

# POLITECNICO DI TORINO

Master Degree Course in Computer Engineering

Master Degree Thesis

Design, development and test of a graphical user interface for visualizing data quality measurements on Italian Open Government Data



**Politecnico  
di Torino**

## **Supervisors**

prof. Antonio Vetrò

## **Co-Supervisors**

prof. Marco Torchiano

## **Candidates**

Siqi CAI

matricola: 273209

**Anno accademico 2021-2022**

# Contents

Abstract .....	5
1. Introduction to data quality tool.....	6
1.1 Main function .....	6
1.2 Industry status.....	7
1.3 ISO/IEC 25012:2008.....	7
1.4 ISO/IEC 25024:2015.....	8
2. Requirement analysis.....	10
2.1 User stories.....	10
2.1.1 Visitor.....	10
2.1.2 User .....	11
2.1.3 Administrator .....	12
2.2 Application design .....	12
2.2.1 Approve .....	13
2.2.2 Manager User .....	13
2.2.3 Notice .....	14
2.2.4 Account Setting.....	14
2.2.5 Manage Dataset.....	15
2.2.6 Analysis Setup .....	16
2.2.7 Saved Result .....	18
3. Technology stack .....	19
3.1. Backend framework, Flask .....	20
3.2. Frontend framework, Vue .....	26
3.3 ORM framework, SQLAlchemy .....	28
3.4 Docker, docker-compose .....	29
4. Implementation .....	32
4.1 Backend API .....	32
4.4.1 Login .....	32
4.4.2 Logout.....	32
4.4.3 Get current logged user information .....	32
4.4.4 Create a new account.....	33
4.4.5 Update a account .....	33
4.4.6 Delete a account.....	33
4.4.7 Get a list of users in pagination.....	34
4.4.8 Apply for access as User .....	34
4.4.9 Approve the application of Visitor .....	34
4.4.10 Get a list of applications in pagination .....	34
4.4.11 Get a list of notices in pagination .....	35

4.4.12 Upload files .....	35
4.4.13 Batch upload files.....	35
4.4.14 Download dataset files .....	35
4.4.15 Get a list of files in pagination.....	36
4.4.16 Delete dataset files.....	36
4.4.17 Save a analysis setup.....	36
4.4.18 Get the detail of a saved analysis setup.....	37
4.4.19 Delete a analysis setup .....	37
4.4.20 Get a list of saved analysis setups in pagination .....	37
4.4.21 Do analysis.....	37
4.4.22 Save an analysis result .....	38
4.4.23 Rename a saved analysis result .....	38
4.4.24 Get the detail of a saved analysis result.....	39
4.4.25 Get a list of saved analysis results in pagination .....	39
4.4.26 Delete a saved analysis result.....	39
4.4.27 Export a result as a csv file .....	39
4.4.28 Export a setup as a csv file .....	40
4.2 Frontend Router .....	40
4.2.1 Guide .....	40
4.2.2 Account Setting.....	40
4.2.3 Analysis Setup.....	40
4.2.4 Approve .....	40
4.2.5 Manage Dataset.....	41
4.2.6 Manage User .....	41
4.2.7 Notice .....	41
4.2.8 Saved Result .....	41
4.2.9 Login .....	41
4.3 Database .....	41
4.4 Introduction to application .....	44
4.4.1 Login .....	44
4.4.2 Notice .....	44
4.4.3 Manage User .....	45
4.4.4 Approve .....	46
4.4.5 Account Setting.....	47
4.4.6 Manage Dataset.....	47
4.4.7 Analysis Setup.....	48
4.4.8 Analysis Result .....	49
4.5 Test case.....	50

4.5.1 What is open data .....	50
4.5.2 Example of measurements on open data .....	51
5. Conclusions .....	59
5.1 Conclusion .....	59
5.2 Future Implementation .....	59
List of references .....	61

# Abstract

Today with the development of the Internet, data is used in more and more places, such as training the model and data analysis for businesses and governments to improve their services or gain benefits. With more and more data, it is difficult for people to monitor the quality and accuracy of all data. Poor quality data can lead to some negative results. The aim of the thesis is to design and develop a web-application to do data quality analysis based on a generalized algorithm provided by Davide Vitaletti, and display the analysis results visually for users. The application includes back-end and front-end. The back-end is developed based on the web framework, Flask and ORM framework, SQLAlchemy. The front-end is developed based on the framework, Vue. Vue is an easy to learn and for rapid development framework. Based on the different account role, the users have different permissions. The main modules of the application include account management, file management, analysis setup, analysis result. Its main function is to analysis the data on Italian Open Government Data for users to visualize data quality measurements including Com-I-1 DevA, Com-I-5, Acc-I-4, Con-I-3, Con-I-2 DevB and Con-I-4 DevC. Thesis quality measurements are defined in ISO/IEC 25024:2015.

# 1. Introduction to data quality tool

## 1.1 Main function

Data quality [1] is a measure of the state of data based on factors such as accuracy, completeness, consistency, reliability, and whether the data is up-to-date. Measuring data quality levels can help organizations identify data errors that need to be addressed and assess whether the data in their IT systems is fit for their intended use.

As data processing becomes more complex in relation to business operations, and the emphasis on data quality in enterprise systems increases, organizations are increasingly using data analytics to help drive business decisions. Data quality management is a core component of the overall data management process, and data quality improvement efforts are often closely tied to data governance programs designed to ensure that data is consistently formatted and used across the organization.

Bad data can have major business consequences for a company. Low-quality data is often cited as a source of operational confusion, inaccurate analytics, and poorly conceived business strategies. Examples of financial losses that can result from data quality issues include: increased fees when products are shipped to the wrong customer address, lost sales opportunities due to erroneous or incomplete customer records, and fines for improper financial or regulatory compliance reporting.

An oft-cited estimate by IBM calculates the annual cost of data quality problems in the U.S. at \$3.1 trillion in 2016. In a 2017 article for the MIT Sloan Management Review, data quality consultant Thomas Redman estimated that correcting data errors and dealing with bad data results in business Problems can reduce the company's revenue by 15% to 25% annually on average.

## 1.2 Industry status

Nowadays, lots of company have launched their products of data quality tool. For example, IBM launched several of products. Each product has specified functions, such as InfoSphere is to turn data to trusted information and monitor data quality and BigQuality provides a solution with a rich set of data profiling, cleaning ,and monitoring capabilities that execute on the data nodes of an Apache Hadoop cluster.

Informatica offers a modular MDM solution that provides a single view of data. The product enables users to create an authoritative view of business-critical data from disparate, duplicate and conflicting sources. Informatica MDM also features AI and machine learning, and includes data quality, data integration, business process management, and data security functionality that allows users to easily enrich master data records with data from external providers. Informatica's MDM capabilities can be deployed on-prem or in the cloud.

Infogix offers a suite of integrated data governance capabilities that include business glossaries, data cataloging, data lineage, and metadata management. The tool also provides customizable dashboards and zero-code workflows that adapt as each organizational data capability matures. Reference customers use Infogix for data governance and for risk, compliance and data value management. The product is also flexible and easy to use, and supports smaller data analysis jobs as well.

## 1.3 ISO/IEC 25012:2008

ISO / IEC 25012:2008 defines a general data quality model for data retained in a structured format in computer systems. It can be used to establish data quality requirements, define data quality measures or plan and perform data quality assessments. For example, it can be used to define and evaluate data quality requirements during data production, collection and integration, determining data quality assurance standards can also be used for data redesign, evaluation and improvement.

To Assess whether the data comply with legal and / or requirements. ISO / IEC 25012:2008 divides quality attributes into 15 characteristics, which are considered from two perspectives: inherent and system related. Data quality characteristics have different importance and priorities for different stakeholders.

Regarding the classification of data quality characteristics, this standard categorizes them from two points of view [2]:

- Inherent data quality, referring to the degree to which data quality characteristics have intrinsic potential to satisfy implicit data needs.
- System-dependent data quality, referring to the degree to which data quality is achieved and preserved through an information system and is dependent on the specific technological context in which the data is used.

