**Use case 1:**

|  |  |
| --- | --- |
| Use case | **Organization Catalog** |
| Use case description | The catalog displays the details of the account, project, resource & technology. |
| Batch jobs | There will daily delta load to insert / update / delete the account, project, resource & technology. |
| Service | To retrieve the account, project, resource & technology details from database. |
| UI | Displays the chosen details with the combination. |

**Batch Jobs:**

1. Account details job –Python program
   1. File type: CSV
   2. File name: Account\_Details.csv
   3. File format: ACCOUNT\_ID|ACCOUNT\_NAME|MODE
   4. Mode will be
      1. I - Insert
      2. U – Update
      3. D – Delete
   5. Based on the mode, the details to be inserted to ACCOUNT\_DETAIL table with current timestamp
   6. Table: ACCOUNT\_DETAIL
      1. ACCOUNT\_ID – Primary Key – Integer
      2. ACCOUNT\_NAME – String
   7. Exception handling: Invalid data & Missing data
   8. And loggers are required

Technology: Pandas, Multithreading (Each thread to process 3 files)

Total file count: 9

1. Project details job
   1. File type: CSV
   2. File name: Project\_Details.csv
   3. File format: ACCOUNT\_ID| PROJECT\_ID |PROJECT\_NAME|START\_DATE|END\_DATE|MODE
   4. Mode will be
      1. I - Insert
      2. U – Update
      3. D – Delete
   5. Based on the mode, the details to be inserted to PROJECT\_DETAIL table with current timestamp
   6. Table: PROJECT\_DETAIL
      1. PROJECT\_ID – Primary Key – Integer
      2. ACCOUNT\_ID – Integer
      3. PROJECT\_NAME – String
      4. START\_DATE – Date
      5. END\_DATE - Date
   7. Exception handling: Invalid data & Missing data

Technology: Pandas, Multithreading (Each thread to process 5 files)

Total file count: 25

1. Resource details job
   1. File type: CSV
   2. File name: Resource\_Details.csv
   3. File format: PROJECT\_ID | RESOURCE\_ID | RESOURCE \_NAME|TECHNOLOGY|DOJ|MODE
   4. Mode will be
      1. I - Insert
      2. U – Update
      3. D – Delete
   5. Based on the mode, the details to be inserted to RESOURCE \_DETAIL table with current timestamp
   6. Table: PROJECT\_DETAIL
      1. RESOURCE\_ID – Primary Key – Integer
      2. PROJECT\_ID – Integer
      3. RESOURCE \_NAME – String
      4. TECHNOLOGY – String Max(24)
      5. DOJ - Date
   7. Exception handling: Invalid data & Missing data

Technology: Pandas, Multithreading (Each thread to process 5 files)

Total file count: 25

**Use Case 2:**

**Service: REST\_API**

1. Account service
   1. Retrieve account details from ACCOUNT\_DETAIL
2. Project service
   1. Retrieve project details based on account id from PROJECT\_DETAIL
3. Resource service
   1. Retrieve resource details based on project id from RESOURCE\_DETAIL & PROJECT\_DETAIL
   2. Calculation
      1. Project closure remaining days = Project end date - Current date
      2. Resource project experience = Current date - DOJ

**Combine the 3 table information and write it to a csv/txt file.**

Technology: Django

Use Case 3:

Write a python program to call existing REST API and fetch the data to write it to a file.

Details will be shared later.