- Sage Cloud
    - Managing Authentication: Authorization request web page
    - Managing Authentication: User Authentication redirect + scope request
    - Managing Authentication: Authentication call back
    - Managing Authentication: Exchange of authorization code
    - Managing Authentication: Credential access validation
    - Managing Authentication: Auth credential Storage
    - Managing Authentication: Refresh Token
    - Managing Authentication: Token Expiry Alert
    - Managing Authentication: Manage multi-tenancy security
    - Data Retrieval: Retrieve the auth credentials
    - Data Retrieval: Handle API connection
    - Data Retrieval: Handle Rate Limit
    - Data Retrieval (Backfill): Identify time window
    - Data Retrieval (Backfill): Retrieve data
    - Data Retrieval (Incremental): Identify deltas
    - Data Retrieval (Incremental): Retrieve data
    - Data Retrieval: Storage of flat files
    - Data Transformation: Retrieve flat file
    - Data Transformation: Identify & Validate the file format
    - Data Transformation: Transformation of data into standard schema
    - Data Loading: Manage multi-tenancy databases
    - Data Loading: Load into database staging tables
    - Data Loading: Merge staging tables into live
    - Data Loading: Post-import batch processing

Epic

➢ Sage Cloud

User Story ID	User Story
US01	Managing Authentication: Authorization request web page

As a user, I need the ‘Integrate’ button on the integration page. So that I can be prompted to sign-in.

Acceptance Criteria:

- In the Integration page, users should be able to select an option of ‘connect to Sage Cloud platform’ under the financial pack section.
- On click, the user will be prompted to the login page of Sage Cloud.
-

As a system, I need to send a request to the Sage server. So that user will be prompted to login page and authenticate the credential..

Acceptance criteria:

- If the user clicks the button to connect, the system must pass a request to the Sage server.
- Request must contain the scope of access for the specific user’s data.

Note:

User Story ID	User Story
US02	Managing Authentication: User Authentication redirect + scope request

As a user, I want to be directed to Sage Cloud so I can enter my credentials.

Acceptance Criteria:

- Users must enter the valid credential of the Sage account.
- After the successful authentication, the user is redirected to authorize the app's scope (Grant Authorization)
- User must grant access to the PowerAnalysis platform.
- After granting, user should be redirected to the integration page of the PowerAnalysis platform.

Comment: Sage does not support any other login methods (such as Google, Microsoft, etc)

User Story ID	User Story
US03	Managing Authentication: Authentication call back

As a system, I should receive a response with an authorization code. So that I can move ahead with access code generation.

Acceptance criteria:

- If the user has successfully authenticated and granted the access, the Sage server should send a response.
- Response must be received on the callback URL provided in the Authorization request.
- Response must contain an authorization code.

Note:

User Story ID	User Story
US04	Managing Authentication: Exchange of authorization code

As a system, I need to exchange the authorization code. So that I can receive the access token.

Acceptance criteria:

- Developer need to attach this authorization code in the request URL and pass it to the Sage server
- Server will verify the code and respond with an access token and refresh token.
- Both the tokens must get stored in the credential database.

Notes:

- Authorization code was received in callback response.

User Story ID	User Story
US05	Managing Authentication: Credential access validation

As a system, I need to validate the user's credentials used while authentication. So that I can confirm it is able to access the data for continuous data retrieval.

Acceptance Criteria:

- Validate the authenticity of credentials provided by the user.
- Validate the credentials for data accessibility from Sage Cloud.
- After validation, the credentials will get stored in the credential database.
- Credentials should be mapped to userID/token of the user.

Note:

User Story ID	User Story
US06	Managing Authentication: Auth credential Storage

As a system, I need to store the user’s credential to retrieve the incremental data from the source. So that system can automate future authentication.

Acceptance criteria:

- After the successful authentication of the user, collect the auth credentials of the user.
- Store the credentials in the credential database.
- Each user’s credential should be mapped by their userID/token.