Because the technical nature of data repositories varies greatly, it is almost impossible to develop common measures to allow comparison between different organizations. Therefore, the data quality assessment environment only considers the inherent data quality characteristics. By this way, the implementation of follow-up measures used in the evaluation can be essentially independent of the particularity of the data set and the technology of the information system supporting the data repository. Therefore, the process of generating data quality metrics is repeatable for any data set in any field, and the results can be compared and benchmarked. The inherent data quality characteristics are described in the following **Table 1**.

Characteristic	Definition
Accuracy	The degree to which the data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use.
Completeness	The degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use.
Consistency	The degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use.
Credibility	The degree to which data has attributes that are regarded as true and believable by users in a specific context of use.
Currentness	The degree to which data has attributes that are of the right age in a specific context of use.

Table 1. Inherent data quality characteristics defined in ISO/IEC 25012 [3]

## 1.4 ISO/IEC 25024:2015

On the Other hand, ISO/IEC 25024 – “Measure of data quality” provides measurements, including measurement methods and related quality measurement elements for the quality characteristics of the data quality model



described above. [2] A quality measurement is an element that represents a way of evaluating certain aspects or particularities of the data contained in a repository. To evaluate the quality of a data repository, an organization should identify the data quality characteristics and the corresponding data quality properties that best fit their stated data quality requirements. **Fig. 1** shows a summary of the inherent data quality characteristics and the data quality properties defined for each of them.

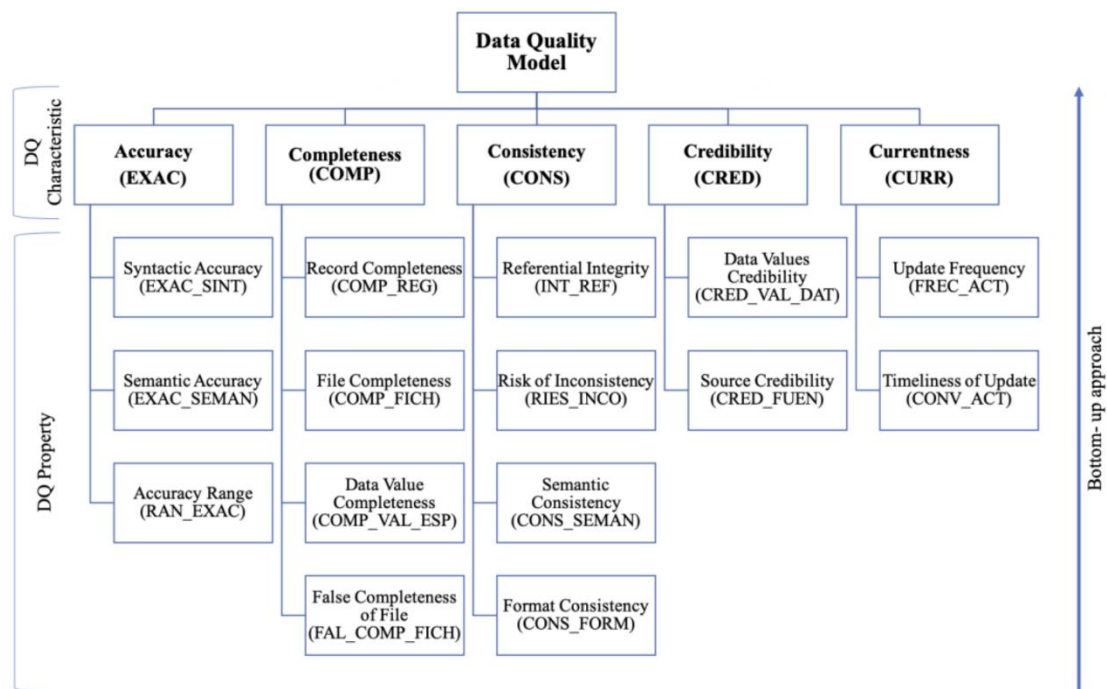


Fig. 1. Inherent data quality characteristics and related data quality properties extracted from ISO/IEC 25012 and ISO/IEC 25024, respectively.

**Table 2** shows an example of how each data quality attribute is described in ISO/IEC 25024 and the information it provides about how its value is calculated. The evaluation team must explain when the low value of the attribute measurement represents a problem in the data repository.

<b>Data quality characteristic</b>	Accuracy
<b>Data quality property</b>	Data Accuracy Range
<b>Measurement description</b>	The data accuracy range focuses on checking whether the data value is included in the required interval. Its value is the ratio of the field value in the data file to the records within the specified interval.
<b>Calculation formula</b>	$X=A/B$

	A= number of data items having a value included in a specified interval (i.e., range from minimum to maximum) B= number of data items for which can be defined a required interval of values
<b>Scale</b>	Ratio
<b>Value range</b>	[0, 1.0]

Table 2. Description for the property "Accuracy Range" (RAN\_EXAC) and its measurement [2]

## 2. Requirement analysis

### 2.1 User stories

User story is a tool used in software development to capture the description of software functionality from the end user's perspective. Users describe the type of users, what they want to do, and why they want to do it. A short description used to identify users and user needs. User story is often recorded in post-it notes and project management software.

User stories usually contain three elements:

1. <Role>: who uses
2. <Action>: what to finish
3. <Benefit>: why do it

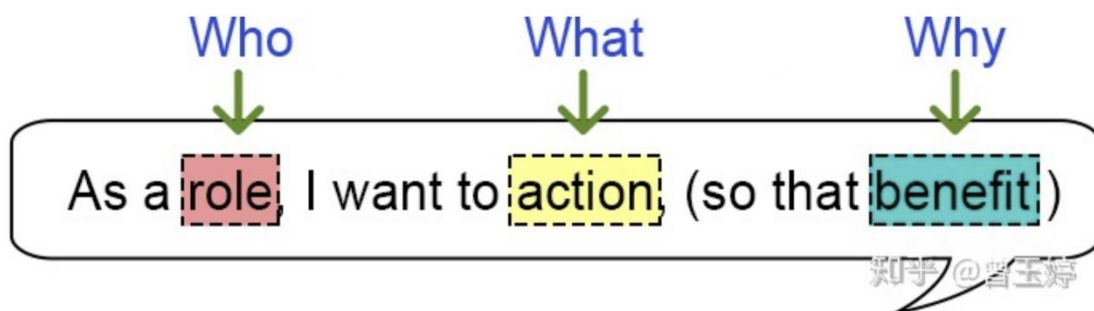


Fig 2. Expression of user story

Next the user stories will be described from the perspective of different users in the three different roles.

#### 2.1.1 Visitor

User story (Visitor)	Implemented
----------------------	-------------

V1. As a visitor, I want to see an example of analysis	✓
V2. As a visitor, I want to see an example of setup file	✓
V3. As a visitor, I want to get general information on the system	✓
V4. As a visitor, I want to request access as a user	✓

Table 3. User stories of visitor

From the **Table 3** above, it is said that a visitor has only limited operation in this application. It could only have a view of analysis result and the related files. It also could apply the access as a user.

### 2.1.2 User

User story (User)	Implemented
U1. As a user, I want to see a tutorial for using the system	✓
U2. As a user, I want to setup the analysis	✓
· specify which datasets to analyze	✓
· select which quality measures to apply to which datasets	✓
U3. As a user, I want to run an analysis with a specified default configuration	✓
U4. As a user, I want to view results of the analysis in a dashboard	✓
· by default, a graph (e.g., histogram with a bar for each quality metric) for each file/dataset is shown	✓
· a switcher will enable to show a measure-based visualization, i.e. a graph for each quality metric	✓
U5. As a user, I want to add/remove datasets to/from an existing analysis	✓
U6. As a user, I want to browse errors in the analysis	✓
U7. As a user, I want to know why an analysis setup is not allowed	✓
· example: exceeded limits (and which type), wrong format, etc.	✓
U8. As a user, I want to login to the system	✓
U9. As a user, I want to download the setup of the analysis	✓
U10. As a user, I want to save download the setup of the analysis	✓
U12. As a user, I want to delete a saved analysis setup	✓
U13. As a user, I want to see an overall plot of the analysis's results as first thing	✓
U14. As a user, I want to browse the results grouped by quality measure	✓
U15. As a user, I want to save the results of the analysis	✓

U17. As a user, I want to delete a saved analysis result	✓
U18. As a user, I want to be notified when changes to my account from a system administrator are made	✓
· account is created	✓
· account is deleted	✓
· analysis results deleted or renamed	✓
· analysis setup deleted or renamed	✓

Table 4. User stories of user

From the **Table 4** above, we can find a formal user has a lot of operation for analysis setup and analysis result, like save, update, delete, upload or download. It also could receive the notice in the application when the information of the account is changed.

