Snapshot Week 12 of Group InfluxUI-

PG01 No-Code Solution for InfluxDB

by

Xiaoyue Rao a1819070

Jianghao Jin a1880849

Tiantian Wang a1894037

Manhong Chen a1904387

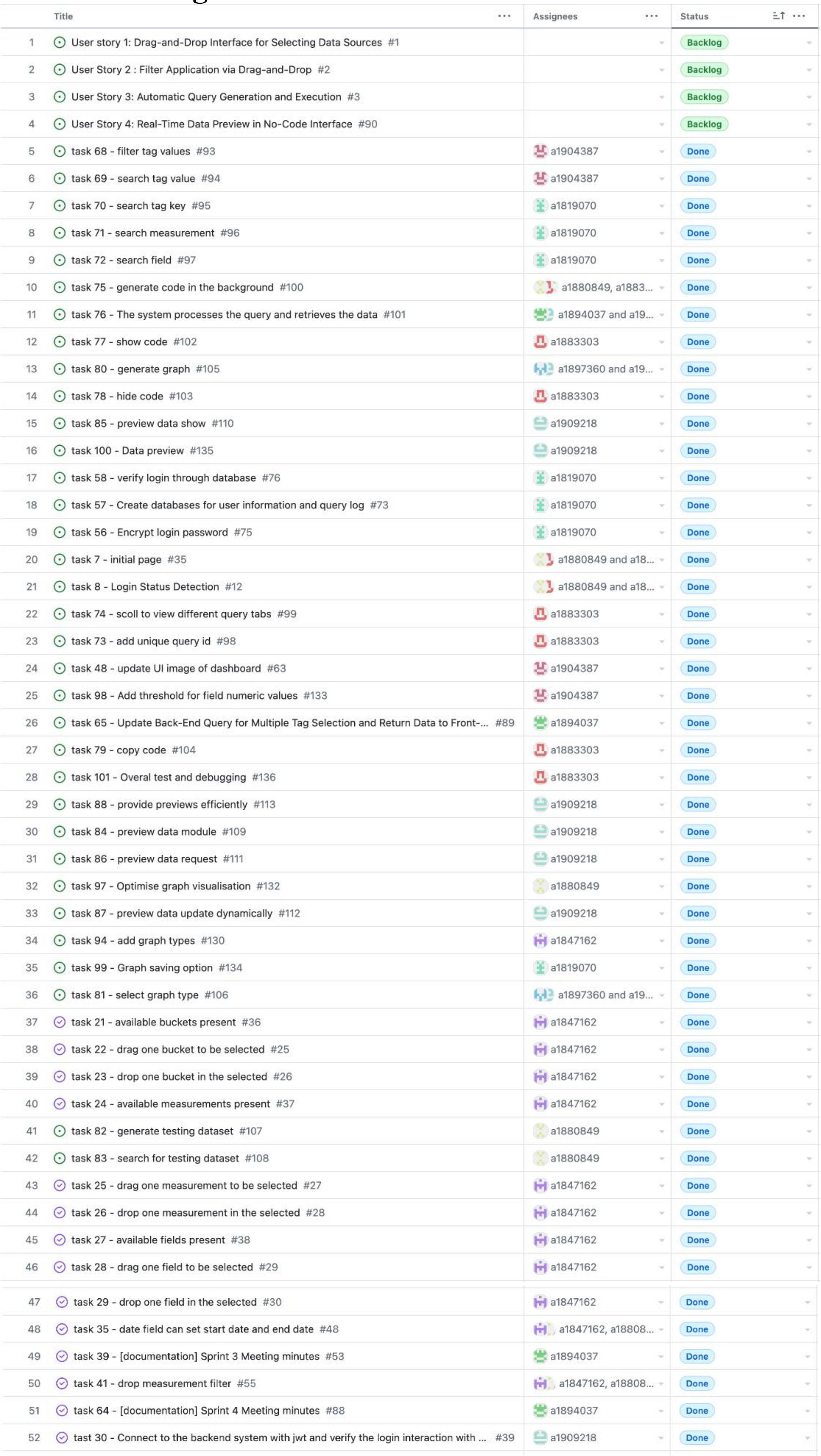
Ziyan Zhao a1883303

Ling Luo a1847162

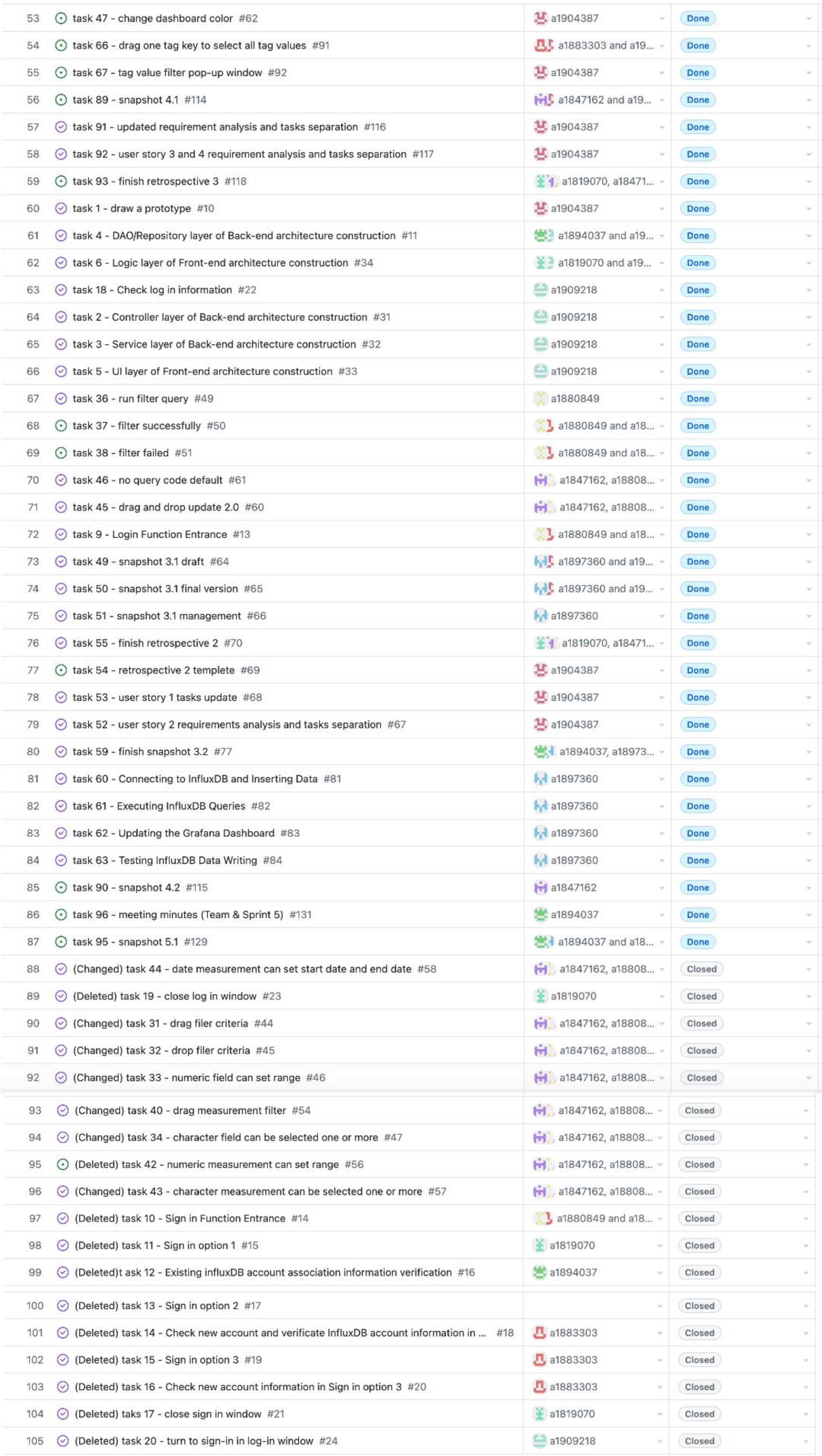
Yufei Wang a1897360

Yinkai Yuan a1909218

1. Product Backlog and Task Board



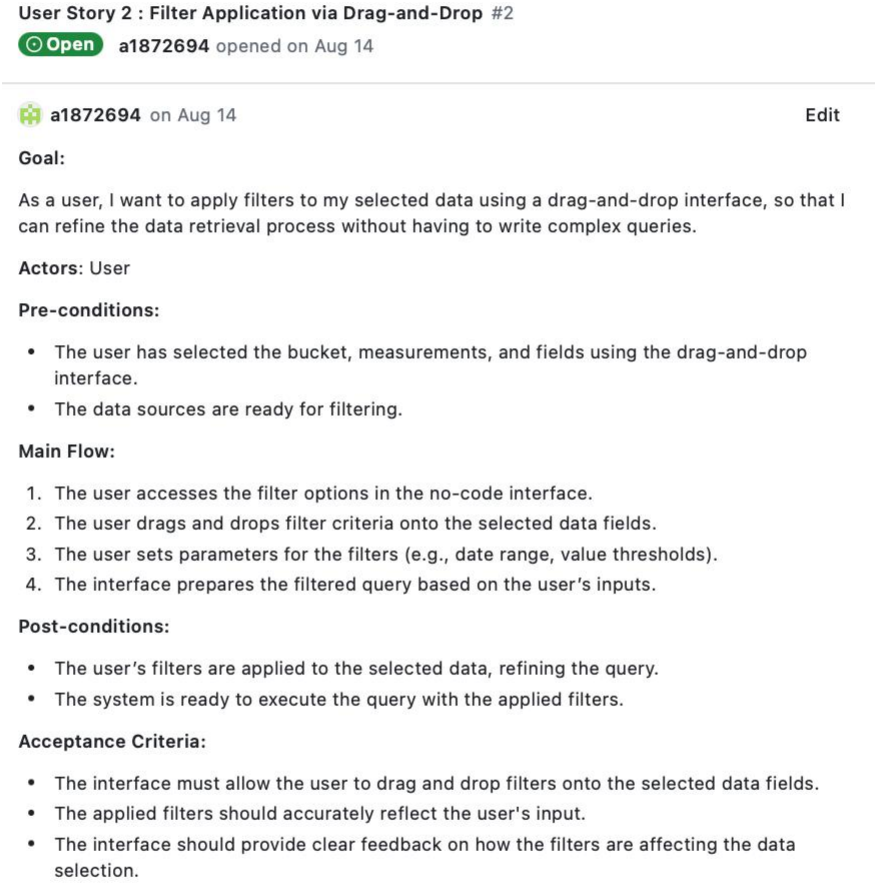
	Title	Assignees	Status
1	User story 1: Drag-and-Drop Interface for Selecting Data Sources #1		Backlog
2	User Story 2 : Filter Application via Drag-and-Drop #2		Backlog
3	User Story 3: Automatic Query Generation and Execution #3		Backlog
4	User Story 4: Real-Time Data Preview in No-Code Interface #90		Backlog
5	task 68 - filter tag values #93	a1904387	Done
6	task 69 - search tag value #94	a1904387	Done
7	task 70 - search tag key #95	a1819070	Done
8	task 71 - search measurement #96	a1819070	Done
9	task 72 - search field #97	a1819070	Done
10	task 75 - generate code in the background #100	a1880849, a1883...	Done
11	task 76 - The system processes the query and retrieves the data #101	a1894037 and a19...	Done
12	task 77 - show code #102	a1883303	Done
13	task 80 - generate graph #105	a1897360 and a19...	Done
