell the user, when deleting a person, to press <enter> to proceed.
          4. When <enter> is pressed

The system prompts, “Are you sure you want to delete this user?”.

‘Yes’ and ‘No’ buttons are displayed below the row.

If yes is pressed:

The system checks to see if the user to be deleted is the current user. If it is, the system responds, “The user to be deleted is you. If you do this, the currently open Idea will be saved, this window will be closed and you will no longer be able to access this Idea. Are you sure you want to delete this user?

‘Yes’ and ‘No’ buttons are displayed below the row.

If ‘No’ is pressed: the open reminder is closed.

If ‘Yes’ is pressed: processing continues

The system verifies this is not the only user on table ‘useridea’ with ‘ideaupdate\_TF’ = TRUE.

If it is the only user, an error is returned explaining, “Cannot delete. Every Idea must have at least one user with authority to update the idea.”

If this is not the only user on table useride with ‘ideaupdate\_TF’ = TRUE, the user is deleted from table ‘useridea’, and

If the user selected yes for the previous question, the events described in the response warning for the question are executed.

* + - 1. If an ‘A’ is entered
         1. The other four boxes must be populated.
         2. If no errors are identified in the input row the user may add this Project Participant
      2. A button or other method is provided nearby to execute the adds and deletes.
      3. Immediately upon execution, the top matrix is updated to show the new current list of Project Participants.
    1. The user can associate an unlimited number of persons with an Idea or project.
  1. Project Participant columns:
     1. ‘Ideanum’ is not included in the Project Participants boxes but is required to enter the data into table ‘useridea’. The system uses as a default the value of the Ideanum of the Idea currently open.
     2. Specific columns:
        1. **Email address.** Should be the email address the person used to register. If the user is not found on table User an error is returned like, “User not yet registered”.
        2. **Project role.** Displays a pull down menu listing the projrolecode from table projrole. The user must select one and only one.
        3. **Idea update.** Is a Boolean Yes or No. Allows the person completing the form to grant or revoke the user’s ability to update the idea. The form will validate there is always at least one user with authority to update each Idea on useridea. When the Idea is first created this defaults to the user creating the idea.
        4. **Anticipated contribution**. Accommodates a brief description (max: 120 characters or about 20 words) of the person’s anticipated contribution to the project.

1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WORKING HERE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. **Adding & removing documents from a (initial or final) submission.**
3. **Send Documents**