Note:

- Create a Credential object containing a user's sign-in information.

User Story ID	User Story
US07	Managing Authentication: Refresh Token

As a system, I need to refresh the access token.So that I can continue pulling the data from the data source.

Acceptance criteria:

- If the access token expires, then a refresh token must be used to generate an access token.
- At the time of data retrieval, the system needs to generate the access token i.e. once in 24 hour.
- callback response will contain the access token and new refresh token

  Note: Each token has a life of 5 minutes & a refresh token can be used only once to obtain a new access token.

  Refresh token usable for 30 days.

Note:

- Token should get stored in the database.

User Story ID	User Story
US08	Managing Authentication: Token Expiry Alert

As a system, I need to get an alert for the expiring refresh token. So that I can be prepared to obtain a new refresh token.

Acceptance criteria:

- Each token has a life of 5 minutes, at the end of the time the system would get an alert of expiration of the access token.
- Each alert must trigger the system to send the request for generation of new access token.

Note:

User Story ID	User Story
US08	Managing Authentication: Manage multi-tenancy security

As a system, I need to manage multi-tenancy security. So that we can secure the tenant's data and privacy.

Acceptance criteria:

- Use only a verified email address to authorise user access to a tenant based on domain match.
- Ensure isolation of credentials and the storage of data for individual tenants.
- Use 1:1 mapping between external Identity Platform and application client to prevent unauthorised cross-tenant access.

Note:

User Story ID	User Story
US09	Data Retrieval: Retrieve the auth credentials

As a system, I need to retrieve the stored user credential. So that I can authenticate and retrieve the data from Sage Cloud.

Acceptance criteria:

- Request the credential from the stored location in the database.
- Use the retrieved credential for the authentication of Sage Cloud.
- This process should get triggered automatically when the system runs the incremental update.

Note:

- Credential retrieval should be done without the user's intervention.

User Story ID	User Story
US010	Data Retrieval: Handle API connection

As a system, I need to connect the API. So that I can import the data from Sage Cloud.

Acceptance criteria:

- Send a request to Sage server for connectivity
- Along with the request, the system must send an access token.
- Sage server will validate the access token before connecting the API for retrieval

Note:

User Story ID	User Story
US011	Data Retrieval: Handle Rate Limit

As a system, I need to apply the rate limit of the Sage platform. So that I can consider the limit while retrieving data.

Acceptance criteria:

- System must follow the rate limit while retrieving data from the Sage cloud.
- System must not exceed the rate limits while extraction.

Note:

- Rate limit is set by the Sage Cloud to restrict the transfer of data from their platform.
- Rate limit of 1,296,000 requests per app per day
- Maximum of 150 concurrent requests at any time (per app)
- In case of limit exceeded, the Sage server API will return a HTTP 429 error response

User Story ID	User Story
US012	Data Retrieval (Backfill): Identify time window

As a user, I should provide the time period of data to be retrieved. So that I can review the analysis report as of expected date range.

Acceptance criteria:

- In the integration page, after the successful authentication user will be provided with an option of selecting the date range (customizable option).
- Once the user selects the date range, there should be a button of ‘Import Data’.
- After the user clicks on the button, the system should pass the same date range to retrieve the data from the source.

Note:

- This process will be a one time process as it will fetch the past data of the user.

User Story ID	User Story
US013	Data Retrieval (Backfill): Retrieve data

As a system, I need to retrieve the data. So that I can transform the data into a standard schema.

Acceptance criteria:

- After getting an input from the user, data retrieval must be done based on the time range provided.
- Retrieve the historical data from the user's Sage Cloud account.

Note:

User Story ID	User Story
US014	Data Retrieval (Incremental): Identify deltas

As a system, I need to retrieve the updated data from the source in the 24 hours interval. So that I can store and transform it.

