THE UNIVERSITY of ADELAIDE

Snapshot Week 9 of Group InfluxUI-PG01

No-Code Solution for InfluxDB

by

Xiaoyue Rao a1819070

Jianghao Jin a1880849

Tiantian Wang a1894037

Manhong Chen a1904387

Ziyan Zhao a1883303

Ling Luo a1847162

Yufei Wang a1897360

Yinkai Yuan a1909218

1. Product Backlog and Task Board

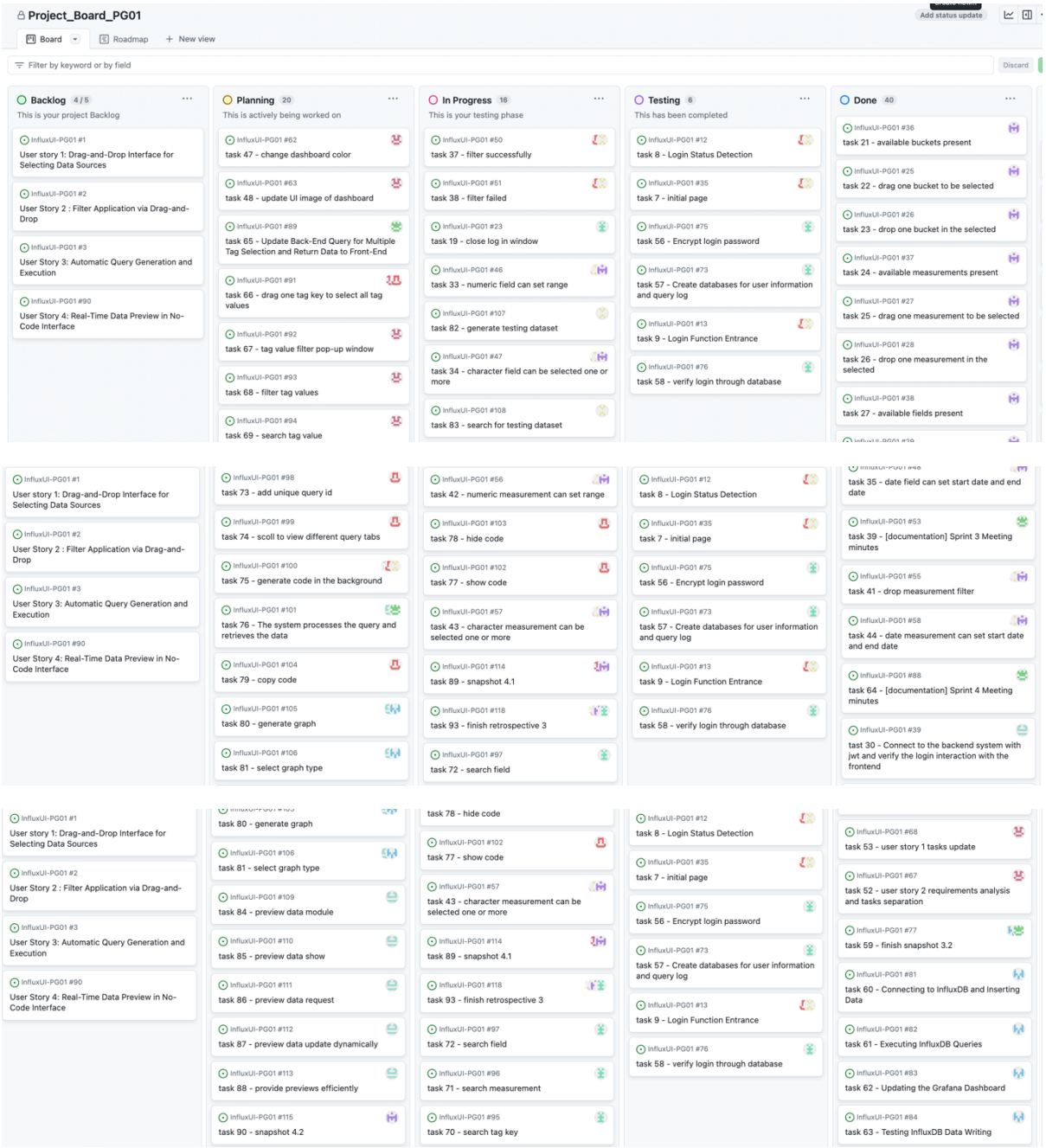
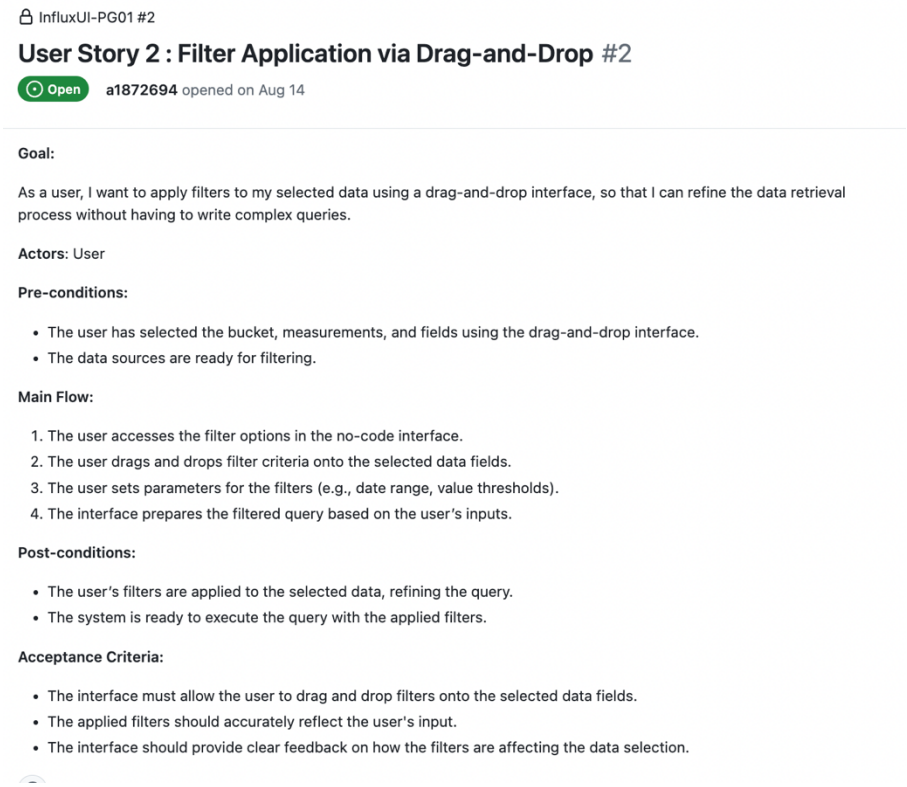


