This document summarizes the overall design and requirements for the header functionality requested by Heather.

The design aims to provide a flexible way for users to create header-like functionality for their data collection needs. It will function similarly to “Session” for Heather’s Field Day, or the header section for the bird data collection example she gave us.

To allow flexibility in the mind of the user, header-like objects will be referred to as “Metas” in Flex, coming from the definition “self-referential, self-reflective, or self-aware. denoting something of a higher or second-order kind.”

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 1.0:** Data View

* “Data View” is selected – top left.
* “Open Meta” is analogous to starting a new session in the old mobile application.
* The user must open a new Meta before clicking “New Entry” if and only if the tab is assigned to a Meta (tabs don’t *have* to be assigned to Metas).
* Although not shown here, if and only if any data entries in a given tab are associated with a Meta, there must be a “Meta Date” column on the left that displays the date and time of the entry’s associated Meta. If some entries have a meta and some do not, the ones that do not display “No Associated Meta”.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 1.1:** Data View

* “Open Meta” was clicked.
* “Open Meta” changed to “Close Meta”
* “New Entry” can now be clicked.
* “Assign Meta” can’t be clicked while a Meta is open.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 2.0:** Open Meta

* “Open Meta” button was clicked.
* With “New Meta” selected, clicking “Continue” will close this window and return to the data view.
* “Existing Meta” is analogous to reopening a session in the old mobile application.
* With “Existing Meta” selected, clicking “Continue” will close this window and open the “Meta Selector” window pictured in figure 2.1 below.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 2.1:** Meta Selector

* **Disclaimer:** The values in these tables aren’t consistent with the UI examples. I didn’t want to take the time to edit the contents.
* Displays past Metas that are associated with the current tab, in this example the “Lizards” tab. Notice there are three scroll bars.
* There can be more than one Meta associated with each tab. In this example, Session meta-data and lizard meta-data.
* Each instance of Meta will be associated with instances of other Metas that are opened at the same time. For example, “Lizards” is associated with Session **and** Lizard-Meta Metas. When “Open Meta” is clicked and a new Meta created, an instance of **both** Lizard-Meta and Session Meta will be created and will be associated with each other through the same date of opening. Therefore, selecting the first option in “Session” would have the same result as selecting the first option in “Lizard-Meta”.
* Pressing “Continue” with a past meta selected will add entries with associations to all associated meta instances (analogous to reopening a session in the old mobile application).

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 3.0:** Assign Meta

* Clicking “Assign Meta” while in data view opens this window.
* Changes are reflected in the database upon clicking “Assign Metas”.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4.0:** Meta View

* This view displays a different set of tabs for the given project.
* Each tab is associated with a Meta, with their respective entries corresponding to Meta instances.
* Meta instances are associated with one-to-many “Data View” entries.
* Pictured here is a “Session” Meta and a “Lizard Meta” Meta.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4.1:** New Meta

* “Show meta on:” allows the user to have control over when and how often a meta needs filled in.
* On “Opening Meta” would result in the Meta form displaying immediately after the initial opening of a Meta (not loading a previous one). This would allow functionality similar to “Session” in the old Field Day mobile.
* “Before Data Entry” or “After Data Entry” would display the Meta each time there is an entry. This might be useful for adding up tallies as the user enters data.
* On “Closing Meta” would result in the Meta form displaying immediately after closing a Meta.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4.2:** Manage Meta