### 2.1.3 Administrator

User story (Administrator)	Implemented
SA1. As a system administrator, I want to create/delete users	✓
SA2. As a system administrator, I want to browse and manage saved analyses	✓
· manage = rename, delete, download, upload, see much space it occupies	✓
· analyses = results and/or configurations	✓
SA3. As a system administrator, I want to set up limits for analyses	✓
· for all or selected users	✓
· in terms of number of datasets to analyze or overall size	✓

Table 5. User stories of administrator

From the **Table 5** above, we know the administrator is able to manage all roles of members in this application, including visitors, users and other administrators. It has the authority to add or change the information of the account and set a limitation for occupied space by each account.

## 2.2 Application design

From Fig 3, we can know that the overall function of the application is divided into the following modules.

1. Approve
2. Manage User

3. Notice
4. Account Setting
5. Manage Dataset
6. Analysis Setup
7. Saved Result

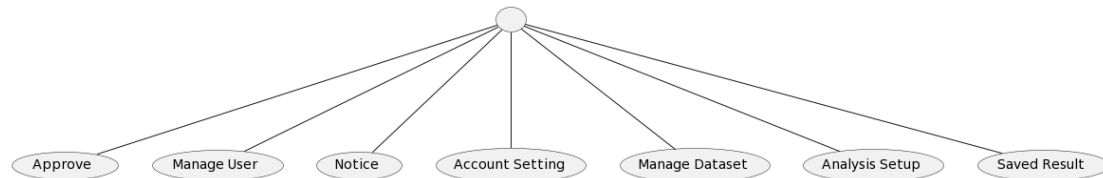


Fig 3. Overall application function menu

### 2.2.1 Approve

As user story said, there are three kinds of roles, **Visitor**, **User** and **Administrator**. The operation of Visitor is limited to only Notice, Account Setting and Saved Result. If one Visitor want to get access as a User, it must submit a application. Then **Administrator** could see all applications in Approve. It will show all applications by page and order by time. The latest application will be displayed in the front.

For each application, there are tow kinds of status, **in process** and **process end**. **In process** means that the application has not been approved. **Administrator** could choose agree or reject to approve it. **Process end** means that the application has been approved by **Administrator**. When the application is agreed, the applicant is granted permission and role of application is changed from **Visitor** to **User**. The function feature of Manage Dataset is shown in Fig 4.

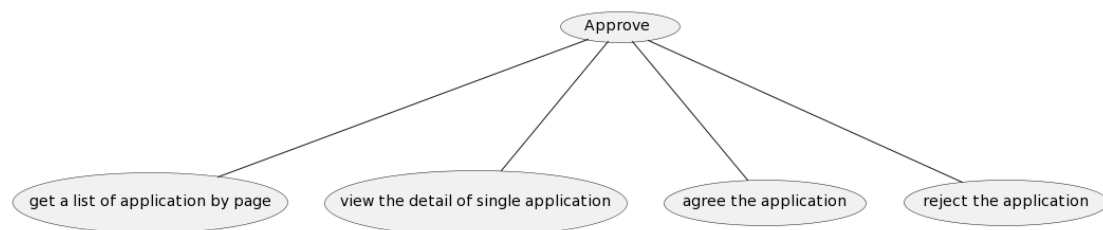


Fig 4. Function of Approve

### 2.2.2 Manage User

As user story said, **Administrator** could create/delete users. In **Manage User**, **Administrator** could manage all information of all accounts including the user name, password, role and the max space granted for dataset files. In previous function, we know a **Visitor** must submit the application for access as a **User**. In

**Manage User, Administrator** also could directly change the role of the account to grant it the permission. The function feature of Manage Dataset is shown in **Fig 5**.

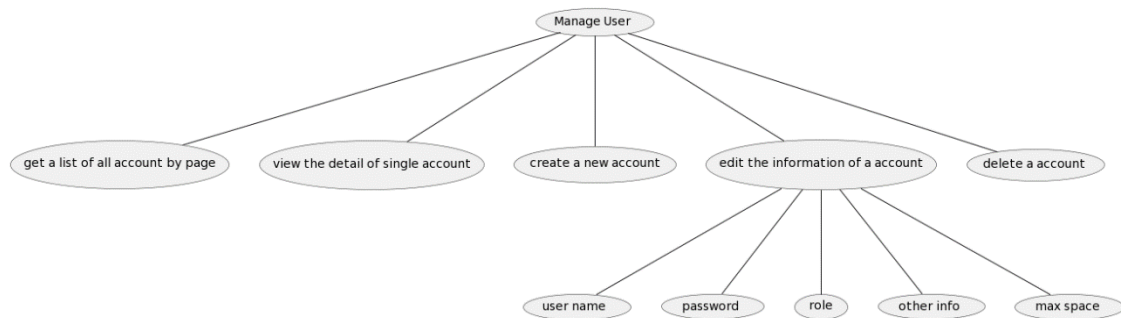


Fig 5. Function of Manager User

### 2.2.3 Notice

As user story said, the application user wants to get the notice when changes to the account. In **Notice**, they could see a list of notices by page. When their account information is changed or their saved analysis setup and saved analysis result are edited, they will receive the notice. The function feature of Manage Dataset is shown in **Fig 6**.

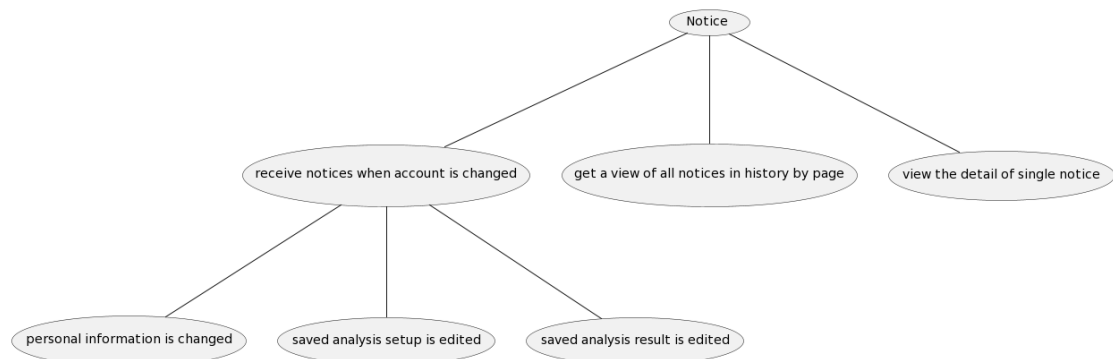


Fig 6. Function of Notice

### 2.2.4 Account Setting

In **Account Setting**, you can have a overall view of your personal information. Also as user story said, **Visitor** could request the access as **User**. So there is a button '**Request user access**' displayed when the role of current logged in user is **Visitor**. They could click this button to submit application for access as **User**. The function feature of Manage Dataset is shown in **Fig 7**.