Fig.1 The Backlog of Project

2. Sprint Backlog and User Stories



InfluxUI-PG01 #2

User Story 2 : Filter Application via Drag-and-Drop #2

Open **a1872694** opened on Aug 14

Goal:

As a user, I want to apply filters to my selected data using a drag-and-drop interface, so that I can refine the data retrieval process without having to write complex queries.

Actors: User

Pre-conditions:

- The user has selected the bucket, measurements, and fields using the drag-and-drop interface.
- The data sources are ready for filtering.

Main Flow:

1. The user accesses the filter options in the no-code interface.
2. The user drags and drops filter criteria onto the selected data fields.
3. The user sets parameters for the filters (e.g., date range, value thresholds).
4. The interface prepares the filtered query based on the user's inputs.

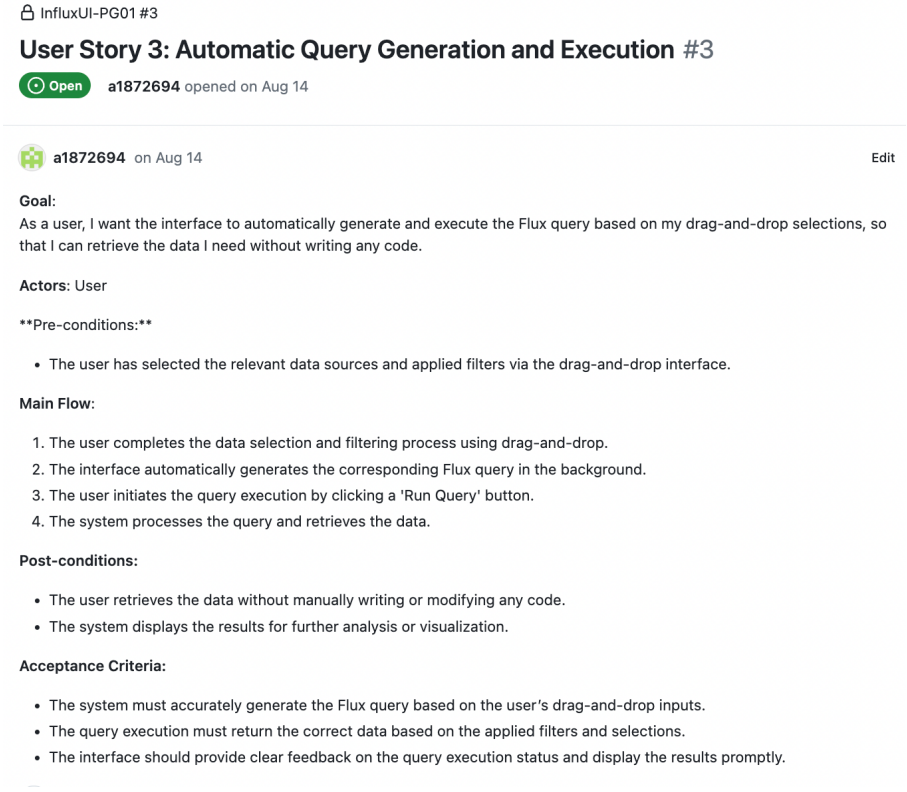
Post-conditions:

- The user's filters are applied to the selected data, refining the query.
- The system is ready to execute the query with the applied filters.

Acceptance Criteria:

- The interface must allow the user to drag and drop filters onto the selected data fields.
- The applied filters should accurately reflect the user's input.
- The interface should provide clear feedback on how the filters are affecting the data selection.

Fig. 2 User Story 2



InfluxUI-PG01 #3

User Story 3: Automatic Query Generation and Execution #3

Open **a1872694** opened on Aug 14

a1872694 on Aug 14 Edit

Goal:
As a user, I want the interface to automatically generate and execute the Flux query based on my drag-and-drop selections, so that I can retrieve the data I need without writing any code.

Actors: User

Pre-conditions:

- The user has selected the relevant data sources and applied filters via the drag-and-drop interface.

Main Flow:

1. The user completes the data selection and filtering process using drag-and-drop.
2. The interface automatically generates the corresponding Flux query in the background.
3. The user initiates the query execution by clicking a 'Run Query' button.
4. The system processes the query and retrieves the data.

