System Management: Admin operations

Final Presentation - 4th Dec 2023

By: Bijli Waale (Sannidhya Gupta, Vinkesh Bansal)

Use cases

- System can be used to automatically upload dataset / model from JSON object file
- System can be used to automatically upload version information from JSON object file
- System can be used to modify existing dataset information
- Only admin user will be allowed to access these functions

Work division

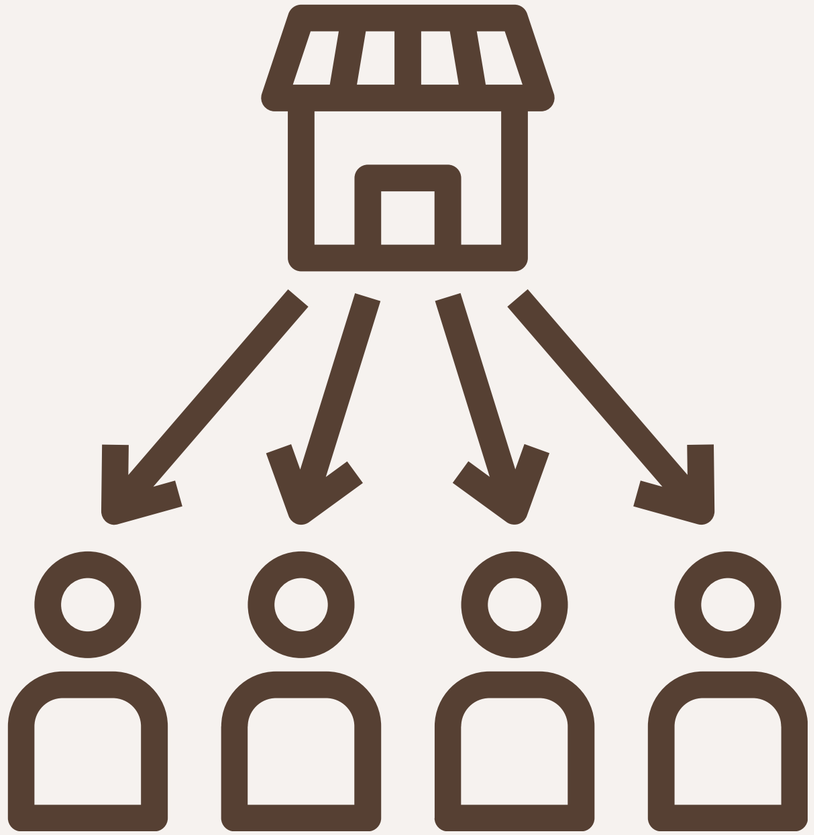
Vinkesh Bansal:

- Workflow and implementation for auto file upload for datasets and models
- File browsing implementation for JSON selection
- JSON ingestion of version data