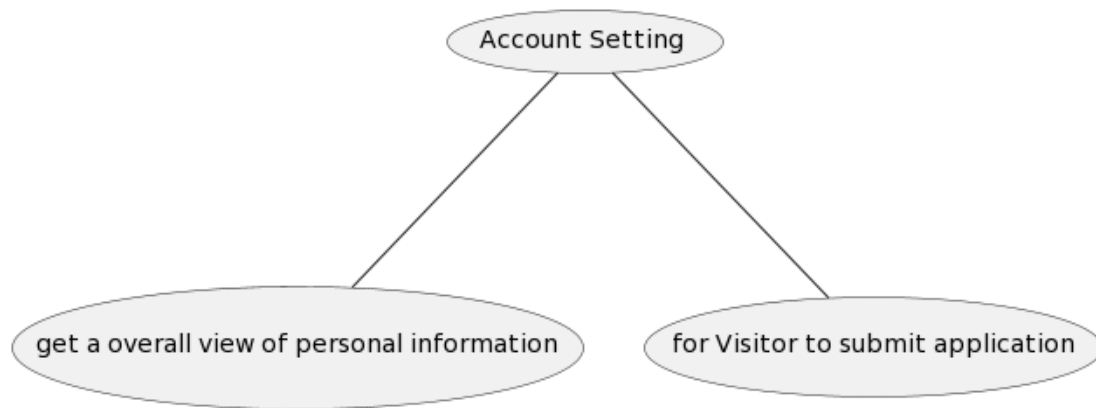


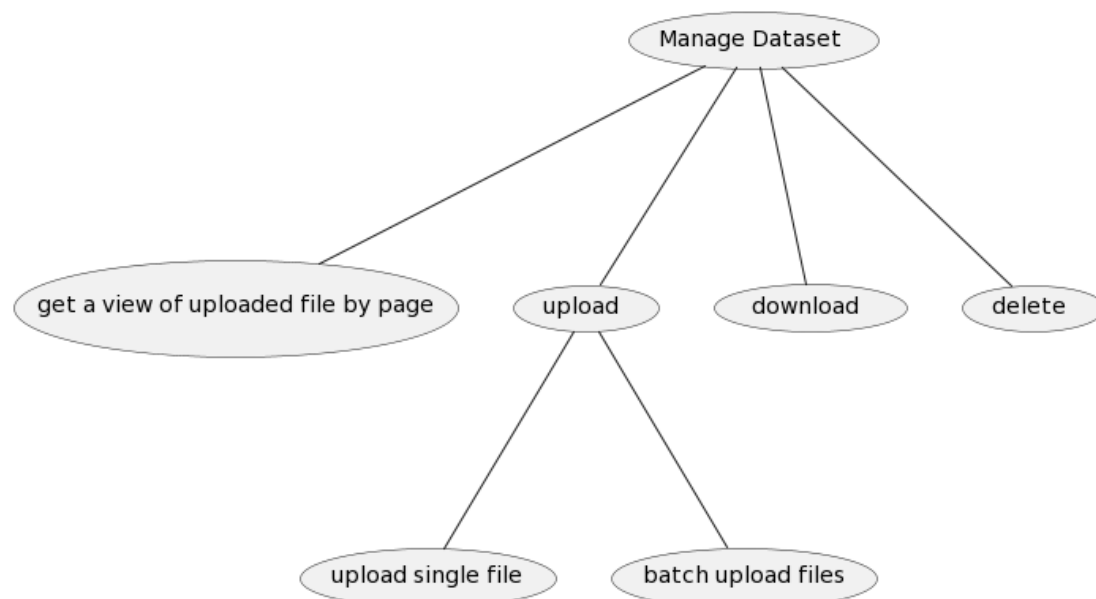
Fig 7. Function of Account Setting

## 2.2.5 Manage Dataset

In **Manage Dataset**, all users could manage their uploaded dataset files, which are used for analysis. The maximum space occupied by all dataset files is limited by the property, **max space**, in **Account Setting**.

All user except **Visitor** could choose to upload single dataset file by file URL or batch upload dataset files from a txt file. The txt file format is one file URL per line. Uploaded files are saved in a specified path in server. In order to avoid file overwriting due to duplication of file names, the names of uploaded files are renamed to random GUID.

Users also could get an overall view of all files uploaded by themselves including the original file name and file URL. They could choose to download dataset files or delete them. When they are downloaded, the file name will be restored to the original name. The function feature of Manage Dataset is shown



in Fig 8.

Fig 8. Function of Manage Dataset

## 2.2.6 Analysis Setup

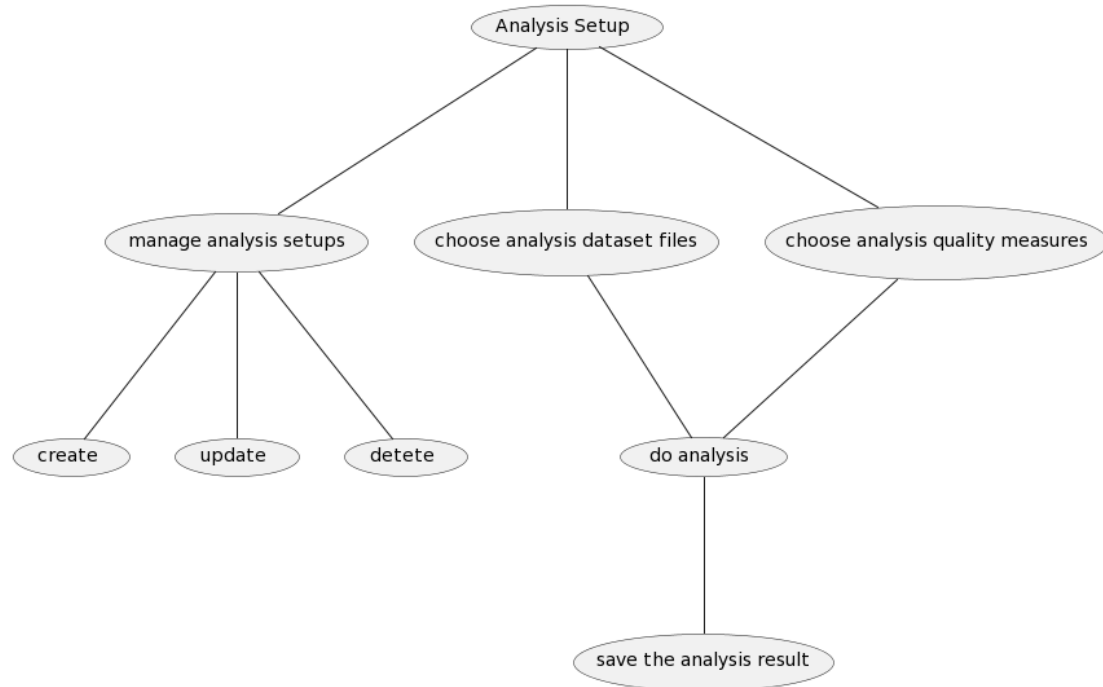


Fig 9. Function of Analysis Setup

The function feature of **Analysis Setup** is shown in Fig 9. In **Analysis Setup**, all users except **Visitor** could manage self-created analysis setup and **Administrator** could manage all analysis setup created by all users. First, **User** could get a view of self-created analysis setup by page, including the name of analysis setup, creation time and who created it.

Users could create a new analysis setup by choose the dataset files, which need to be analyzed, and quality measures from **Com-I-1 DevA**, **Com-I-5**, **Acc-I-4**, **Con-I-3**, **Con-I-2-DevB** and **Con-I-4-DevC**. The definition of thesis quality measure is shown from **Table 1.1 ~ 1.6**.

### ■ Com-I-1 DevA

<b>Id:</b> Com-I-1 DevA <b>Dimension:</b> Completeness <b>Name:</b> Data set completeness	
Description	Measurement function
Ratio of null values within a data file	Average of X where $X = A/B$ A = number of null value in the whole data set B = number of data items considered



Table 1.1. Com-I-1 DevA derivative from ISO/IEC 25024

■ **Com-I-5**

<b>Id:</b> Com-I-5 <b>Dimension:</b> Completeness <b>Name:</b> Empty records in a data file	
Description	Measurement function
False completeness of records within a data file	$X = 1 - A/B$ A = number of records where all data items are empty B = number of records in a data file

Table 1.2. Com-I-5 derivative from ISO/IEC 25024

■ **Acc-I-4**

<b>Id:</b> Acc-I-4 <b>Dimension:</b> Accuracy <b>Name:</b> Risk of data set inaccuracy	
Description	Measurement function
Ratio of null values within a data file	$X = A/B$ A = number of data values that are outliers B = number of data values to be considered in a data set

Table 1.3. Acc-I-4 DevA derivative from ISO/IEC 25024

■ **Con-I-3**

<b>Id:</b> Con-I-3 <b>Dimension:</b> Consistency <b>Name:</b> Risk of data inconsistency	
Description	Measurement function
Risk of having inconsistency due to duplication of data value	$X = A/B$ A = Number of data items where exist duplication in value B = Number of data items considered

Table 1.4. Con-I-3 derivative from ISO/IEC 25024

■ **Con-I-2-DevB**

<b>Id:</b> Con-I-2 DevB <b>Dimension:</b> Consistency <b>Name:</b> Data type consistency	
Description	Measurement function
Average consistency of data type of data item in the same attribute	Average of X where $X = A/B$ A = number of data items that have the correct type in the attribute B = number of data items considered for a single

	column
--	--------

Table 1.5. Con-I-2 DevB derivative from ISO/IEC 25024

#### ■ Con-I-4-DevC

<b>Id:</b> Con-I-4 DevC <b>Dimension:</b> Consistency <b>Name:</b> Data structure consistency	
Description	Measurement function
Degree to which the data structure remains coherent over the data file	$X = A/B$ A = Number of rows that respect the data structure B = Number of rows contained in the data file

Table 1.6. Con-I-4 DevC derivative from ISO/IEC 25024

The analysis algorithm is provided by Davida[5]. It is a practical application of the algorithm that is used to assess the data quality of the Italian open government data sets. In the thesis, the data quality application will transfer the result of this algorithm to a readable format, which could be displayed in UI.

### 2.2.7 Saved Result

After users get a output from analysis setup, they could choose to save the analysis result. Then the output will be shown in **Saved Result**. In Saved Result There is a difference like in **Analysis Setup**: **Administrator** could manage all saved analysis results created by all users and **User** only could manage self-created analysis result. First, in **Saved Result**, users could have an overall view of saved analysis results including the result name, who saved it, creation time and actions that can be performed. When users want to save their analysis result from **Analysis Setup**, the result must be named. It also could be renamed in **Saved Result**. Every analysis result should be able to be re-viewable as if they were saved from **Analysis Setup**, Users also could export the analysis result as a csv file and delete useless or unwanted results.

The function feature of **Saved Result** is shown in Fig 10.

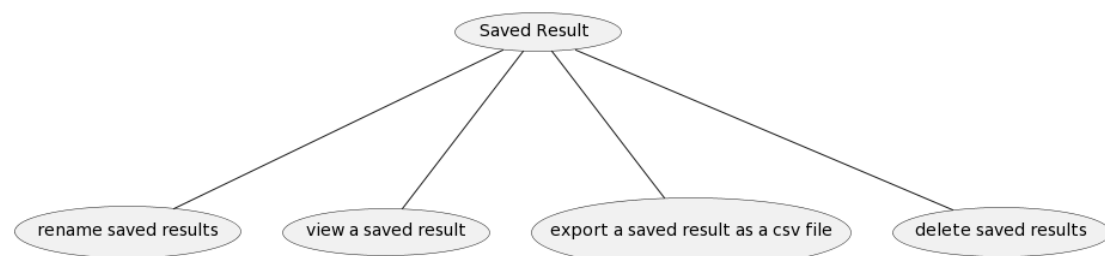


Fig 10. Function of Saved Result

### 3. Technology stack

With the emergence of new technologies, the rapid development of coding standards and improved infrastructure have led to the development of web development strategies and strategies. The medium that promotes successful web development by combining various parts is called web application architecture. Front end and back end are two main sub parts of web application architecture. A frontend is a section that a user can see while the backend is an infrastructure supporting it [6].

The front end of the website is the part that users see and interact with the help of the browser. Also known as the client, it covers all content directly from the user experience. For example, text, colors, images, navigation menus, icons, and so on. HTML, CSS and JavaScript are the basic languages for front-end development. In addition, bootstrap, angular framework and JavaScript libraries (such as react, Vue, jQuery and CSS extensions) also belong to the front end.

The back end is the part of the website that is far from the sight of users. The back end, also known as server-side code, facilitates data management and interaction in an organized manner. Communication between the back end and the front end helps to display information on Web pages. For example, when filling out a contact form, the web address is entered in the browser. The browser sends a request to the server, and the server returns the requested information as the front-end code interpreted by the browser and displayed to the user.

Here I choose the development mode of front-end and back-end separation. Because it could cause the following benefits:

- **Broad range of technical experts:** In a multi-tier development environment architecture, complex technologies are responsible for these tasks. Therefore, in order to create a complex system, dedicated technical experts are required. Dividing front-end and back-end helps to get expert programmers in their respective technical fields. Furthermore, removing constraints on technology choices, the two choices may impose on each other. This makes the development process smoother in such a development environment. This data quality application is developed by me alone. I am not good at front-end development so page design interaction is relatively simple. Those who maintain this project in the future can improve it.
- **Modularity:** Since the components or modules in this type of development model are independent, the replacement of modules or any changes are smooth. Changes in the backend module of the web application will not affect

the frontend part and vice versa. Therefore, do not overwrite or interfere with the work of others.

- **Rapid development:** As various teams work on projects in parallel and in full alignment, this facilitates rapid and synchronized development of web applications, resulting in rapid application deployment.
- **API Integration:** With the availability of a large number of devices, various versions of code (websites, iOS apps, android apps) need to be managed. Most of them require the same codebase. An API-based website simplifies everything for developers because now the API handles code management. Therefore, developers need to deal with less code.

### 3.1. Backend framework, Flask

In this thesis, the backend framework of the data quality application is Flask. Before I explain the feature of it, I will explain what is the framework. I would like to share one example from stackoverflow[7]. If I told you to cut a piece of paper with dimensions 5m by 5m, then surely you would do that. But suppose I ask you to cut 1000 pieces of paper of the same dimensions. In this case, you won't do the measuring 1000 times; obviously, you would make a frame of 5m by 5m, and then with the help of it you would be able to cut 1000 pieces of paper in less time. So, what you did was make a framework which would do a specific type of task. Instead of performing the same type of task again and again for the same type of applications, you create a framework having all those facilities together in one nice packet, hence providing the abstraction for your application and more importantly many applications.

Flask is a backend framework which is for programming in python. There is another famous framework, Django. Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. [12] Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. It is more suitable for developing complex projects.

Flask is a micro framework offering basic features of web app. This framework has no dependencies on external libraries. The framework offers extensions for form validation, object-relational mappers, open authentication systems, uploading mechanism, and several other tools. Next, the features of Flask will be explained [9]:

## ■ Routing

Flask is a back-end framework. It acts like a pure API. The back-end returns just JSON data. So it uses the **route()** decorator to bind a function to a URL. The front-end could request JSON data by the bonded URL.

```
@app.route('/')
def index():
    return 'Index Page'

@app.route('/hello')
def hello():
    return 'Hello, World'
```

## ■ Variable Rules

The variable sections could be added to a URL by making sections **<variable\_name>**. Your function then receives the **<variable\_name>** as a keyword argument. Optionally, you can use a converter to specify the type of the argument like **<converter:variable\_name>**.

```
from markupsafe import escape

@app.route('/user/<username>')
def show_user_profile(username):
    # show the user profile for that user
    return f'User {escape(username)}'

@app.route('/post/<int:post_id>')
def show_post(post_id):
    # show the post with the given id, the id is an integer
    return f'Post {post_id}'

@app.route('/path/<path:subpath>')
def show_subpath(subpath):
    # show the subpath after /path/
    return f'Subpath {escape(subpath)}'
```

Converter types:

<b>string</b>	(default) accepts any text without a slash
<b>int</b>	accepts positive integers
<b>float</b>	accepts positive floating point values
<b>path</b>	like <b>string</b> but also accepts slashes
<b>uuid</b>	accepts UUID strings

## ■ HTTP Methods

Web applications use different HTTP methods when accessing URLs. By default, a route only answers to GET requests. We can use the methods argument of the **route()** decorator to handle different HTTP methods.

```
from flask import request

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        return do_the_login()
    else:
        return show_the_login_form()
```

If **GET** is present, Flask automatically adds support for the **HEAD** method and handles **HEAD** requests according to the **HTTP RFC** [8]. Likewise, **OPTIONS** is automatically implemented for you.

## ■ The Request Object

The request object will be documented in the **API** section. Here's a board overview of some of the most common operations. First of all we have to import it from the **flask** module:

```
from flask import request
```

The current request method is available by using the method attribute. To access form data (data transmitted in a POST or PUT request), we can use the form attribute. Here is a full example of the two attributes mentioned above:

```
@app.route('/login', methods=['POST', 'GET'])
def login():
    error = None
    if request.method == 'POST':
        if valid_login(request.form['username'],
                       request.form['password']):
            return log_the_user_in(request.form['username'])
    else:
        error = 'Invalid username/password'
    # the code below is executed if the request method
    # was GET or the credentials were invalid
    return render_template('login.html', error=error)
```

What happens if the key does not exist in the form attribute? In that case a special **KeyError** is raised. We can catch it like a standard **KeyError** but if you don't do that, a HTTP 400 Bad Request error page is shown instead. So, for many situations we don't have to deal with that problem.

## ■ File Uploads

We can handle uploaded files with **Flask** easily. Just make sure set the `enctype="multipart/form-data"` attribute on the HTML form, otherwise the browser will not transmit files at all.

Uploaded files are stored in memory or at a temporary location on the filesystem. We can access those files by looking at the **files** attribute on the request object. Each uploaded file is stored in that dictionary. It behaves just like a standard Python **file** object, but it also has a **save()** method that allows you to store that file on the filesystem of the server. Here is a simple example showing how that works:

```
from flask import request

@app.route('/upload', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['the_file']
        f.save('/var/www/uploads/uploaded_file.txt')
    ...
```

To know how the file was named on the client before it was uploaded to the application, we can access the `filename` attribute. However this value can be forged so we can not trust that value. If we want to use the filename of the client to store the file on the server, pass it through the **secure\_filename()** function that Werkzeug provides for you:

```
from werkzeug.utils import secure_filename

@app.route('/upload', methods=['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        file = request.files['the_file']
        file.save(f"/var/www/uploads/{secure_filename(file.filename)}")
    ...
```

## ■ Cookie

To access cookies we can use the **cookies** attribute. To set cookies we can use the **set\_cookie** method of response objects. The `cookies` attribute of request objects is a dictionary with all the cookies the client transmits. If we want to use sessions, do not use the cookies directly but instead use the Sessions in Flask that add some security on top of cookies for you.

### Reading cookies:

```
from flask import request

@app.route('/')
def index():
    username = request.cookies.get('username')
    # use cookies.get(key) instead of cookies[key] to not get a
    # KeyError if the cookie is missing.
```

### Storing cookies:

```
from flask import make_response

@app.route('/')
def index():
    resp = make_response(render_template(...))
    resp.set_cookie('username', 'the username')
    return resp
```

Cookies are set on response objects. Since we normally just return strings from the view functions, Flask will convert them into response objects. If we explicitly want to do that, we can use the **make\_response()** function and then modify it.

## ■ About Responses

The return value from a view function is automatically converted into a response object. If the return value is a string it's converted into a response object with the string as response body, a **200 OK** status code and a *text/html* mimetype. If the return value is a dict, **jsonify()** is called to produce a response. The logic that Flask applies to converting return values into response objects is as follows:

1. If a response object of the correct type is returned it's directly returned from the view.
2. If it's a string, a response object is created with that data and the default parameters.
3. If it's a dict, a response object is created using **jsonify**.
4. If a tuple is returned the items in the tuple can provide extra information. Such tuples have to be in the form **(response, status)**, **(response, headers)**, or **(response, status, headers)**. The status value will override the status code and headers can be a list or dictionary of additional header values.
5. If none of that works, Flask will assume the return value is a valid WSGI application and convert that into a response object.

If we want to get hold of the resulting response object inside the view, we can use the **make\_response()** function.



Imagine there is a view like this:

```
from flask import render_template

@app.errorhandler(404)
def not_found(error):
    return render_template('error.html'), 404
```

We just need to wrap the return expression with **make\_response()** and get the response object to modify it, then return it:

```
from flask import make_response

@app.errorhandler(404)
def not_found(error):
    resp = make_response(render_template('error.html'), 404)
    resp.headers['X-Something'] = 'A value'
    return resp
```

## ■ APIs with JSON

A common response format when writing an API is JSON. It's easy to get started writing such an API with Flask. If you return a dict from a view, it will be converted to a JSON response.

```
@app.route("/me")
def me_api():
    user = get_current_user()
    return {
        "username": user.username,
        "theme": user.theme,
        "image": url_for("user_image", filename=user.image),
    }
```

- Depending on the API design, I may want to create JSON responses for types other than dict. In that case, we can use the **jsonify()** function, which will serialize any supported JSON data type.

```
from flask import jsonify

@app.route("/users")
def users_api():
    users = get_all_users()
    return jsonify([user.to_json() for user in users])
```

## 3.2. Frontend framework, Vue

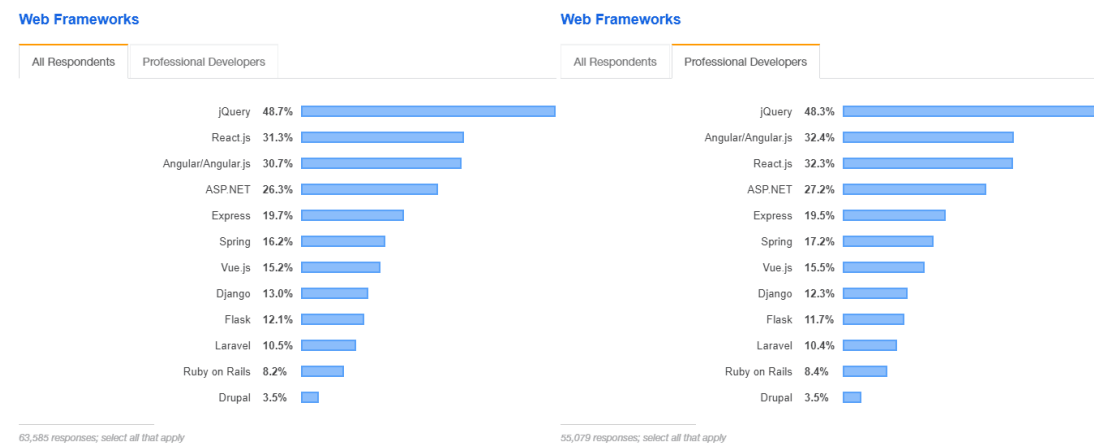


Fig 11. Popularity of Web Frameworks

A front-end framework is essentially a bundle of JavaScript code that someone else has written which you can include in your application to help you build it faster. At present, the popular front-end frameworks are React, Angular and Vue. Here, I chose Vue to develop the data quality application.

