Snapshot Week 11 of Group

InfluxUI-PG01 No-Code Solution for

InfluxDB

by

Xiaoyue Rao a1819070

Jianghao Jin a1880849

Tiantian Wang a1894037

Manhong Chen a1904387

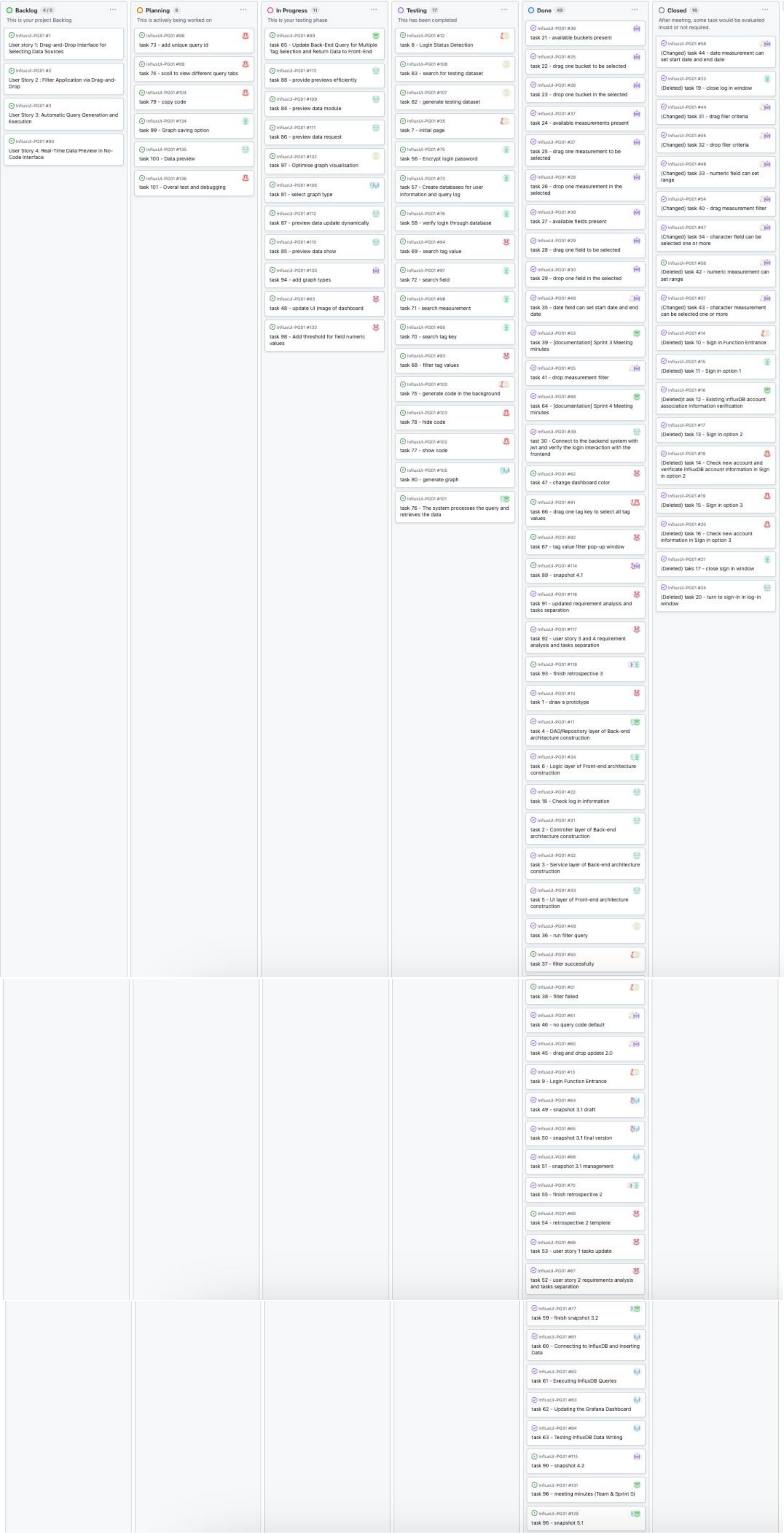
Ziyan Zhao a1883303

Ling Luo a1847162

Yufei Wang a1897360

Yinkai Yuan a1909218

1. Product Backlog and Task Board



Backlog 4 / 5

This is your project Backlog

- InfluxUI-PG01 #1 — User story 1: Drag-and-Drop Interface for Selecting Data Sources
- InfluxUI-PG01 #2 — User Story 2 : Filter Application via Drag-and-Drop
- InfluxUI-PG01 #3 — User Story 3: Automatic Query Generation and Execution
- InfluxUI-PG01 #90 — User Story 4: Real-Time Data Preview in No-Code Interface

Planning 6

This is actively being worked on

- InfluxUI-PG01 #98 — task 73 - add unique query id
- InfluxUI-PG01 #99 — task 74 - scoll to view different query tabs
- InfluxUI-PG01 #104 — task 79 - copy code
- InfluxUI-PG01 #134 — task 99 - Graph saving option
- InfluxUI-PG01 #135 — task 100 - Data preview
- InfluxUI-PG01 #136 — task 101 - Overal test and debugging

In Progress 11

This is your testing phase

- InfluxUI-PG01 #89 — task 65 - Update Back-End Query for Multiple Tag Selection and Return Data to Front-End
- InfluxUI-PG01 #113 — task 88 - provide previews efficiently
- InfluxUI-PG01 #109 — task 84 - preview data module
- InfluxUI-PG01 #111 — task 86 - preview data request