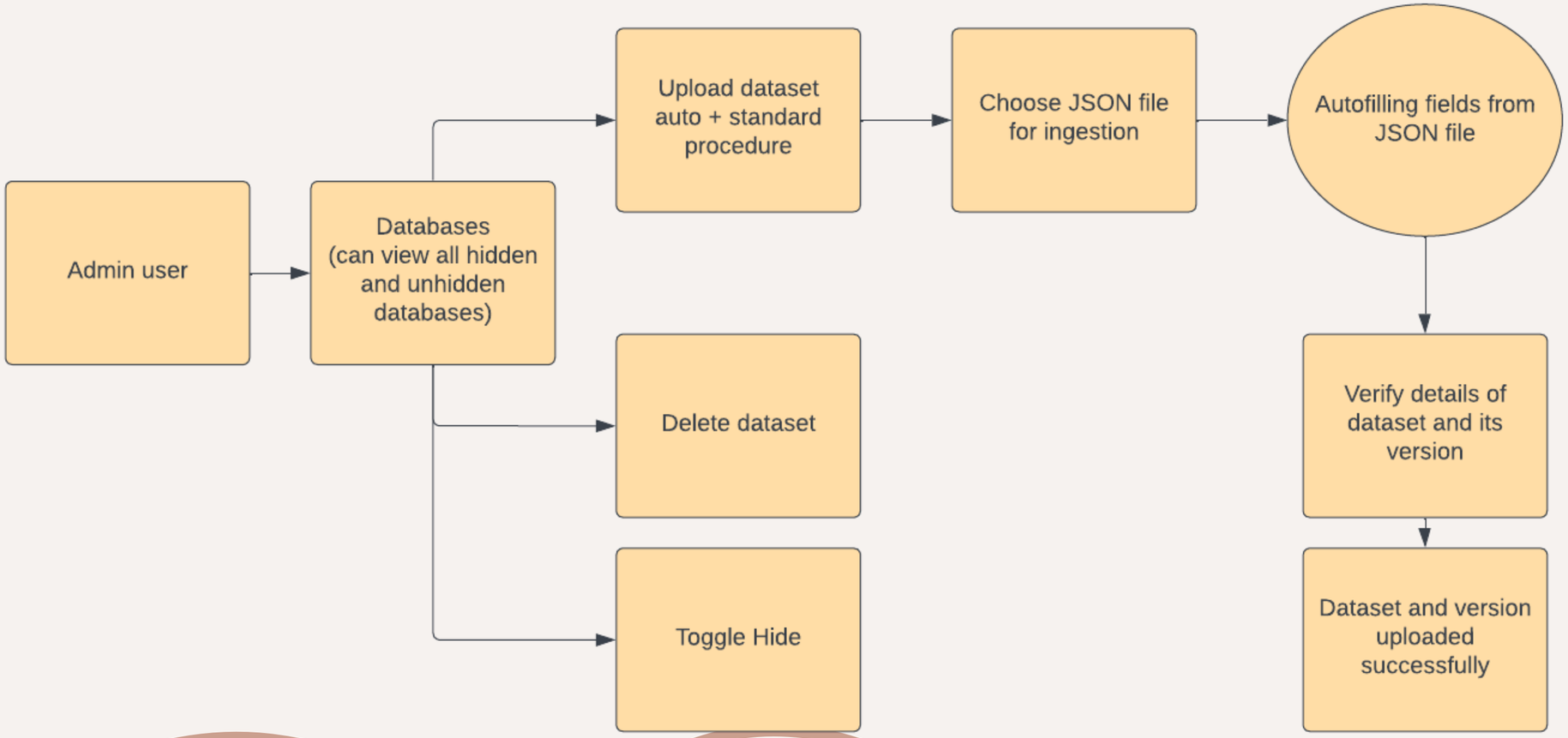
Sannidhya Gupta:

- Workflow and implementation for hiding and deletion of datasets and models
- Presentation
- Final touches for better visuals

