



Snapshot Week 8 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

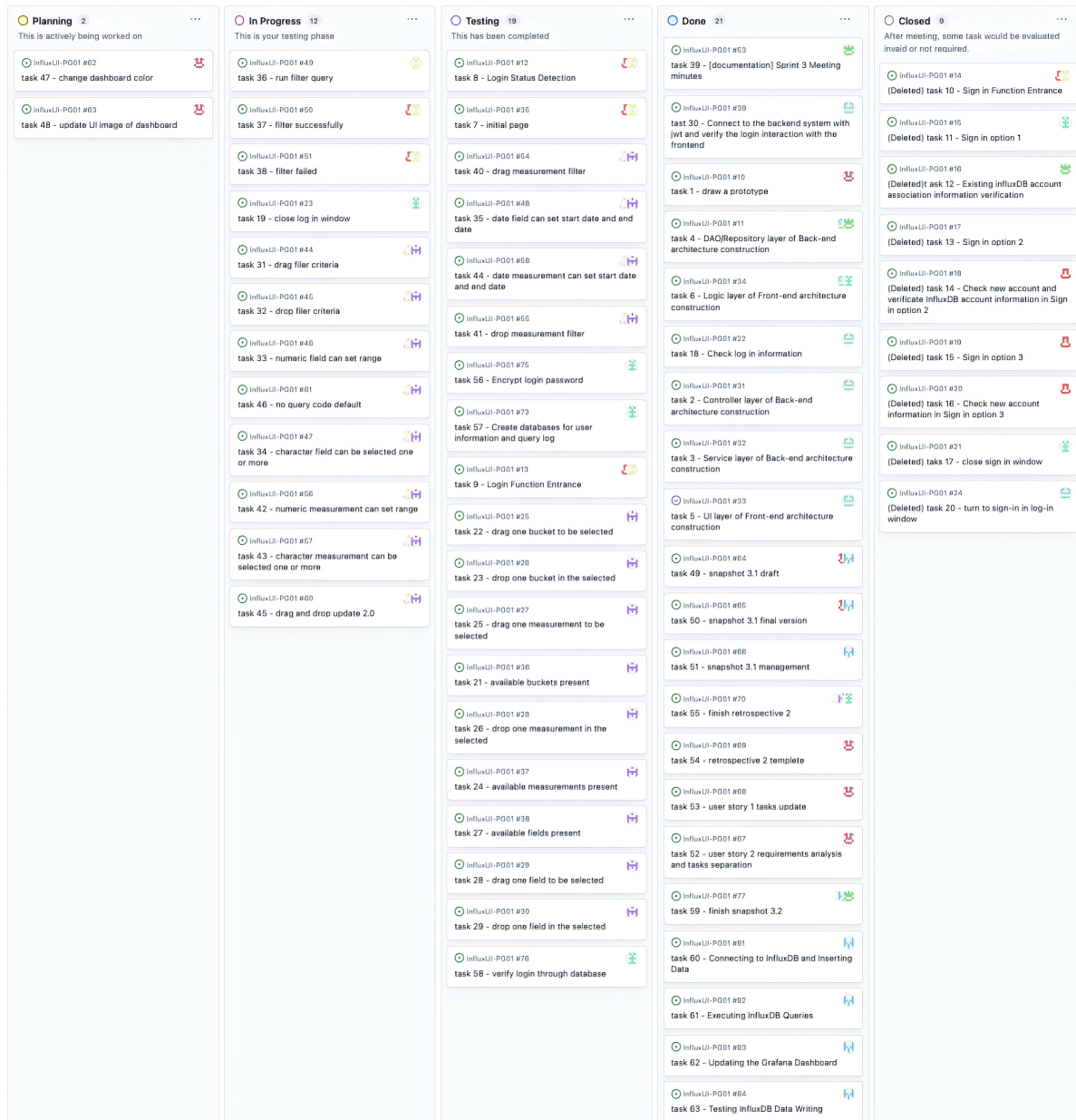





Fig.1 The Backlog of Project

2. Sprint Backlog and User Stories

 InfluxUI-PG01 #1

User story 1: Drag-and-Drop Interface for Selecting Data Sources #1

 **Open** a1872694 opened last month

 a1872694 last month Edit

User Story 1: Drag-and-Drop Interface for Selecting Data Sources

Goal:
As a user, I want to use a drag-and-drop interface to select the bucket, measurements, and fields from InfluxDB, so that I can easily choose the data I need without writing code.

Actors:
User

Pre-conditions:
The user is logged into the no-code interface.