14	task 78 - hide code #103	a1883303	Done
15	task 85 - preview data show #110	a1909218	Done
16	task 100 - Data preview #135	a1909218	Done
17	task 58 - verify login through database #76	a1819070	Done
18	task 57 - Create databases for user information and query log #73	a1819070	Done
19	task 56 - Encrypt login password #75	a1819070	Done
20	task 7 - initial page #35	a1880849 and a18...	Done
21	task 8 - Login Status Detection #12	a1880849 and a18...	Done
22	task 74 - scoll to view different query tabs #99	a1883303	Done
23	task 73 - add unique query id #98	a1883303	Done
24	task 48 - update UI image of dashboard #63	a1904387	Done
25	task 98 - Add threshold for field numeric values #133	a1904387	Done
26	task 65 - Update Back-End Query for Multiple Tag Selection and Return Data to Front-... #89	a1894037	Done
27	task 79 - copy code #104	a1883303	Done
28	task 101 - Overal test and debugging #136	a1883303	Done
29	task 88 - provide previews efficiently #113	a1909218	Done
30	task 84 - preview data module #109	a1909218	Done
31	task 86 - preview data request #111	a1909218	Done
32	task 97 - Optimise graph visualisation #132	a1880849	Done
33	task 87 - preview data update dynamically #112	a1909218	Done
34	task 94 - add graph types #130	a1847162	Done
35	task 99 - Graph saving option #134	a1819070	Done
36	task 81 - select graph type #106	a1897360 and a19...	Done
37	task 21 - available buckets present #36	a1847162	Done
38	task 22 - drag one bucket to be selected #25	a1847162	Done
39	task 23 - drop one bucket in the selected #26	a1847162	Done
40	task 24 - available measurements present #37	a1847162	Done
41	task 82 - generate testing dataset #107	a1880849	Done
42	task 83 - search for testing dataset #108	a1880849	Done
43	task 25 - drag one measurement to be selected #27	a1847162	Done
44	task 26 - drop one measurement in the selected #28	a1847162	Done
45	task 27 - available fields present #38	a1847162	Done
46	task 28 - drag one field to be selected #29	a1847162	Done
47	task 29 - drop one field in the selected #30	a1847162	Done
48	task 35 - date field can set start date and end date #48	a1847162, a18808...	Done
49	task 39 - [documentation] Sprint 3 Meeting minutes #53	a1894037	Done
50	task 41 - drop measurement filter #55	a1847162, a18808...	Done
51	task 64 - [documentation] Sprint 4 Meeting minutes #88	a1894037	Done
52	tast 30 - Connect to the backend system with jwt and verify the login interaction with ... #39	a1909218	Done