Additional workflows were developed and implemented together

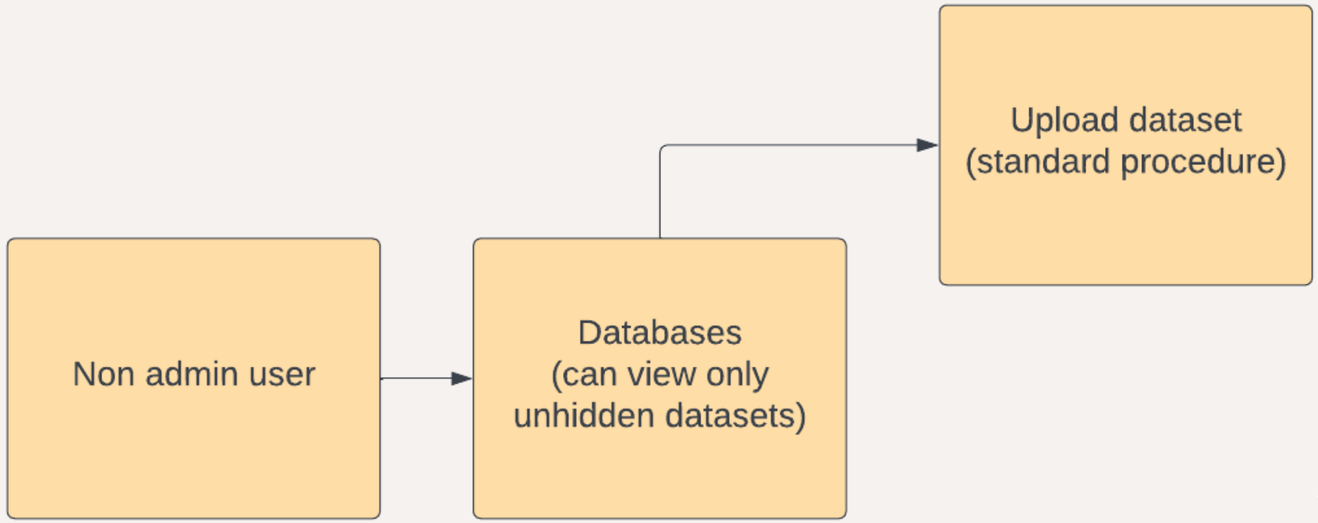


Workflow - Admin user



SQL queries are passed and database tables are updated successfully

Workflow - non admin user



Authentication permissions

Can access all datasets and models, has full control

Admin

User

Can access all datasets and models, has full control

Auto file upload

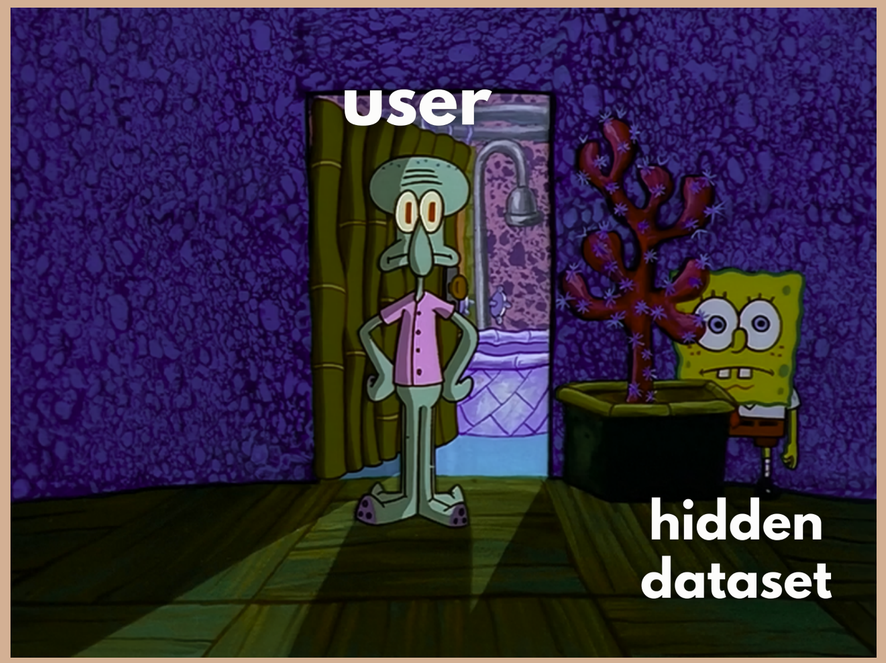
- A JSON object is used to store the required data fields about the dataset/model and versions.
- Data is extracted from the object file and pipelining into the data foundations file upload system
- The workflow and interface is kept similar to the standard file upload procedure

Deletion of datasets / models

- Successfully added functionality for deleting datasets pipelined with user authentication
- Allows Admin role users to have privileges to delete any dataset / model from the system
- Removes dataset and all its version from the system for all users