Main Flow:
The user logs into the no-code interface.
The user is presented with a list of available buckets, measurements, and fields.
The user selects the desired data sources by dragging and dropping items into the query builder area.
The interface automatically prepares these selections for the next steps in the data query process.

Post-conditions:
The selected buckets, measurements, and fields are ready for filtering and querying.
The user successfully prepares the data sources without writing any code.

Acceptance Criteria:
The interface must allow the user to drag and drop items to select buckets, measurements, and fields.
The selected items must be accurately reflected in the query builder.







Fig. 2 User Story 1

 InfluxUI-PG01 #2

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

 Open

 a1872694 opened last month

 a1872694 last month

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. 3 User Story 2

2.1 Brief description of the user stories selected

There are two user stories focused on in Sprint 3. One is user story 1, whose aim is as a user, they want to use a drag-and-drop interface to select buckets, measurements, and fields from influxDB, so that they can easily choose the data they need without writing code. The other is user story 2, in which people 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.

In the last sprint, we finished the front-end development, but didn't connect the website with a real database. Also, with more details about the drag-and-drop function, we should update what we have done in Sprint 3. In addition, user story 2 should be started in this Sprint based on our plan. Therefore, our team would focus on these two user stories in Sprint 3.

3. Definition of Done

Items No.	Modules	checklist
1	Filtering	User can drag a field to the filter
2		User can drop a field in the filter
3		If the field is a data with numeric type, user can set a data range (both lower bound and upper bound)
4		If the field is a data with character type, the user can select one or more data of the field.
5		If the field is a data with date type, user can set the start date and end date
6		Users click "Run" button and the system run the filter query
7		If the filter query run successfully, a message "filtering successfully" will be shown
8		If the filter query run failed, a message "filtering failed" will be shown
9		User can drag a measurement to the filter
10		user can drop a measurement in the filter
11		If the measurement is a character type, user can select one or more data of the measurement
12		If the measurement is a data with numeric type, user can set a data range (both lower bound and upper bound)

13		If the measurement is a data with date type, user can set the start date and end date
14	Drag and drop	After drag a field, measurement, or bucket, the selected one will be shown in the selected baskets and the original baskets
15		After drag and drop, there is no query code shown in the query board until the users run the generate code function
16	Dashboard	Change the UI color of the dashboard into black background and blue
17		Add open-source logos and icons in dashboard
18	Documentations	After the meeting, summarize meeting notes for both the team and Sprint meetings and upload for future reference.
19		Before Friday, 6th September, upload a first draft of snapshot 3.1 in the group chat
20		Before Sunday, 8th September, submit the final version of snapshot 3.1 on myUni
21		Upload the final version of snapshot 3.1 with PDF and docx. on the GitHub
22		Analyze the requirements demonstrated in user story 2, separate it into tasks, and write the task on the project board
23		Update the task information and add new tasks on the GitHub taskboard for user story 1 after sprint 3 PO meeting

4. Summary of Changes

- **Planning:**

- **Dashboard Enhancements:**

The customisation of the dashboard will be done through updating the UI of dashboard and enhancing its functionality.

- **Back-end Completion:**

Default behaviour will be set for missing query inputs to ensure robust system performance.

- **In Progress:**

- **Filter Query and Error Handling:**

Filtered queries are actively being run, and the system's response to successful and failed operations is being managed. Ongoing development work is focused on ensuring that these queries are processed smoothly and errors are minimized.

- **Measurement Selection and Filter Criteria:**

The feature enabling numeric, character, and date-type field selection is being refined. Work is also being done to allow filters to be applied and ranges to be set for these fields more intuitively.

- **Login Encryption and Database Creation:**

Encrypting the login password and creating databases for user information and query log are being tested.

- **Done**

- **The UI layer:Filter Application via Drag-and-Drop:**

The front-end pages have been constructed using React, with Material UI to enhance visual appearance. The drag-and-drop interface for applying filters, as outlined in User Story 2, has been completed. Users can now seamlessly apply filters to selected data fields with a smooth drag-and-drop mechanism.

- **UI Layer:**

The main components of the UI have now become functional and support the user interactions required for further testing.

- **Bucket Selection and Data Display:**

Dragging and dropping buckets, measurements, and fields have been tested.

- **InfluxDB Connection and Data Handling:**

The connection to InfluxDB has been established, queries have been executed, data has been inserted, and the data-writing functionality has been tested.

Additionally, the Grafana dashboard has been updated to reflect the latest changes and ensure seamless data visualization.