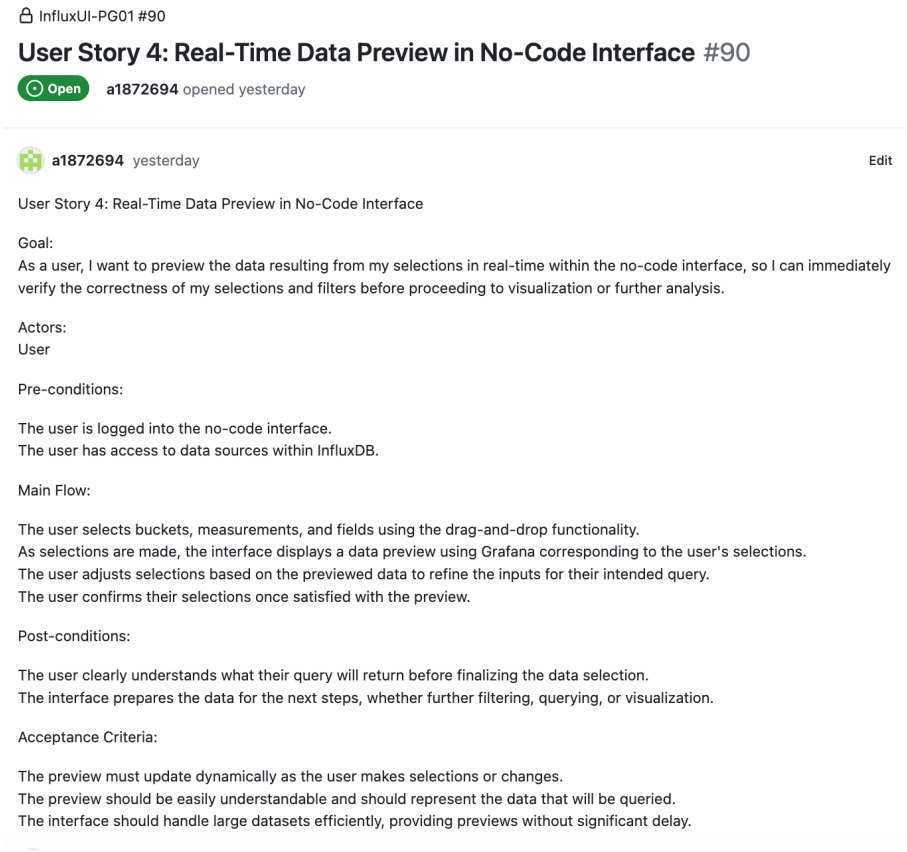
Post-conditions:

- The user retrieves the data without manually writing or modifying any code.
- The system displays the results for further analysis or visualization.

Acceptance Criteria:

- The system must accurately generate the Flux query based on the user's drag-and-drop inputs.
- The query execution must return the correct data based on the applied filters and selections.
- The interface should provide clear feedback on the query execution status and display the results promptly.

Fig. 3 User Story 3


InfluxUI-PG01 #90

User Story 4: Real-Time Data Preview in No-Code Interface #90

Open **a1872694** opened yesterday

a1872694 yesterday Edit

User Story 4: Real-Time Data Preview in No-Code Interface

Goal:
As a user, I want to preview the data resulting from my selections in real-time within the no-code interface, so I can immediately verify the correctness of my selections and filters before proceeding to visualization or further analysis.

Actors:
User

Pre-conditions:

The user is logged into the no-code interface.
The user has access to data sources within InfluxDB.

Main Flow:

The user selects buckets, measurements, and fields using the drag-and-drop functionality.
As selections are made, the interface displays a data preview using Grafana corresponding to the user's selections.
The user adjusts selections based on the previewed data to refine the inputs for their intended query.
The user confirms their selections once satisfied with the preview.

Post-conditions:

The user clearly understands what their query will return before finalizing the data selection.
The interface prepares the data for the next steps, whether further filtering, querying, or visualization.

Acceptance Criteria:

The preview must update dynamically as the user makes selections or changes.
The preview should be easily understandable and should represent the data that will be queried.
The interface should handle large datasets efficiently, providing previews without significant delay.

Fig. 4 User Story 4

2.1 Brief description of the user stories selected

In this sprint, our group focuses on user story 2, user story 3 and user story 4. As a user in user story 2, they want to apply filters to their selected data using a drag-and-drop interface, so that they can refine the data retrieval process without having to write complex queries as a user. As a user in user story 3, they want the interface to automatically generate and execute the Flux query based on their drag-and-drop selections, so that they can retrieve the data they need without writing any code. Finally, in user story 4, as a user, they want to preview the data resulting from their selections in real-time within the no-code interface, so they can immediately verify the correctness of their selections and filters before proceeding to visualization or further analysis.