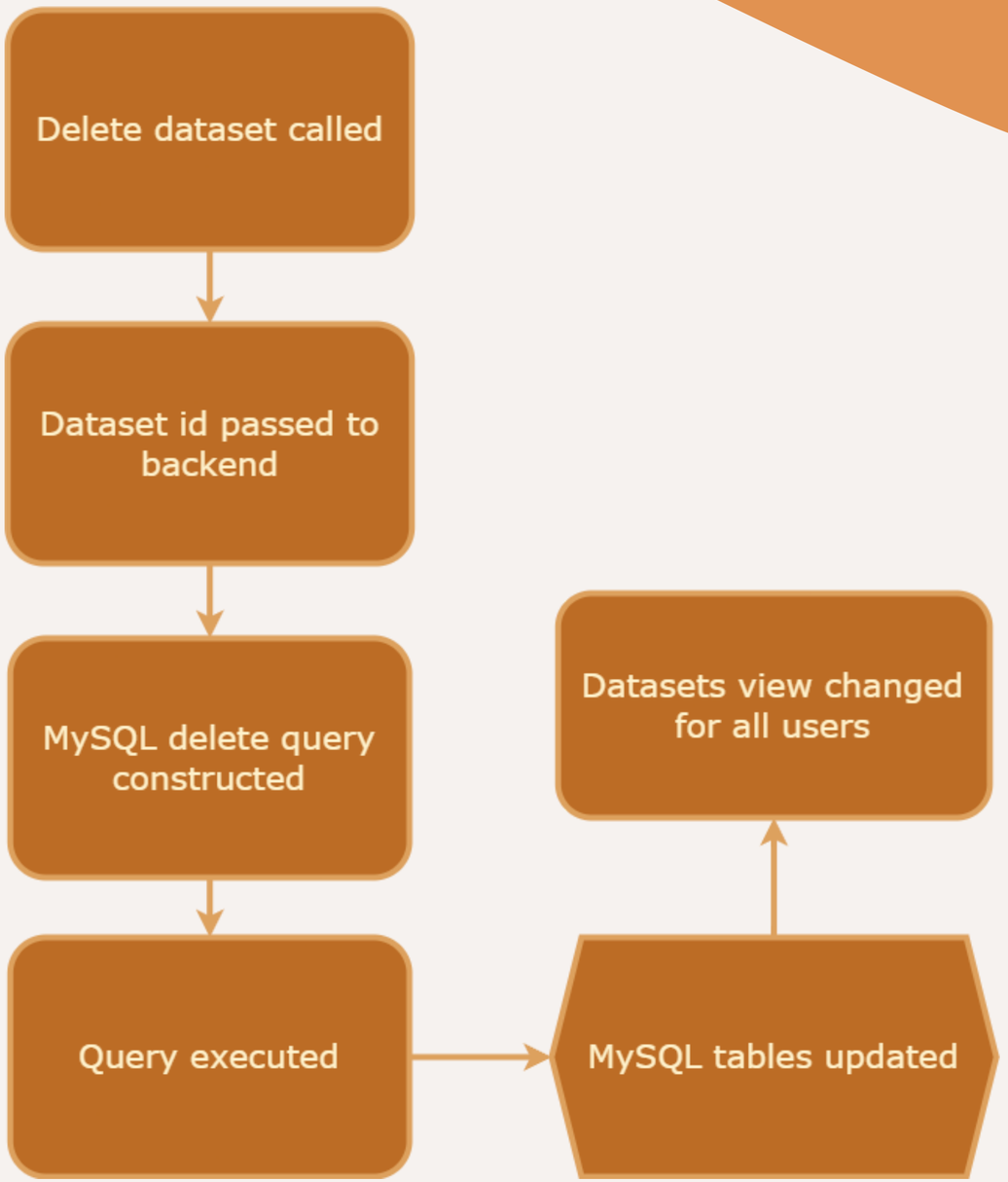
Hiding functionality

- Changed the schema and pipeline of the system to allow for addition of “hiding” for datasets and models
- Admin role users can hide / unhide a dataset or a model which will reflect in all other non admin user’s instance of the system
- Added a visibility column in the dataset for handling this across users



