Fragalysis Changes WP1- Action Capture PoC

*(For discussion)*

# First Principles

1. Currently a researcher using Fragalysis is able to create a session (overall project) using a Target set of data, do investigative actions and then (optionally) save one or many snapshots of the current state of the screen. These snapshots can be shared with other Chemists.
2. What is missing is that the Chemists’ *actions* used to reach the snapshot (at the point it is saved) are not recorded. These are important as a record of the workflow used to reach a decision.
3. Furthermore, a record of the Chemists’ actions are important *during the research itself*. The Chemist may wish to undo/redo an action.
4. For the case of the PoC at least, an action in the case of fragalysis can broadly be thought of as a screen action - selecting a dataset / pushing a button to view aspects of it / select filters and so on (complete set in appendix).

# Proof of Concept Solution:

There seems to be two places here where actions might be recorded:

1. On an *ongoing* basis for research at project level - allowing the Chemist to undo/redo actions.
2. As a documented history leading to a decision/idea (e.g. saved with a snapshot). In this case the user might want to edit/filter a set of actions before saving them.

The actions to be recorded could also be configurable to limit which ones would be visible to the user.

**Initial LImitations**

To speed up development of the PoC, I suggest that we initially limit functionality - perhaps only doing project level actions. Once this is up and running, any extra functionality can be discussed.

# 

# Use Cases (Main):

Items marked in red are important for the overall front end/ back end design.

### Project Level:

|  |  |
| --- | --- |
| UC | Description |
| 1 | **New project (Logged In):**  Select Target  *Create session\_project to record actions against (1)*  Get current list of action types  Start recording of actions  *Ability to undo/redo last/latest action (2)*  *Press Save button (3)* - most recent version of actions stored as JSON field in postgres (session actions) alongside session project at end of session. |
| 2 | **Existing project (Logged In):**  Select project  Get last saved version of actions for project  Show current state  Get current list of action types  *Ability to last/latest action (2)*  *Press Save button (3)* - Most recent version of actions stored as JSON field in postgres (session actions) alongside session project at end of session. |
| 3 | **New project (Anonymous User):**  Select Target  *Create session\_project to record actions against (4)*  Get current list of action types  Start recording of actions  *Ability to undo/redo last/latest action (2)*  *Press Save button (3)* - most recent version of actions stored as JSON field in postgres (session actions) alongside session project at end of session. |
| 4 | **Existing project (Anonymous User):**  Select project (4)  Get last saved version of actions for project  Show current state  ... |

**Notes/Questions (from notes in red above):**

1. *Create session\_project (logged in):* Project is created as soon as a target is selected to provide a starting point for recording actions. Currently it is only created when the save button is pressed.
2. *Ability to undo/redo:* It should(?!) be relatively straightforward to add an undo/redo button for actions. Adding more complex editing of historical actions may be more complicated due to issues like the grandfather paradox. However it might be possible to operate, for example, on a complete series of linked actions. We might also consider hiding actions rather than removing them.
3. *Press Save button:* Fundamental to the design is deciding at what point actions will be saved. One efficient option is to only save actions if the user presses a Save button. Other options might include auto-saving actions or, if the user tries to navigate away from the window, asking whether they would like the current state to be saved or not.
4. *Create session\_project (anonymous):* To allow anonymous users to record actions, projects would also need to be created on selection of a target. At the moment these do not appear on the landing page, but possibly we could create a cookie so that the user could drop back into the existing project. This will lead to more projects being created so might we have to consider maintenance of old projects?

### Snapshot Level:

|  |  |
| --- | --- |
| UC | Description |
| 5 | **New Snapshot (Logged In):**  On Save Snapshot window for project:  *Show last saved version of actions for project (5)*  *Allow actions to be edited (6)*  Save snapshot of actions alongside snapshot. |
| 6 | **Existing Snapshot (Logged In):**  Get Existing snapshot for project:  Get last saved version of actions for snapshot  Start recording of actions  On Save Snapshot window:  *Show last saved version of actions for snapshot (7)*  *Allow actions to be edited (6)*  Show last saved version of actions for project  Save snapshot of actions alongside snapshot. |
| 7 | **New Snapshot (Anonymous):**  On Save Snapshot window:  *Show last saved version of actions for project (5)*  *Allow actions to be edited (6)*  *Save snapshot of actions alongside snapshot (8)*. |
| 8 | ***Existing Snapshot (Anonymous): (8)*** |

**Notes/Questions (from notes in red above):**

1. *Show last saved version of actions for project:* There were some discussions here about whether base state should be from the previous saved snapshot or the whole project. As research is not necessarily a linear action it seems to make more sense that the starting point for each snapshot (idea) is the whole project.
2. *Allow actions to be edited:* Actions linked to a snapshot (seem to) have two main purposes: 1. To act as a record of decisions made to reach a decision/idea. 2. To enable the snapshot to be recreated. One method may be to store all actions, but allow the actions to be edited so that some can be made not visible (rather than being deleted) - and maybe a button added to show only important actions?
3. *Show last saved version of actions for snapshot:* When a previous snapshot is selected, the updated saved actions should be shown.
4. *Save snapshot of actions alongside snapshot:* For an anonymous user this is currently only accessible via a URL (which links to the snapshot alone rather than to the actions). Maybe we need to revisit this either via cookie or anonymous projects list?

# 

# Tables/API Changes:

### Action\_types

**Used to filter / display action types visible to user (and to allow flexibility in database)**

GET ../api/action-types - Returns all active action types as a JSON string (id, description).

GET ../api/action-types/?id=<id> - Return a single action type.

### Session\_actions

**A record of actions the user has performed since the start of the project.**

In Session\_actions / Snapshot\_actions, action data will be saved as a big JSON field to maximise flexibility/minimise number of calls.

GET ../api/session-actions/?session\_project=

POST ../api/session-actions/{session\_project}/

PUT ../api/session-actions/{session\_project}/{session\_action\_id}/

DELETE ../api/session-actions/{session\_project}/{session\_action\_id}/

### Snapshot\_actions

**A snapshot of actions from the start of the project to the desired state of the snapshot.**

GET ../api/snapshot-actions/?snapshot=

POST ../api/snapshot-actions/{snapshot}/

PUT ../api/snapshot-actions/{snapshot\_action\_id}/.

DELETE ../api/snapshot-actions/{snapshot\_action\_id}/

# Action Types:

Tracked shall be following actions:

1. Loaded target
2. Site turned on
3. Ligand turned on for a molecule
4. Sidechains turned on for a molecule
5. Interactions turned on for a molecule
6. Surface turned on for a molecule
7. Vectors turned on for a molecule
8. Vector selected in NGL view
9. Molecule added to shopping cart
10. Compound from custom dataset is selected
11. Ligand turned on for a compound from custom dataset
12. Sidechains turned on for a compound from custom dataset
13. Interactions turned on for a compound from custom dataset
14. Surface turned on for a compound from custom dataset
15. Ligand turned on for a inspiration of a compound
16. Sidechains turned on for a inspiration of a compound
17. Interactions turned on for a inspiration of a compound
18. Surface turned on for a inspiration of a compound
19. Vectors turned on for a inspiration of a compound
20. Ligand turned on for a inspiration of a compound
21. Sidechains turned on for a inspiration of a compound
22. Interactions turned on for a inspiration of a compound
23. Surface turned on for a inspiration of a compound