With the discussion in sprint 3 review meeting and sprint 4 planning meeting, we found that the understanding between client and our group is a little bit different, and there are more details about the filtering function shown to us, so we decide to update what we

have developed for filtering to meet the client's requirements. In addition, to keep up with our development schedule, we would like to work on user story 3 and 4.

3. Definition of Done

Definition of Done		
Items No.	Modules	checklist
1	Updated Filtering	For tags, user can drag one tag key to select all tag values for that key
2		For tags, user can use filter to select one or more tag values at the same time
3		For tags, user can search the key in the drag-box
4		For fields, user can search the field name in the drag-box
5		As filtering tag values, user can search the values in a pop-up window
6		For measurements, user can search the measurement name in the drag-box
7	Updated Query	For each query tab, it will have its unique query id
8		When there are too many query tabs, user can scroll the mouse to view all of them
9	Code Generation	The interface automatically generates the corresponding Flux query in the background
10		User can click the button "run query" to create a query and generate code
11		The system processes the query and retrieves the data.
12		User can click the button "show code" to view the code
13		User can click the button "hide code" to not show the code
14		User can click the button "copy" to copy the code
15	Graph generation	When user click "run query", there is a figure shown in the dashboard
16		Add a graph module in the dashboard for future to show the figure
17		User can select the type of graph, such as line, bar, and so on

18	Testing dataset	Create some dataset with more than 10000 data
19		Find some existing dataset to test all functions
20	Real-time data preview	Add a preview data module in the dashboard
21		Based on user's selection in drag-and-drop, the preview data will be shown in 3 seconds if the user doesn't have any other movements.
22		Once user finishes their selection (in 3 seconds without any other movements), the front-end send the request to the backend successfully
23		The preview must update dynamically as the user makes selections or changes.
24		The interface should handle large datasets efficiently, providing previews without significant delay.
25	Documentation	Finish and submit snapshot 4.1 on time
26		Finish and submit snapshot 4.2 on time
27		Analysis the updated requirements demonstrated in sprint meeting, separate it into tasks, and write the tasks on the project board
28		After the meeting, summarize meeting notes for both the team and Sprint meetings and upload for future reference.

4. Summary of Changes

- **Planning:**
    - **Update Filter Selection Logic**

      To deal with large datasets, we were advised to introduce features to search each filter value instead of user scrolling for locating the filter value, and to modify the existing tag filter to cater for situations that multiple tag keys coupled with multiple tag values in large datasets.

    - **Code Generation and Query Logic**

      After selection, Flux code for each query will be generated and only show until user click “show code” button. User can click “Hide Code” or “Copy Code” button to hide or copy code. For each query user establishes, unique query ID will be implemented in the logic so that

    - **Preview Data after Selection**

      We have been introduced to a new user story 4, which is about preview data after selection. Our plan is to add a “preview data” button so that user can click and the selected data will be retrieved and returned to the frontend. Preview data will show the summary of lines and entries of all the selected data with header information so that users can determine if they would like to proceed with the query and graph generation.

- **In Progress:**
    - **Filter Search Logic**

      Will add new search feature for each filter to enable filter value search feature in some large datasets, especially for tag and field filters. Will add a new filter logic to the tag filter to cope with situations that handle multiple tag keys pair with multiple tag values.

    - **Code Generation**

      Working on user story 3 to support automatic Flux code generation at the background, and when “show code” button is clicked, the interface will display the flux code based on the selections. Once the code is displayed, the button will change state to “Hide Code”, once it’s clicked, the code area will be hidden.

- **Done**
    - **Filter Logic**

      Dynamic available buckets and measurement, tag and field filters have been done. When users log into the interface, they can select a bucket from their existing account database, and then measurements will be presented automatically, and then the tags and field filters. Users can also select the start and end time range via the filter logic as well.

    - **Meeting Minutes**

      Sprint 3 review and sprint 4 planning meeting minutes are finished and documented.