Brief

We have created a python dash application with a number of screens. We have developed 3 screens so far and we now want to create a new screen called *Process* which will consist of a *KANBAN* style dashboard *(see diagrams below)* with dynamically generated cards and a database tracking changes in these card properties.

Just to stress – we require drag & drop functionality for the UI of the Kanban.

Requirements

- KANBAN dashboard with 9 lanes (swim lanes)
- Each lane is to be titled:
    - *Ideas, Correction of Errors Report, Short Note, Q&A, Model, Pre Mortem, Full Note, Buy List, Fail List*
- The first lane must have an add feature on the right of the title which generates a card template *(see diagram below)*
- These cards can be dragged and dropped
- An unlimited number of cards can be added and there should be a vertical scrollbar when a lane is filled such that the number of cards extend past the visible viewport height
- Once a new card is added in the Idea lane there should be certain fields which are able to be filled out:
    - Stock Name
    - Created date
    - Due date
    - Primary Analyst
    - Secondary Analyst
    - Attachments
        - There are 6 possible attachment types for a given stock
        - Attachments should be collapsible and viewed in a vertical list order when toggled
- Once a card is created ***only*** the attachments and secondary analyst tab should be amendable in the future (all other fields should be greyed out)
- Specific updates and changes to a card should be recorded as an entry in a DB
- Here are the events that cause an entry to be logged to the DB:
    - Creation of new card
    - Dragging and dropping into a new lane
    - Adding attachments
    - Changing secondary analyst name
- The following fields are to be recorded
    - Stock_name
    - Entry_date (datetime the **entry** was created)

  - Due_date
  - Type (this will be ‘New Ideas’ for all entries)
  - Stage (this will be the lane the card is currently in)
  - Primary_analyst
  - Secondary_analyst
  - Sedol (random 8 digit int)
  - ISIN (random 8 digit int)
  - Link_1 (all files uploaded that include the term ‘Full Note’ in them)
  - Link_2 (all files uploaded that include the term ‘Pre Mortem’ in them)
  - Link_3 (all files uploaded that include the term ‘Q&A’ in them)
  - Link_4 (all files uploaded that include the term ‘Short Note’ in them)
  - Link_5 (all files uploaded that include the term ‘Model’ in them)
  - Other
- When the app is refreshed I want the most recent entry_date for each stock to be the information used to populate the cards
- Please make the colour scheme resemble the one in the diagrams below as closely as possible

----

Note:

- I have attached an example csv file to show how the database will look like with example entries.
- If possible use SQLite db as this works well for testing purposes when using python

Resources

The following git repo provides access to an open source KANBAN dashboard built in Python / JS which could be of assistance: https://github.com/AlexanderWillner/KanbanView

Ideally this will be coded in python using dash-bootstrap-components as the main framework, however if external libraries / packages / frameworks are required then please consult so we can decide this going forward

Source Control

Please can we use a GitHub / GitLab repo to centralise all work so it can easily be pulled and merged. This will be more easier than having to manually extract files from .zip files!

Proposed approach

To create a KANBAN-style drag and drop dashboard with card movement tracking in your Python Dash app, it is recommended to use JavaScript libraries as they provide more robust support for interactive UI components like drag and drop functionality. While Dash does not have built-in support for drag and drop interactions, you can integrate JavaScript libraries seamlessly into your Dash application.

Here's a suggested approach to integrating JavaScript into your existing Dash app:

1. Choose a JavaScript library: There are several popular JavaScript libraries available for implementing drag and drop functionality, such as SortableJS, Draggable, or React DnD. These libraries provide the necessary features to create draggable cards and handle card movements.

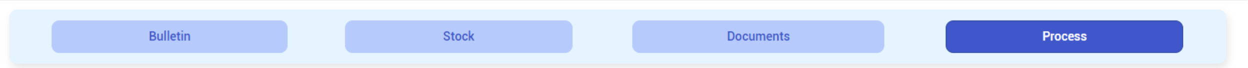
2. Set up your JavaScript environment: You can include JavaScript code in your Dash app by utilizing the `assets` directory. Create a new folder called `assets` in the root directory of your Dash app and include the necessary JavaScript library files and any custom JavaScript files you need. You can download the library files from their respective websites or use package managers like npm or yarn.

3. Define the KANBAN board layout in your Dash app: Create the layout for your KANBAN board using the components provided by Dash. This layout will contain the draggable cards and the different lanes of the KANBAN board. You can use Dash's `html.Div` or `dbc.Card` components to represent the cards and the lanes.

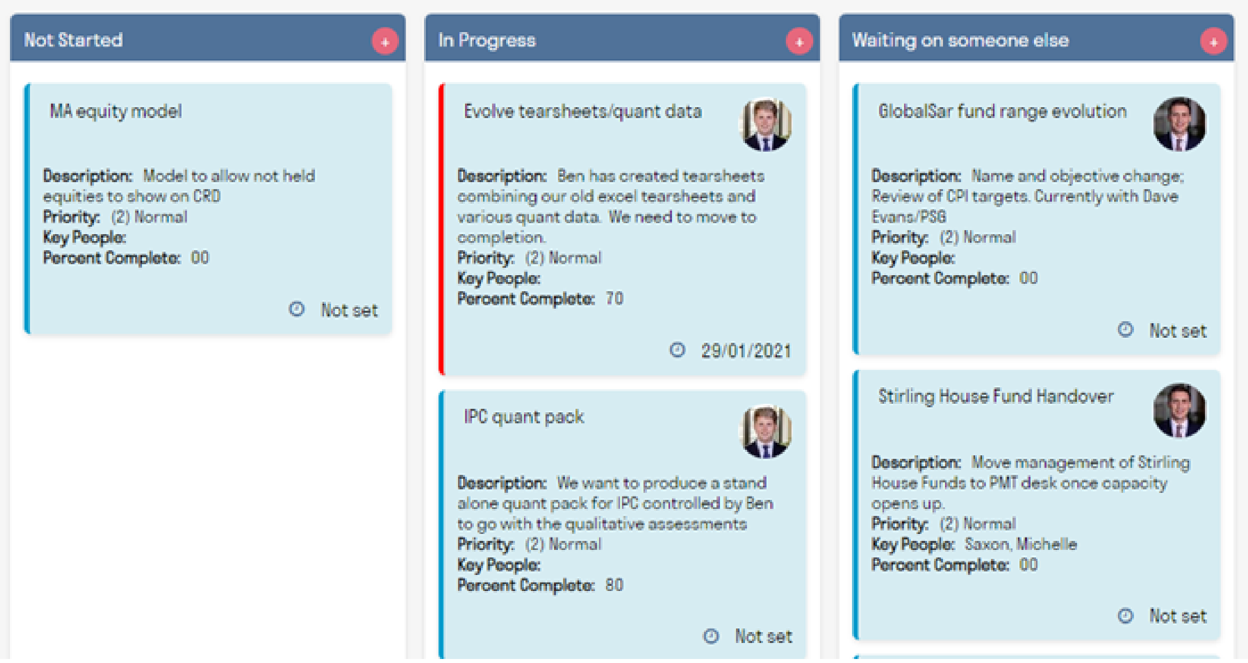
4. Implement the JavaScript code: Write the JavaScript code that initializes the drag and drop functionality and handles card movements. You can create a new JavaScript file or include the code directly in the Dash app's main HTML file. Refer to the documentation of your chosen JavaScript library for specific instructions on how to implement the drag and drop functionality.

5. Communicate with the server and SQL database: To track the movement of cards within the dashboard and store the data in a SQL

App Navbar



KANBAN style dashboard examples



Not Started

MA equity model

Description: Model to allow not held equities to show on CRD
Priority: (2) Normal
Key People:
Percent Complete: 00

Not set

In Progress

Evolve tearsheets/quant data

Description: Ben has created tearsheets combining our old excel tearsheets and various quant data. We need to move to completion.
Priority: (2) Normal
Key People:
Percent Complete: 70

29/01/2021

IPC quant pack

Description: We want to produce a stand alone quant pack for IPC controlled by Ben to go with the qualitative assessments
Priority: (2) Normal
Key People:
Percent Complete: 80