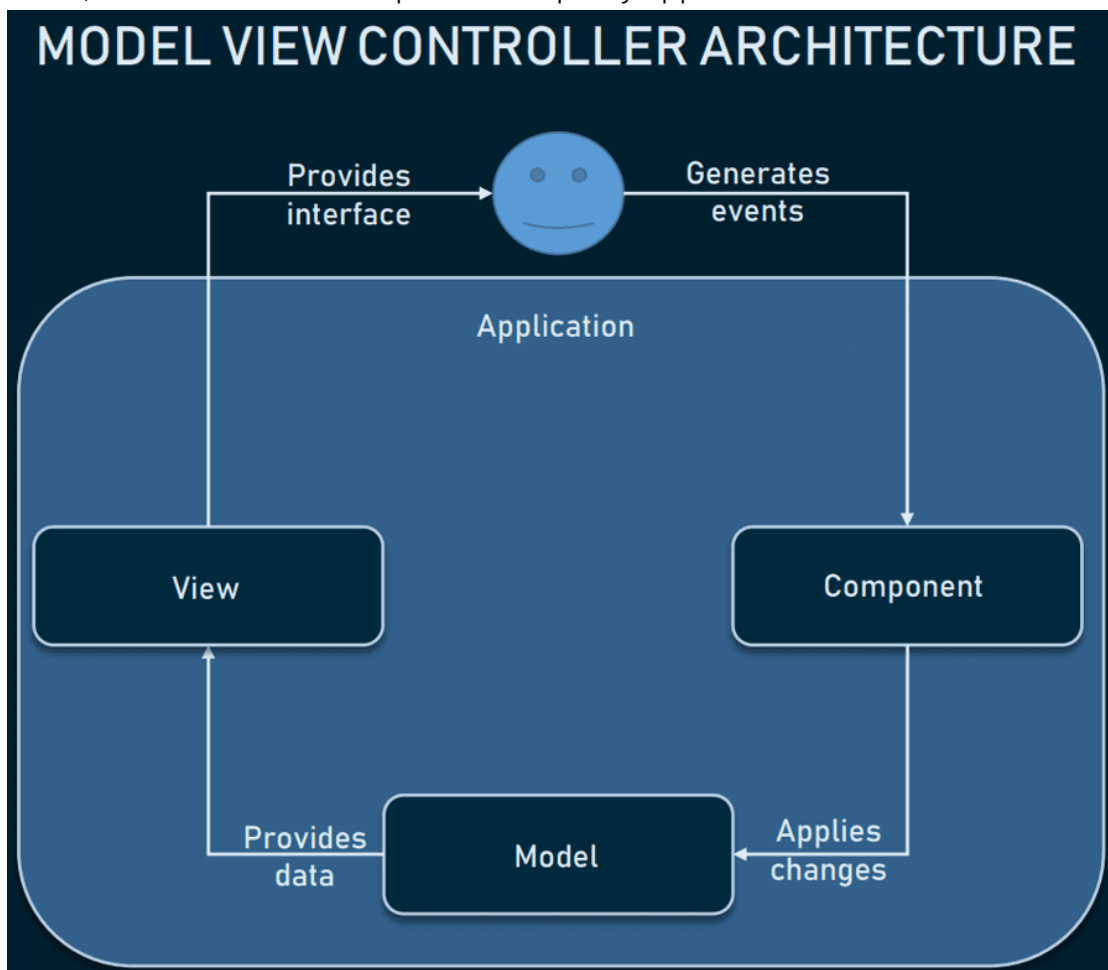


Fig 12. Model View of Controller Architecture

Vue is a progressive framework designed for the frontend development of

web applications and websites. It follows the Model–View–View-Model (MVVM) architecture and is mainly used for building user interfaces and single-page applications [10].

Created by Evan You, this lightweight, easy-to-use framework focuses on the View or Presentation layer of a web page. That is it takes care of everything that a user will see when visiting your website, be it the graphics or the clickable elements, or the login page.

The application (Data Quality Tool) is a relatively lightweight project. And Vue is friendly for beginners. Before I learned about Vue, I just known how to a backend server. Vue could help me quickly get started developing for a user interface.

The downloaded zip with the framework weighs 18 KB. As a feather-weight, the framework is not only a fast to download and install the library, it also positively impacts your SEO and UX.

A *Document Object Model* (DOM) is something we will probably encounter when rendering web pages [13]. A DOM is a representation of HTML pages with its styles, elements, and page content as objects. The objects stored as a tree structure are generated by a browser when loading a page. Performance is one of the key factors that may predetermine framework choice. Actual benchmarks are provided on the Vue comparison page. For example, when testing DOM

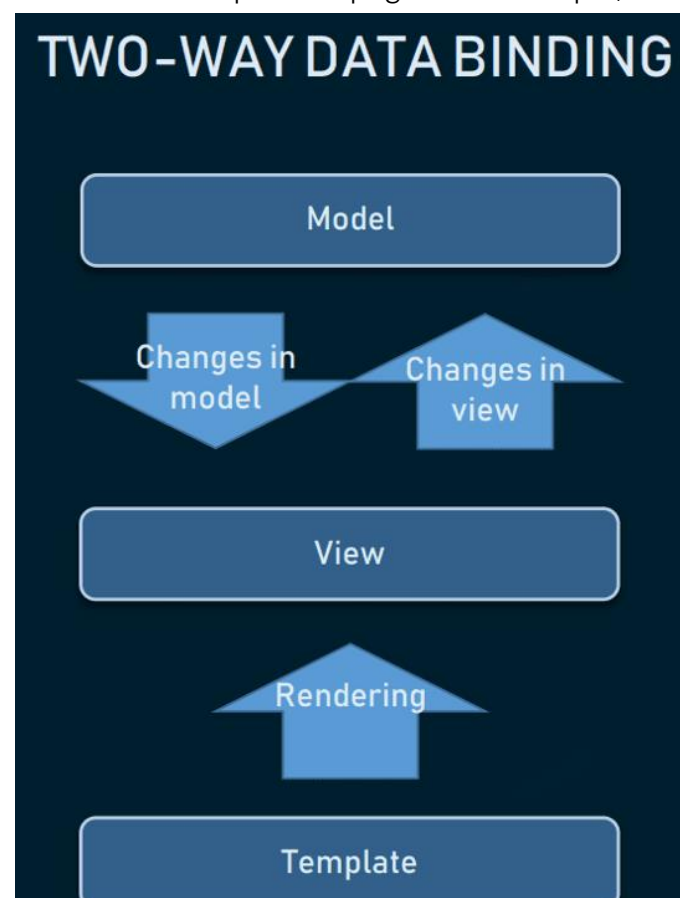


Fig 13. Reactive two-way data binding

components bound with data updated, Vue.js seems to be more performant than Angular and React.

Another benefit in DOM manipulations is two-way data binding inherited by Vue from Angular. Two-way data binding is a connection between model data updates and view (UI). Bound components contain data that can be updated from time to time. With the help of two-way data binding, it's easier to update related components and track data the updates.

### 3.3 ORM framework, SQLAlchemy

Flask and Vue work for the interaction between the front-end and back-end. We still need a solution to work for the interaction between the back-end and database. Object-Relational Mapping (ORM) is a technique that lets us query and manipulate data from a database using an object-oriented paradigm. When talking about ORM, most people are referring to a library that implements the Object-Relational Mapping technique, hence the phrase "an ORM".

An ORM library is a completely ordinary library written in the language of choice that encapsulates the code needed to manipulate the data, so we don't use SQL anymore; we interact directly with an object in the same language we are using.

SQLAlchemy is a popular SQL toolkit and Object Relational Mapper. It is written in Python and gives full power and flexibility of SQL to an application developer. It is an open source and cross-platform software released under MIT license. SQLAlchemy is famous for its object-relational mapper (ORM), using which classes can be mapped to the database, thereby allowing the object model and database schema to develop in a cleanly decoupled way from the beginning.

Major SQLAlchemy features include:

- An industrial strength ORM, built from the core on the identity map, unit of work, and data mapper patterns. These patterns allow transparent persistence of objects using a declarative configuration system. Domain models can be constructed and manipulated naturally, and changes are synchronized with the current transaction automatically.
- A relationally-oriented query system, exposing the full range of SQL's capabilities explicitly, including joins, subqueries, correlation, and most everything else, in terms of the object model. Writing queries with the ORM uses the same techniques of relational composition you use when writing SQL. While you can drop into literal SQL at any time, it's virtually never needed.

- A comprehensive and flexible system of eager loading for related collections and objects. Collections are cached within a session, and can be loaded on individual access, all at once using joins, or by query per collection across the full result set.
- A Core SQL construction system and DBAPI interaction layer. The SQLAlchemy Core is separate from the ORM and is a full database abstraction layer in its own right, and includes an extensible Python-based SQL expression language, schema metadata, connection pooling, type coercion, and custom types.
- All primary and foreign key constraints are assumed to be composite and natural. Surrogate integer primary keys are of course still the norm, but SQLAlchemy never assumes or hardcodes to this model.
- Database introspection and generation. Database schemas can be "reflected" in one step into Python structures representing database metadata; those same structures can then generate CREATE statements right back out - all within the Core, independent of the ORM.