- InfluxUI-PG01 #132 — task 97 - Optimise graph visualisation
- InfluxUI-PG01 #106 — task 81 - select graph type
- InfluxUI-PG01 #112 — task 87 - preview data update dynamically
- InfluxUI-PG01 #110 — task 85 - preview data show
- InfluxUI-PG01 #130 — task 94 - add graph types
- InfluxUI-PG01 #63 — task 48 - update UI image of dashboard
- InfluxUI-PG01 #133 — task 98 - Add threshold for field numeric values

Testing 17

This has been completed

- InfluxUI-PG01 #12 — task 8 - Login Status Detection
- InfluxUI-PG01 #108 — task 83 - search for testing dataset
- InfluxUI-PG01 #107 — task 82 - generate testing dataset
- InfluxUI-PG01 #35 — task 7 - initial page
- InfluxUI-PG01 #75 — task 56 - Encrypt login password
- InfluxUI-PG01 #73 — task 57 - Create databases for user information and query log
- InfluxUI-PG01 #76 — task 58 - verify login through database
- InfluxUI-PG01 #94 — task 69 - search tag value
- InfluxUI-PG01 #97 — task 72 - search field
- InfluxUI-PG01 #96 — task 71 - search measurement
- InfluxUI-PG01 #95 — task 70 - search tag key
- InfluxUI-PG01 #93 — task 68 - filter tag values
- InfluxUI-PG01 #100 — task 75 - generate code in the background
- InfluxUI-PG01 #103 — task 78 - hide code
- InfluxUI-PG01 #102 — task 77 - show code
- InfluxUI-PG01 #105 — task 80 - generate graph
- InfluxUI-PG01 #101 — task 76 - The system processes the query and retrieves the data

Done 49

- InfluxUI-PG01 #36 — task 21 - available buckets present
- InfluxUI-PG01 #25 — task 22 - drag one bucket to be selected
- InfluxUI-PG01 #26 — task 23 - drop one bucket in the selected
- InfluxUI-PG01 #37 — task 24 - available measurements present
- InfluxUI-PG01 #27 — task 25 - drag one measurement to be selected
- InfluxUI-PG01 #28 — task 26 - drop one measurement in the selected
- InfluxUI-PG01 #38 — task 27 - available fields present
- InfluxUI-PG01 #29 — task 28 - drag one field to be selected
- InfluxUI-PG01 #30 — task 29 - drop one field in the selected
- InfluxUI-PG01 #48 — task 35 - date field can set start date and end date
- InfluxUI-PG01 #53 — task 39 - [documentation] Sprint 3 Meeting minutes
- InfluxUI-PG01 #55 — task 41 - drop measurement filter
- InfluxUI-PG01 #88 — task 64 - [documentation] Sprint 4 Meeting minutes
- InfluxUI-PG01 #39 — tast 30 - Connect to the backend system with jwt and verify the login interaction with the frontend
- InfluxUI-PG01 #62 — task 47 - change dashboard color
- InfluxUI-PG01 #91 — task 66 - drag one tag key to select all tag values
- InfluxUI-PG01 #92 — task 67 - tag value filter pop-up window
- InfluxUI-PG01 #114 — task 89 - snapshot 4.1
- InfluxUI-PG01 #116 — task 91 - updated requirement analysis and tasks separation
- InfluxUI-PG01 #117 — task 92 - user story 3 and 4 requirement analysis and tasks separation
- InfluxUI-PG01 #118 — task 93 - finish retrospective 3
- InfluxUI-PG01 #10 — task 1 - draw a prototype
- InfluxUI-PG01 #11 — task 4 - DAO/Repository layer of Back-end architecture construction
- InfluxUI-PG01 #34 — task 6 - Logic layer of Front-end architecture construction
- InfluxUI-PG01 #22 — task 18 - Check log in information
- InfluxUI-PG01 #31 — task 2 - Controller layer of Back-end architecture construction
- InfluxUI-PG01 #32 — task 3 - Service layer of Back-end architecture construction
- InfluxUI-PG01 #33 — task 5 - UI layer of Front-end architecture construction
- InfluxUI-PG01 #49 — task 36 - run filter query
- InfluxUI-PG01 #50 — task 37 - filter successfully
- InfluxUI-PG01 #51 — task 38 - filter failed
- InfluxUI-PG01 #61 — task 46 - no query code default
- InfluxUI-PG01 #60 — task 45 - drag and drop update 2.0
- InfluxUI-PG01 #13 — task 9 - Login Function Entrance
- InfluxUI-PG01 #64 — task 49 - snapshot 3.1 draft
- InfluxUI-PG01 #65 — task 50 - snapshot 3.1 final version
- InfluxUI-PG01 #66 — task 51 - snapshot 3.1 management
- InfluxUI-PG01 #70 — task 55 - finish retrospective 2
- InfluxUI-PG01 #69 — task 54 - retrospective 2 templete
- InfluxUI-PG01 #68 — task 53 - user story 1 tasks update
- InfluxUI-PG01 #67 — task 52 - user story 2 requirements analysis and tasks separation
- InfluxUI-PG01 #77 — task 59 - finish snapshot 3.2
- InfluxUI-PG01 #81 — task 60 - Connecting to InfluxDB and Inserting Data
- InfluxUI-PG01 #82 — task 61 - Executing InfluxDB Queries
- InfluxUI-PG01 #83 — task 62 - Updating the Grafana Dashboard
- InfluxUI-PG01 #84 — task 63 - Testing InfluxDB Data Writing
- InfluxUI-PG01 #115 — task 90 - snapshot 4.2
- InfluxUI-PG01 #131 — task 96 - meeting minutes (Team & Sprint 5)
- InfluxUI-PG01 #129 — task 95 - snapshot 5.1

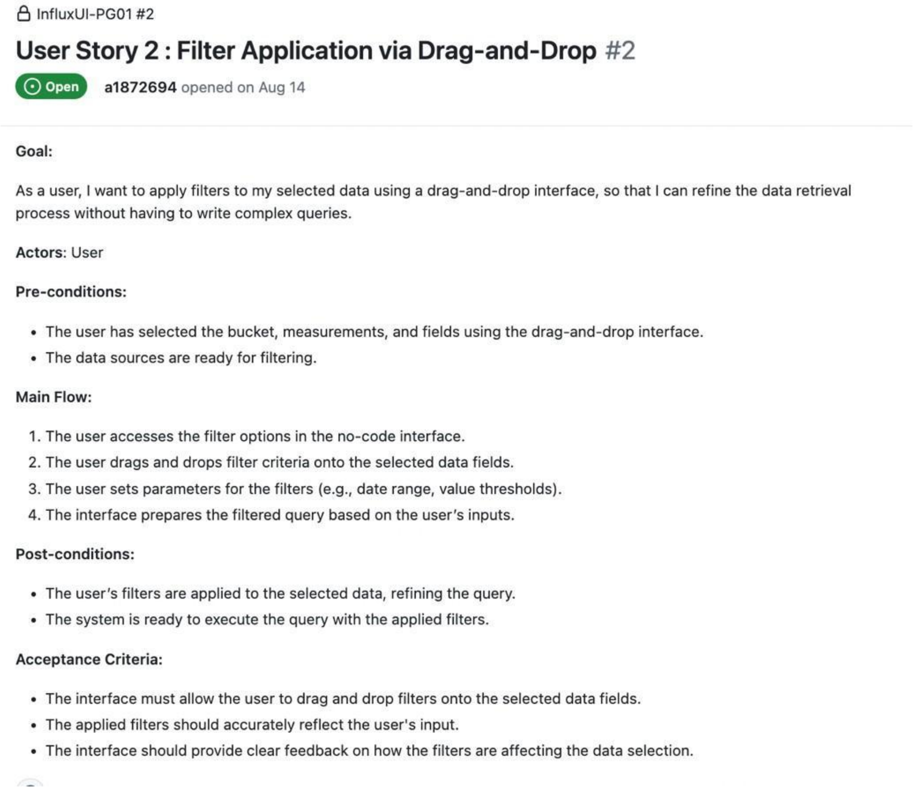
Closed 18

After meeting, some task would be evaluated invaid or not required.

- InfluxUI-PG01 #58 — (Changed) task 44 - date measurement can set start date and end date
- InfluxUI-PG01 #23 — (Deleted) task 19 - close log in window
- InfluxUI-PG01 #44 — (Changed) task 31 - drag filer criteria
- InfluxUI-PG01 #45 — (Changed) task 32 - drop filer criteria
- InfluxUI-PG01 #46 — (Changed) task 33 - numeric field can set range
- InfluxUI-PG01 #54 — (Changed) task 40 - drag measurement filter
- InfluxUI-PG01 #47 — (Changed) task 34 - character field can be selected one or more
- InfluxUI-PG01 #56 — (Deleted) task 42 - numeric measurement can set range
- InfluxUI-PG01 #57 — (Changed) task 43 - character measurement can be selected one or more
- InfluxUI-PG01 #14 — (Deleted) task 10 - Sign in Function Entrance
- InfluxUI-PG01 #15 — (Deleted) task 11 - Sign in option 1
- InfluxUI-PG01 #16 — (Deleted)t ask 12 - Existing influxDB account association information verification
- InfluxUI-PG01 #17 — (Deleted) task 13 - Sign in option 2
- InfluxUI-PG01 #18 — (Deleted) task 14 - Check new account and verificate InfluxDB account information in Sign in option 2
- InfluxUI-PG01 #19 — (Deleted) task 15 - Sign in option 3
- InfluxUI-PG01 #20 — (Deleted) task 16 - Check new account information in Sign in option 3
- InfluxUI-PG01 #21 — (Deleted) taks 17 - close sign in window
- InfluxUI-PG01 #24 — (Deleted) task 20 - turn to sign-in in log-in window

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

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.

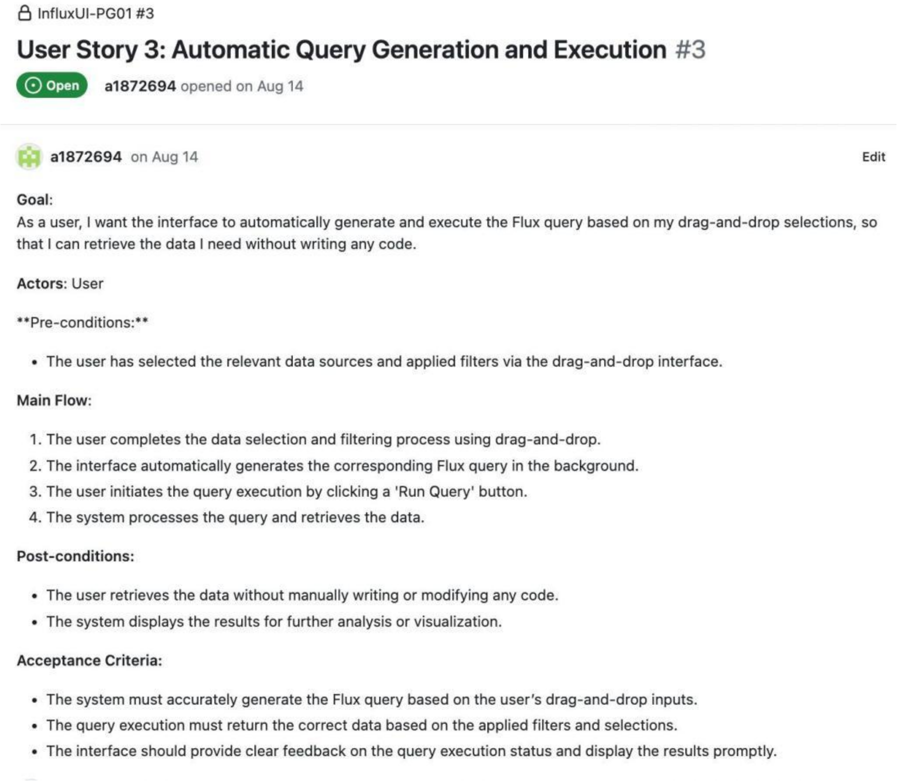
Post-conditions:

- 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

a1872694 opened on Aug 14

a1872694 on Aug 14

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.

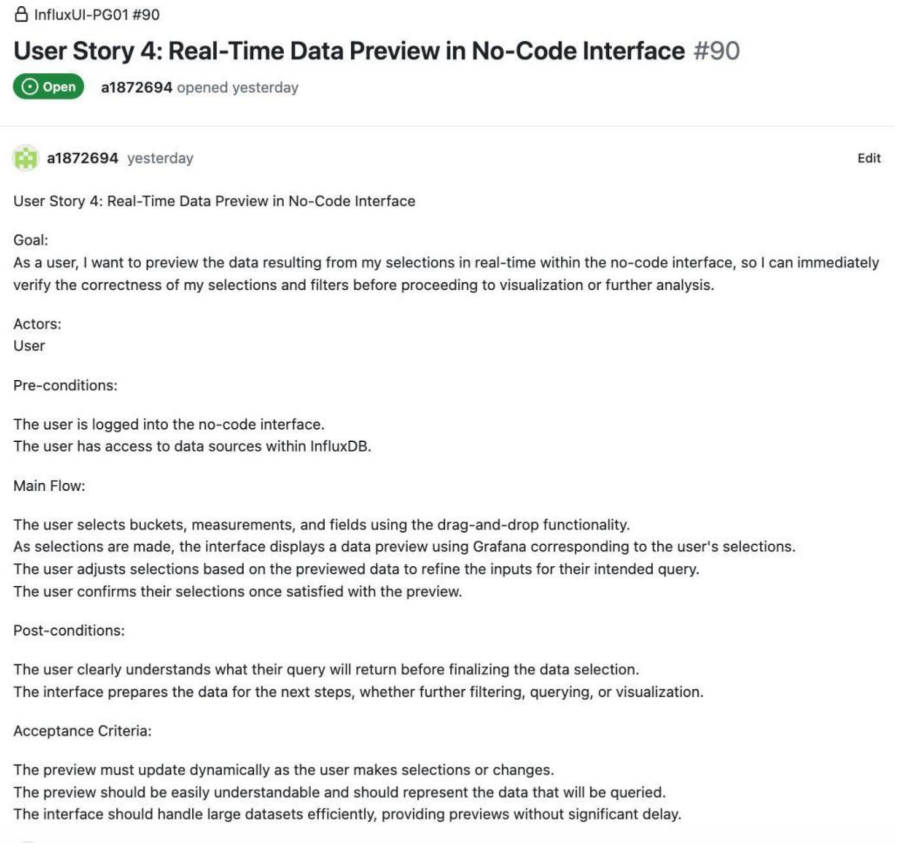
Post-conditions:

- 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

a1872694 opened yesterday

a1872694 yesterday

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.

After the discussion occurred in the Spinrt 5 PO meeting, we discovered that there is still improvement needed for the User Story 2 filter feature, as the PO wanted a range of filtering for numeric values. As for User story 3 and 4, we will continue to work on graph display in terms of providing various visualisation options for users, and adding a download function. We will also work on testing and debugging in the meanwhile.

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		For fields, user can apply range filter to numeric values.
6		As filtering tag values, user can search the values in a pop-up window
7		For measurements, user can search the measurement name in the drag-box
8	Updated query	The interface automatically generates the corresponding Flux query in the background
9		User can click the button "run query" to create a query and generate code
10		The system processes the query and retrieves the data.
11		User can click the button "show code" to view the code
12		User can click the button "hide code" to not show the code
13		User can click the button "copy" to copy the code
14	Graph generation	When user click "run query", there is a figure shown in the dashboard
15		Add a graph module in the dashboard for future to show the figure
16		User can view data in for form of a table.
17		User can select the type of graph, such as line, bar, and so on
18		User can download the graph and/or output as a file.

19	Real-time data preview	Add a preview data module in the dashboard
20		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.
21		Once user finishes their selection (in 3 seconds without any other movements), the front-end send the request to the backend successfully
22		The preview must update dynamically as the user makes selections or changes.
23		The interface should handle large datasets efficiently, providing previews without significant delay.
24	Documentation	Finish and submit snapshot 5.1 on time
25		Finish and submit snapshot 5.2 on time
26		Analysis the updated requirements demonstrated in sprint meeting, separate it into tasks, and write the tasks on the project board
27		After the meeting, summarize meeting notes for both the team and Sprint meetings and upload for future reference.

4. Summary of Changes

- **Planning:**
    - **Query Time Range Adjustment:**

      Users will be allowed to set the appropriate timeframe for the query to ensure that the resultant data is within the desired timeframe.

    - **Field Threshold Setting:**

      The ability to add threshold settings to numeric fields is planned to be developed, which will allow users to define acceptable ranges for numeric fields to make it easier to identify anomalous data and maintain data integrity.

- **In Progress:**
    - **Data Preview Enhancement:**

      A real-time data preview feature is under development This feature enables users to dynamically view and validate the correctness of selections within the interface by using Grafana for the visual presentation of data.

    - **Saving Visual Outputs:**

      For better sharing and reporting of data, an image saving feature is under development. This feature will enable users to save Grafana-generated images to a variety of formats such as PDF and PNG.

    - **Data Visualization by Table or Graph:**

      Multiple graphical image presentation features are being developed. This functionality allows users to choose between tabular data presentations and graphical formats (such as line graphs and bar charts) to ensure that users have the flexibility to analyse and interpret the data in the most appropriate way.

    - **UI Bug Fixes:**

      Bugs in the current user interface (e.g. table numbering, etc.) are being identified and resolved.