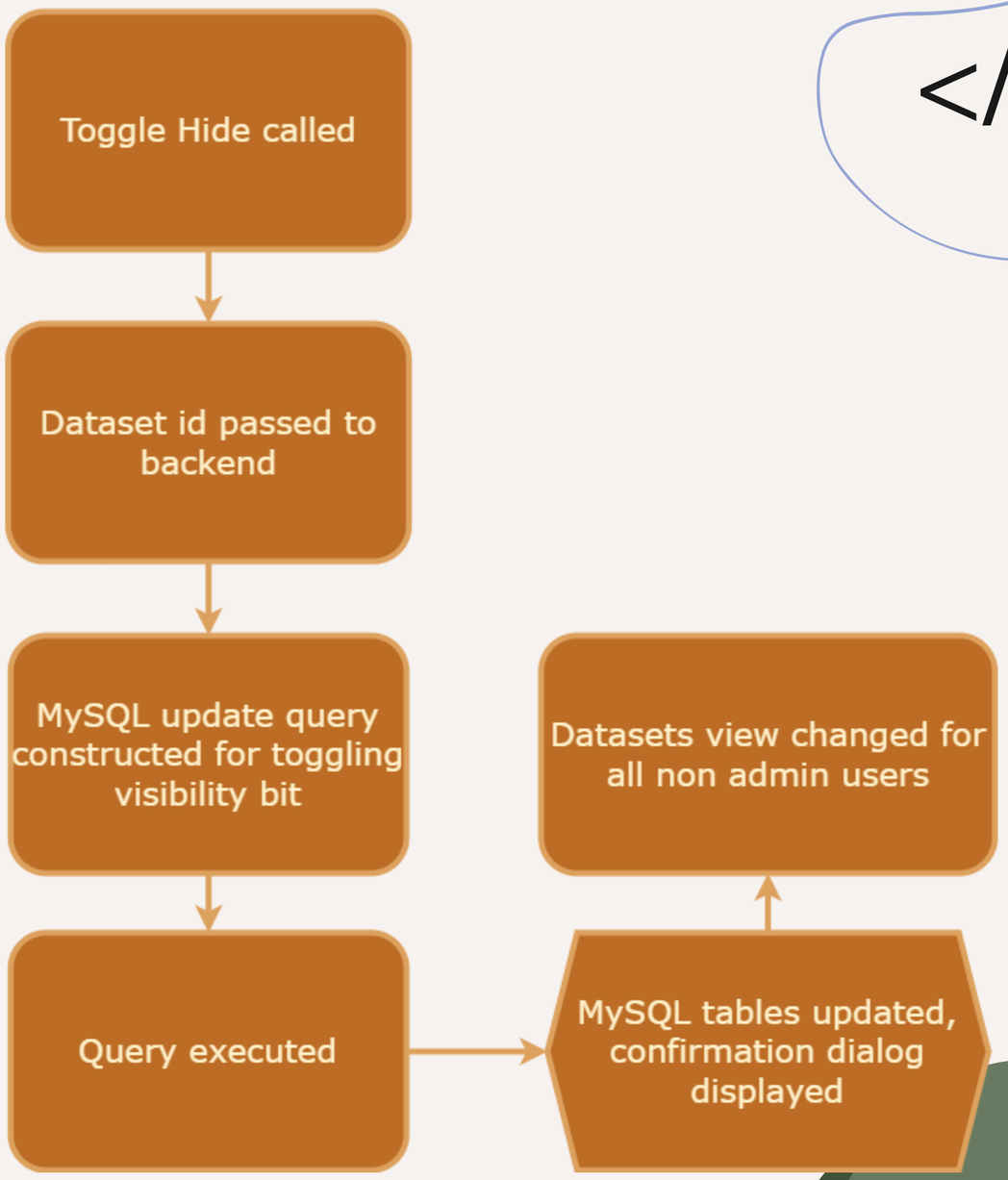
<Activation Diagrams>

process overview for deletion of dataset / model



</Activation Diagrams>

process overview
for hiding
functionality of
dataset / model



<Project Timeline>

September:
Project Acquisition, developing a plan and workflow idea

October:
Initial implementation of JSON based dataset ingestion from scratch

November week 1:
Mid Eval and extending former system to Data Foundation website

</Project Timeline>

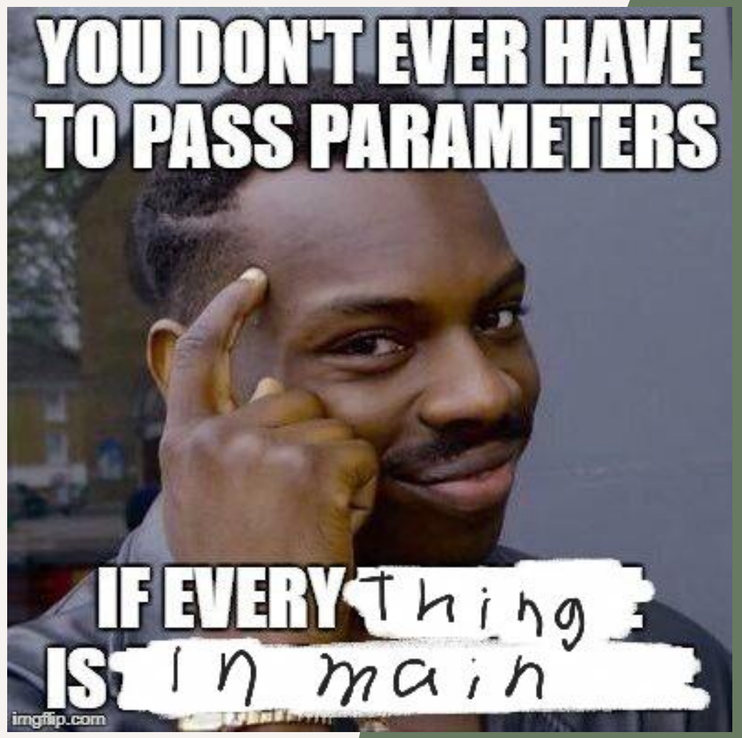
November Week 2: Adding functionality for JSON based version ingestion. Extending system to work for models

November Week 3-4: Adding functionality for hiding/deleting dataset. Adding file browser for JSON selection

December Week 1: Finishing touches, unit tests, cleaning up code and ensuring robust functioning of code

System architecture and code

- The system has been built completely on the existing platform of data foundation systems, indicating seamless architectural integrity
- Perfectly integrable, easy to read and understand methods
- Code involves modularisation and heavy usage of functions in order to have similar workflow for models as for datasets



Future Plans

- Adding functionality for admin to edit dataset after the dataset has been uploaded
- Dataset integrity verification algorithms
- Fixing dataset and versions mismatch between public and private setting, causing issues
- Currently focused on functionality, can improve interface and frontend of added pathways
- MinIO integration

Understanding code workflow

database structures

Deployment

Learning aspect

FUN!!

Pipelining of JS apps with databases

Importance of clean code

Modularity

Thank you.