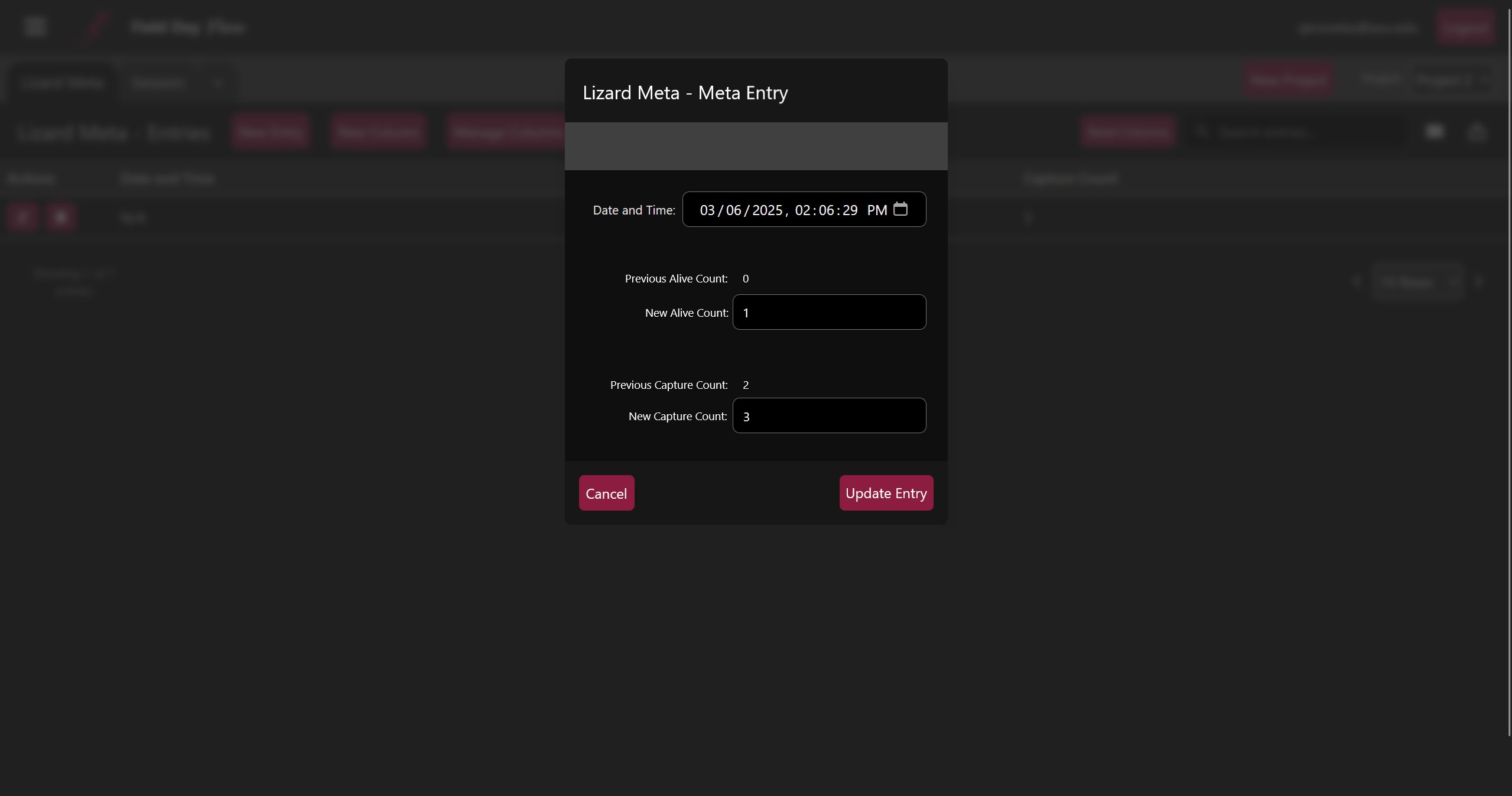
* We need to make sure that any changes to one associated element are propagated to all other associated elements. Not only for this function but all modifying functions.
* Pretty straight-forward.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4.3:** Meta Entry

* This is an example of a “Show Meta on Opening” that displays immediately after clicking “Open Meta” so that the Session data can be recorded before any entries are recorded.
* All lizard entries are associated with this Session entry from now until the Meta is closed. The Meta can be found in:   
  “Meta View” -> “Session” Tab.
* This Meta instance will be associated with the “Lizard Meta” Meta instance, and vise-versa.
* This Meta instance will be associated with all “Data View” entries from this Meta session and vise-versa.



**Figure 4.4:** Meta Entry

* This is an example of a “Show Meta After Data Entry” that displays immediately after each submission of a lizard entry, which enables the user to collect Meta data about the lizards themselves.
* This is not multiple “Lizard Meta” instances. It is an update of the instance created at the opening of the Meta.
* Date should not be editable here.
* All lizard entries are associated with this “Lizard Meta” entry from now until the Meta is closed.  
  The Meta can be found in: “Meta View” -> “Lizard Meta” Tab.
* This Meta instance will be associated with the “Session” Meta instance, and vise-versa.
* This Meta instance will be associated with all “Data View” entries from this Meta session, and vice-versa.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4.5:** Delete Meta

* We need to make sure that any changes to one associated element are propagated to all other associated elements. Not only for this function but all modifying functions.
* This window is pretty straight-forward.

A diagram of a computer code

AI-generated content may be incorrect.

**Figure 5.0:** Database Structure

* This is tentative. Although it is likely very close to the overall database architecture, feel free to add fields to documents if you need in order to implement the requirements.
* Fields with “+” indicate associations with other documents. This is to clarify possible field name confusion.