Not set

Waiting on someone else

GlobalSar fund range evolution

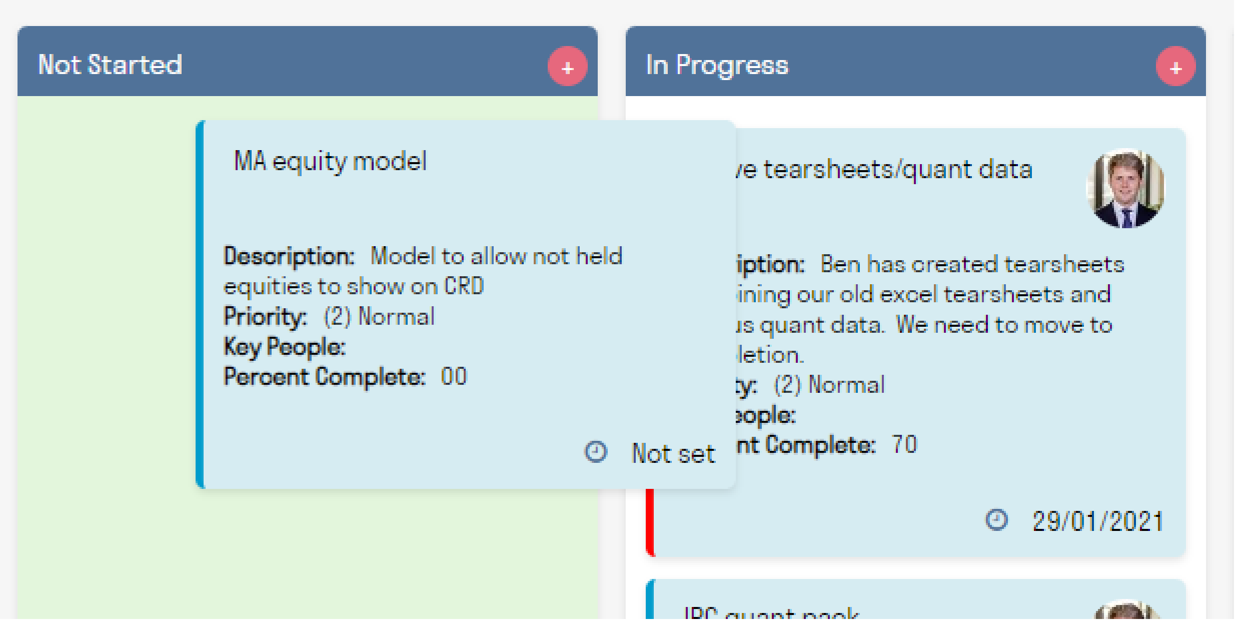
Description: Name and objective change; Review of CPI targets. Currently with Dave Evans/PSG
Priority: (2) Normal
Key People:
Percent Complete: 00

Not set

Stirling House Fund Handover

Description: Move management of Stirling House Funds to PMT desk once capacity opens up.
Priority: (2) Normal
Key People: Saxon, Michelle
Percent Complete: 00

Not set



Not Started

MA equity model

Description: Model to allow not held equities to show on CRD
Priority: (2) Normal
Key People:
Percent Complete: 00

Not set

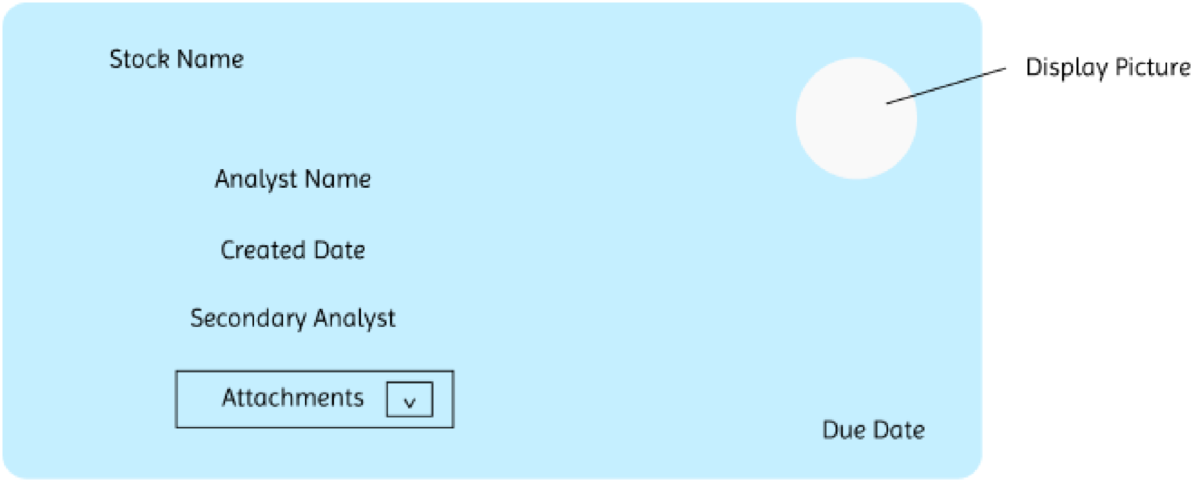
In Progress

ve tearsheets/quant data

iption: Ben has created tearsheets
ining our old excel tearsheets and
us quant data. We need to move to
letion.
ty: (2) Normal
eople:
nt Complete: 70

29/01/2021

IPC quant pack


Stock Name
Display Picture
Analyst Name
Created Date
Secondary Analyst
Attachments
v
Due Date