Acceptance criteria:

- Data change detection - Retrieval of only changed & new data.
- For identifying delta columns, we can assume the column of timestamp can be used for comparison.
- The column with a new/latest timestamp can be retrieved from the database.

Example:

```
"updated_at": "2018-06-25T05:46:18Z"
```

https://developer.Sage.com/accounting/guides/concepts/invoicing/

Note:

User Story ID	User Story
US015	Data Retrieval (Incremental): Retrieve data

As a system, I need to retrieve the updated data from the source in the 24 hours interval. So that I can store and transform it.

Acceptance criteria:

- Use the last extracted date & timestamp to compare with the present date & timestamp in the data source.
- If the date & timestamp does not match, consider the data as updated.
- Retrieve the data from the source
- Pass the check for duplication in the row.

Note:

User Story ID	User Story
US016	Data Retrieval: Storage of flat files

As a system, I need to save retrieved data as a flat file for a temporary basis.

Acceptance criteria:

- The extracted file should be directly imported into the storage location in the form of flat file(s).
- System has to run a command for loading the flat files in the storage.
- Access path of the file should be like <client>/<integration>/<year>/<month>/<date>
- File name should be named as current date in format <filecontents>_<hhmmss>.type.

Note:

User Story ID	User Story
US017	Data Transformation: Retrieve flat file

As a system, I need to retrieve data from a storage area to start the process of transformation.

Acceptance criteria:

- System will run the command to retrieve the flat file from the storage area.
- Retrieval of the flat file will be processed in the batches.
- The file in the storage will be retrieved as the file will be imported after the extraction.

Note:

User Story ID	User Story
US018	Data Transformation: Identify & Validate the file format

As a system, I need to retrieve data from a storage area to start the process of transformation.

Acceptance criteria:

- Identify the row delimiters (can be anything comma, tabs, tube (|), etc)
- Validate the empty or corrupted values in the file.
- If present the null/empty/corrupted values encountered, skip the value and continue with the further process.

Note:

User Story ID	User Story
US019	Data Transformation: Transformation of data into standard schema

As a system, I need to transform the data into a defined format. So that system can store it in the database;.

Acceptance criteria:

- Enter Input flat file
- Flat file transform to uniform format
- Write an algorithm/procedure to convert the extracted data into defined format.
- Convert the file into the format and save it as defined file type.
- File name should be named as YYYY_MM_DD

Note:

User Story ID	User Story
US020	Data Loading: Manage multi-tenancy databases

As a system, I need to set up a multi-tenancy database as it is more secure and easy to map the data.

Acceptance criteria:

- Each user will have their own tenants. .
- Create a folder inside each tenant by user's token.
- Create a folder inside the above folder by name 'YYYY'
- Create a folder inside the above folder by name 'Month'
- Insert each transformed file in the month folder by the name 'YYYY_MM_DD'.

Note:

User Story ID	User Story
US021	Data Loading: Load into database staging tables

As a system, I need to load the transformed data in the staging area. So that I can transfer it in the live database.

Acceptance criteria:

- System must take backups of the database.
- System must insert all the transformed data into the staging area.
- Once system completes a successful import on a set of records from one or more staging tables
- System must set the import process to run automatically on a schedule.

Note:

User Story ID	User Story
US022	Data Loading: Merge staging tables into live

As a system, I need to merge the transformed data in the original database. So that analysis process can begin.

Acceptance criteria:

- Create a staging table and populate it with updated rows from table
- Start a new transaction
- Delete any rows from table that already exist
- Insert all the rows from the staging table into the target table
- End transaction and commit
- Drop the staging table

Note:

User Story ID	User Story
US023	Data Loading: Post-import batch processing

As a system, I need to start the import process again automatically.

Acceptance criteria:

- Navigate to Batch server > execute a query
- Run the code for import.
- Store the extracted files.
- Transform it as per the standard schema and load it in the database
- Above process shall be implemented as a batch.
- Automatic code should be implemented to run each batch.

Note: