|  |  |  |
| --- | --- | --- |
|  |  | User Requirement Analysis  Google Sheet Into DB |

May 25, 2016

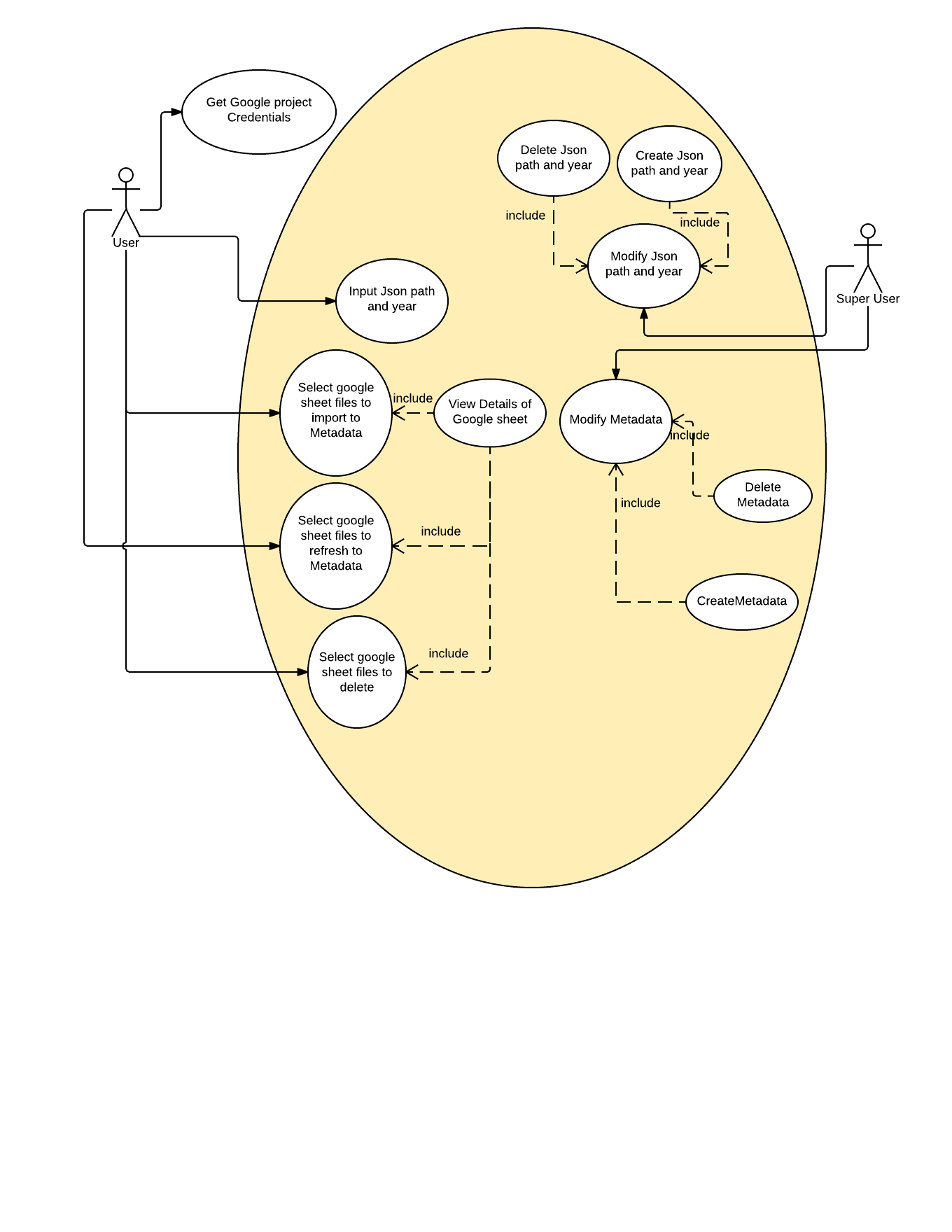
# Purpose

This open source project is for G2F collaborators to have a web interface to exchange their experiment data in google sheet, so it would be easy for the collaborators retrieve all the data from a database with easy to use web interface. We would like to use Django framework to put the data into a Postgresql database since Django and Postgresql are all open source and Django is easy to create web interfaces.



## User diagram

The user diagram is created through an open source software design tool [www.lucidart.com](http://www.lucidart.com). Here is the link for the user diagram: <https://www.lucidchart.com/documents/edit/91daddcb-de2d-4d87-8844-cae0189c6eb4>



## Requirement analysis

## Through the user diagram, the user will input the JSON file path for Google project credentials and trial year for the experiment. The JSON file is required for accessing the google sheet through Python. The user also need to check the contents of each google sheet, import them into database, refresh them or delete them. The user also need to add select all in both refresh and import data so that he/she can save some time since there could be more than 40 google sheet to deal with.

## Web page design

<https://www.lucidchart.com/documents/edit/384e09f3-3407-4e02-a777-2c4ef88198d0/4>



<https://www.lucidchart.com/documents/edit/384e09f3-3407-4e02-a777-2c4ef88198d0/5>

## Database design

Django database design followed the tutorial of Django, only the Postgresql column names in table cannot contain either space or “/”, so I replaced space with “\_” , “/” with “or” in the column names from google sheet to column names of Metadata table. The Json\_path table is needed because JSON file path and trial year are inputted by user and it might change next year. So the program only get the last row value in the table to keep the data and trial year matching perfectly.

## Data structure and methods design

To make the data\_index.html showing as it designed, it need a new dictionary design to contain all the information the design needed.   
single\_object={**'title'**:googleSheet.title, **'id'**:googleSheet.id,**'imported\_status'**:**'F'**,**'imported\_date'**: **''**,**'import\_data'**: **'T'**,**'refresh\_data'**: **'F'**}

Also, it needs a method to compare the two lists from google and from database to give the single\_objects in an array certain Boolean or date values.

**def** get\_googleSheet\_list(list\_from\_google, list\_from\_db)

Certain methods and lists are created in the views.py so that these data can exchange freely between the method in views.py.

## Reference:

(Rajaraman, 2016)