53	⊙	task 47 - change dashboard color #62	a1904387	Done
54	⊙	task 66 - drag one tag key to select all tag values #91	a1883303 and a19...	Done
55	⊙	task 67 - tag value filter pop-up window #92	a1904387	Done
56	⊙	task 89 - snapshot 4.1 #114	a1847162 and a19...	Done
57	✓	task 91 - updated requirement analysis and tasks separation #116	a1904387	Done
58	✓	task 92 - user story 3 and 4 requirement analysis and tasks separation #117	a1904387	Done
59	⊙	task 93 - finish retrospective 3 #118	a1819070, a18471...	Done
60	✓	task 1 - draw a prototype #10	a1904387	Done
61	✓	task 4 - DAO/Repository layer of Back-end architecture construction #11	a1894037 and a19...	Done
62	✓	task 6 - Logic layer of Front-end architecture construction #34	a1819070 and a19...	Done
63	✓	task 18 - Check log in information #22	a1909218	Done
64	✓	task 2 - Controller layer of Back-end architecture construction #31	a1909218	Done
65	✓	task 3 - Service layer of Back-end architecture construction #32	a1909218	Done
66	✓	task 5 - UI layer of Front-end architecture construction #33	a1909218	Done
67	✓	task 36 - run filter query #49	a1880849	Done
68	⊙	task 37 - filter successfully #50	a1880849 and a18...	Done
69	⊙	task 38 - filter failed #51	a1880849 and a18...	Done
70	✓	task 46 - no query code default #61	a1847162, a18808...	Done
71	✓	task 45 - drag and drop update 2.0 #60	a1847162, a18808...	Done
72	✓	task 9 - Login Function Entrance #13	a1880849 and a18...	Done
73	✓	task 49 - snapshot 3.1 draft #64	a1897360 and a19...	Done
74	✓	task 50 - snapshot 3.1 final version #65	a1897360 and a19...	Done
75	✓	task 51 - snapshot 3.1 management #66	a1897360	Done
76	✓	task 55 - finish retrospective 2 #70	a1819070, a18471...	Done
77	⊙	task 54 - retrospective 2 templete #69	a1904387	Done
78	✓	task 53 - user story 1 tasks update #68	a1904387	Done
79	✓	task 52 - user story 2 requirements analysis and tasks separation #67	a1904387	Done
80	✓	task 59 - finish snapshot 3.2 #77	a1894037, a18973...	Done
81	✓	task 60 - Connecting to InfluxDB and Inserting Data #81	a1897360	Done
82	✓	task 61 - Executing InfluxDB Queries #82	a1897360	Done
83	✓	task 62 - Updating the Grafana Dashboard #83	a1897360	Done
84	✓	task 63 - Testing InfluxDB Data Writing #84	a1897360	Done
85	⊙	task 90 - snapshot 4.2 #115	a1847162	Done
86	⊙	task 96 - meeting minutes (Team & Sprint 5) #131	a1894037	Done
87	⊙	task 95 - snapshot 5.1 #129	a1894037 and a18...	Done
88	✓	(Changed) task 44 - date measurement can set start date and end date #58	a1847162, a18808...	Closed
89	✓	(Deleted) task 19 - close log in window #23	a1819070	Closed
90	✓	(Changed) task 31 - drag filer criteria #44	a1847162, a18808...	Closed
91	✓	(Changed) task 32 - drop filer criteria #45	a1847162, a18808...	Closed
92	✓	(Changed) task 33 - numeric field can set range #46	a1847162, a18808...	Closed
93	✓	(Changed) task 40 - drag measurement filter #54	a1847162, a18808...	Closed
94	✓	(Changed) task 34 - character field can be selected one or more #47	a1847162, a18808...	Closed
95	⊙	(Deleted) task 42 - numeric measurement can set range #56	a1847162, a18808...	Closed
96	✓	(Changed) task 43 - character measurement can be selected one or more #57	a1847162, a18808...	Closed
97	✓	(Deleted) task 10 - Sign in Function Entrance #14	a1880849 and a18...	Closed
98	✓	(Deleted) task 11 - Sign in option 1 #15	a1819070	Closed
99	✓	(Deleted)t ask 12 - Existing influxDB account association information verification #16	a1894037	Closed
100	✓	(Deleted) task 13 - Sign in option 2 #17		Closed
101	✓	(Deleted) task 14 - Check new account and verificate InfluxDB account information in ... #18	a1883303	Closed
102	✓	(Deleted) task 15 - Sign in option 3 #19	a1883303	Closed
103	✓	(Deleted) task 16 - Check new account information in Sign in option 3 #20	a1883303	Closed
104	✓	(Deleted) taks 17 - close sign in window #21	a1819070	Closed
105	✓	(Deleted) task 20 - turn to sign-in in log-in window #24	a1909218	Closed

Fig.1 The Backlog of Project

2. Sprint Backlog and User Stories



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

⊙ Open a1872694 opened on Aug 14

a1872694 on Aug 14 Edit

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.

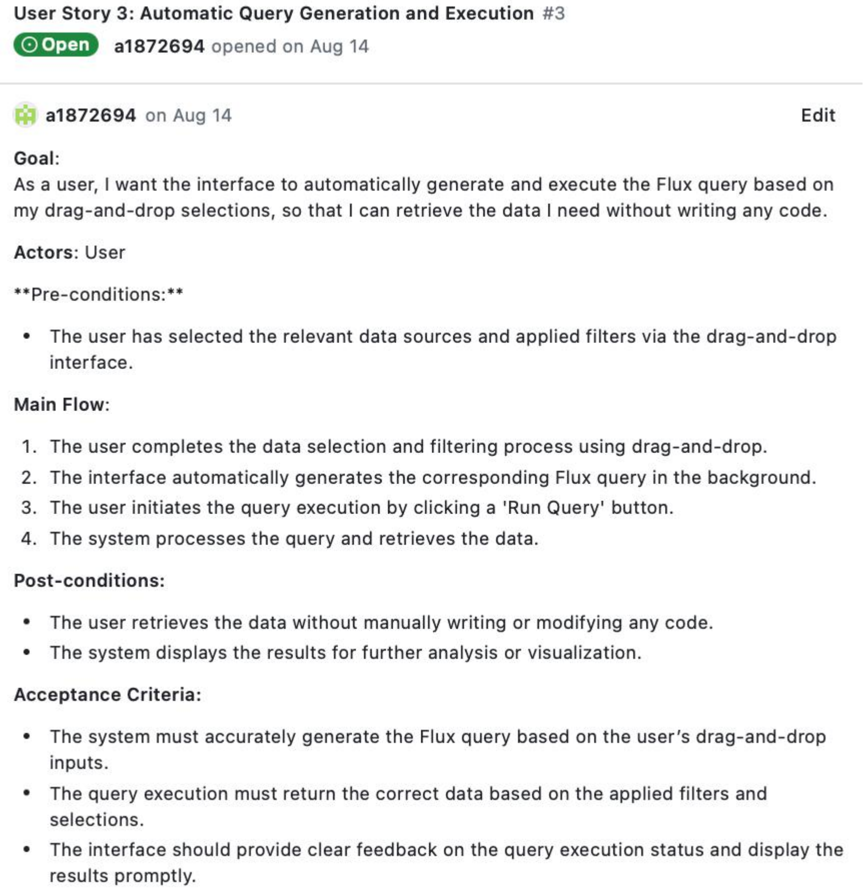
Post-conditions:

- 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



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.

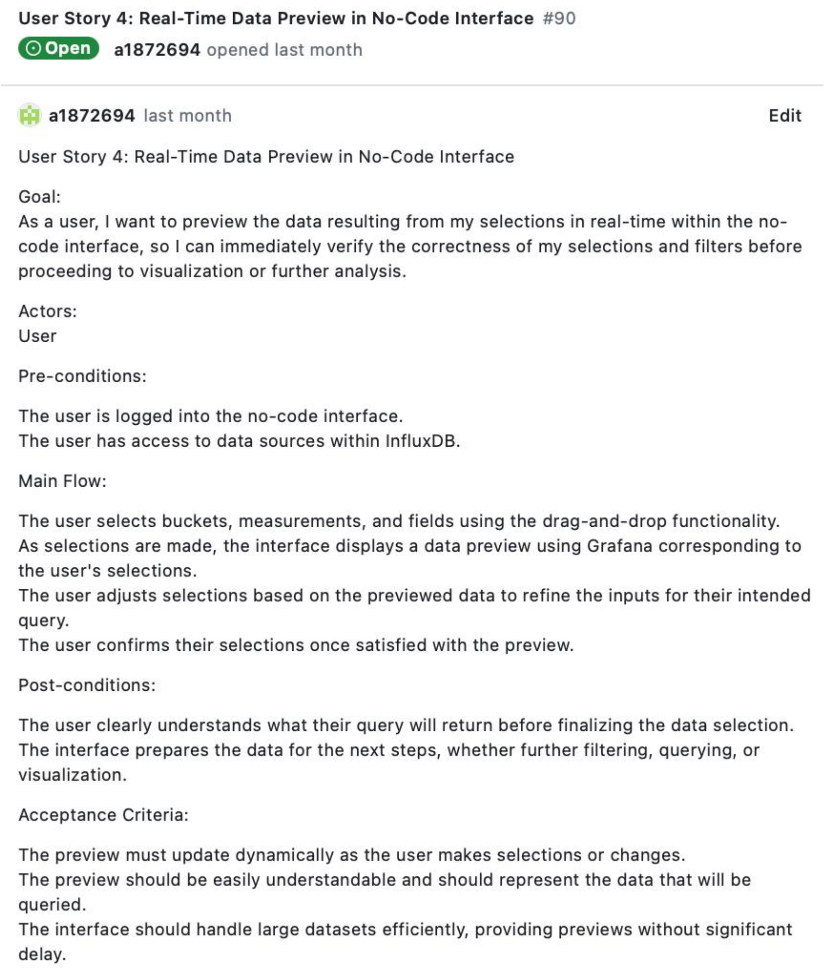
Post-conditions:

- 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


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

Open a1872694 opened last month

a1872694 last month 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.

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		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	Real-time data preview	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

- **Done:**

  - **Query Time Range Adjustment:**

    Users have been 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 has been developed, allowing users to define acceptable ranges for numeric fields to identify anomalous data more easily and maintain data integrity.

  - **Data Preview Enhancement:**

    A real-time data preview feature has been developed. This feature has enabled 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 has been developed. This feature has enabled users to save Grafana-generated images in various formats such as PDF and PNG.

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

Multiple graphical image presentation features have been developed. This functionality has allowed users to choose between tabular data presentations and graphical formats (such as line graphs and bar charts) to ensure flexibility in analyzing and interpreting the data.

- **UI Bug Fixes:**

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