### 3.4 Docker, docker-compose

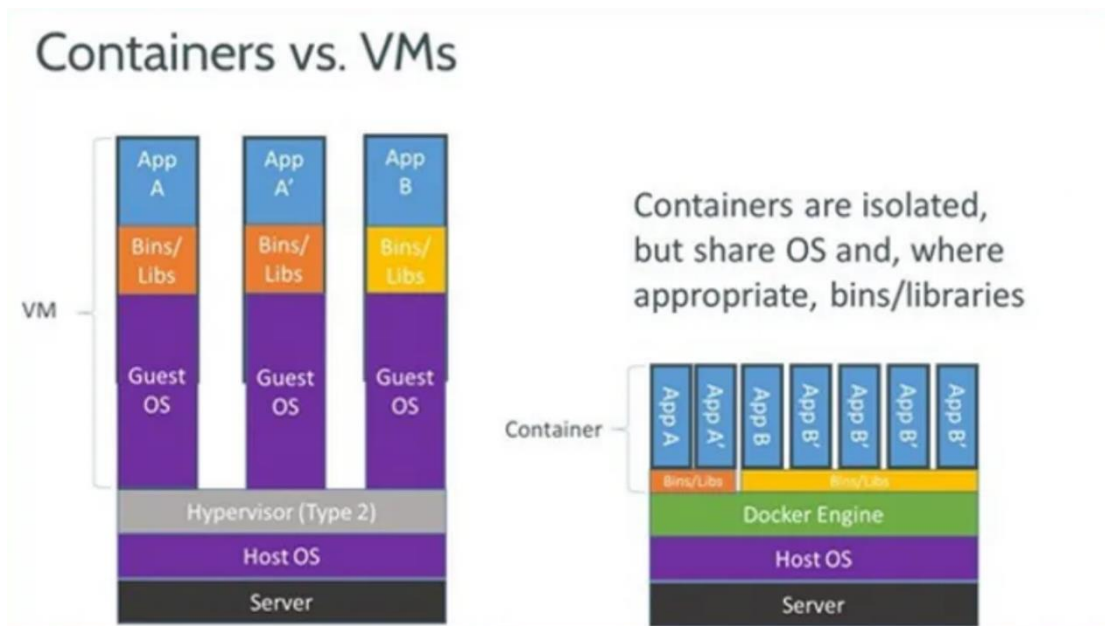


Fig 14. Differences between Containers and VMs

#### Before Docker containers

For many years now, enterprise software has typically been deployed either on “bare metal” (i.e. installed on an operating system that has complete control over the underlying hardware) or in a virtual machine (i.e. installed on an operating system that shares the underlying hardware with other “guest” operating systems). Naturally, installing on bare metal made the software painfully difficult to move

around and difficult to update—two constraints that made it hard for IT to respond nimbly to changes in business needs.

Then virtualization came along. Virtualization platforms (also known as “hypervisors”) allowed multiple virtual machines to share a single physical system, each virtual machine emulating the behavior of an entire system, complete with its own operating system, storage, and I/O, in an isolated fashion [14]. IT could now respond more effectively to changes in business requirements, because VMs could be cloned, copied, migrated, and spun up or down to meet demand or conserve resources.

Virtual machines also helped cut costs, because more VMs could be consolidated onto fewer physical machines. Legacy systems running older applications could be turned into VMs and physically decommissioned to save even more money.

But virtual machines still have their share of problems. Virtual machines are large (gigabytes), each one containing a full operating system. Only so many virtualized apps can be consolidated onto a single system. Provisioning a VM still takes a fair amount of time. Finally, the portability of VMs is limited. After a certain point, VMs are not able to deliver the kind of speed, agility, and savings that fast-moving businesses are demanding.

## **Docker container benefits**

Containers work a little like VMs, but in a far more specific and granular way. They isolate a single application and its dependencies—all of the external software libraries the app requires to run—both from the underlying operating system and from other containers. All of the containerized apps share a single, common operating system (either Linux or Windows), but they are compartmentalized from one another and from the system at large.

### **1. Docker enables more efficient use of system resources:**

Instances of containerized apps use far less memory than virtual machines, they start up and stop more quickly, and they can be packed far more densely on their host hardware. All of this amounts to less spending on IT.

The cost savings will vary depending on what apps are in play and how resource-intensive they may be, but containers invariably work out as more efficient than VMs. It’s also possible to save on costs of software licenses, because we need many fewer operating system instances to run the same workloads.

### **2. Docker enables faster software delivery cycles:**

Enterprise software must respond quickly to changing conditions. That means both easy scaling to meet demand and easy updating to add new features as the

business requires.

Docker containers make it easy to put new versions of software, with new business features, into production quickly—and to quickly roll back to a previous version if you need to. They also make it easier to implement strategies like blue/green deployments.

### **3. Docker enables application portability:**

Where you run an enterprise application matters—behind the firewall, for the sake of keeping things close by and secure; or out in a public cloud, for easy public access and high elasticity of resources. Because Docker containers encapsulate everything an application needs to run (and only those things), they allow applications to be shuttled easily between environments. Any host with the Docker runtime installed—be it a developer's laptop or a public cloud instance—can run a Docker container.

### **Docker-compose**

Docker Compose is a tool that was developed to help define and share multi-container applications. With Compose, we can create a YAML file to define the services and with a single command, can spin everything up or tear it all down.

The big advantage of using Compose is you can define your application stack in a file, keep it at the root of your project repo (it's now version controlled) [15], and easily enable someone else to contribute to your project. Someone would only need to clone your repo and start the compose app.

When development of this application (Data Quality Tool), there will be three containers, one front-end container, one back-end container and one database container.

## 4. Implementation

### 4.1 Backend API

There is a lot of APIs for front-end to request data to render page. Here I will explain the usage of each interface, including the http type and parameter format.

#### 4.4.1 Login

Interface Address: /api/account/login

Request Type: POST

Parameter:

Field	Type	Explanation
username	string	username of the account
password	string	password of the account

In the page of login, after users entered the username and password, this API will check if the username and password are correct. If wrong, it will return a response of failure. If correct, it will return a response of success. And the user information will be set in cookie. The validity period of the cookie is 24 hours, which means users will not have to log in again for 24 hours.

#### 4.4.2 Logout

Interface Address: /api/account/logout

Request Type: GET

This interface is used for logged users to log out. It will clear the cookie for logged users.

#### 4.4.3 Get current logged user information

Interface Address: /api/account/getLoginUser

Request Type: GET

After logging in, the application needs to get a information of account, including user id, username, operations and role by decrypting the cookie



#### 4.4.4 Create a new account

Interface Address: /api/account/create

Request Type: POST

Parameter:

Field	Type	Explanation
username	string	username of the account
password	string	password of the account
occupiedSpace	float	the maximum space occupied by files
otherInfo	string	other information of the account
roleId	int	id for role

**Administrator** can create new accounts through this interface.

#### 4.4.5 Update a account

Interface Address: /api/account/update

Request Type: POST

Parameter:

Field	Type	Explanation
userId	int	id for the account
username	string	username of the account
password	string	password of the account
occupiedSpace	float	the maximum space occupied by files
otherInfo	string	other information of the account
roleId	int	id for role

**Administrator** can update the information of existing accounts through this interface.

#### 4.4.6 Delete a account

Interface Address: /api/account/delete

Request Type: POST

Parameter:

Field	Type	Explanation
userId	int	id for the account

**Administrator** can delete existing accounts through this interface.

#### 4.4.7 Get a list of users in pagination

Interface Address: /api/account/getMemberPageList

Request Type: POST

Parameter:

Field	Type	Explanation
pageIndex	int	page number
pageSize	int	display quantity per page

**Administrator** can get the user paging list through this interface.

#### 4.4.8 Apply for access as User

Interface Address: /api/account/apply

Request Type: POST

Parameter:

Field	Type	Explanation
reason	string	reason of apply

**Visitor** can submit the application for access as **User** through this interface.

#### 4.4.9 Approve the application of Visitor

Interface Address: /api/account/apply

Request Type: POST

Parameter:

Field	Type	Explanation
action	string	code of approval action

The parameter, action, could be agree or reject. Agree means the application is passed. Reject the application is denied.

#### 4.4.10 Get a list of applications in pagination

Interface Address: /api/account/getApplyPageList

Request Type: POST

Parameter:

Field	Type	Explanation
-------	------	-------------

pageIndex	int	page number
pageSize	int	display quantity per page

**Administrator** can get a paging list of application through this interface.

#### 4.4.11 Get a list of notices in pagination

Interface Address: /api/notice/getNoticePageList

Request Type: POST

Parameter:

Field	Type	Explanation
pageIndex	int	page number
pageSize	int	display quantity per page

All users could get a paging list of notices through this interface.

#### 4.4.12 Upload files

Interface Address: /api/file/upload

Request Type: POST

Parameter:

Field	Type	Explanation
url	string	url of the file

Users could upload single dataset file through this interface.

#### 4.4.13 Batch upload files

Interface Address: /api/file/batchUpload

Request Type: POST

Parameter:

Field	Type	Explanation
file	form-data	the txt file

Users could batch upload dataset files through the txt file.

#### 4.4.14 Download dataset files

Interface Address: /api/file/download

Request Type: GET

Parameter:

Field	Type	Explanation
id	int	id of file

Users could download previously uploaded files.

#### 4.4.15 Get a list of files in pagination

Interface Address: /api/notice/getFilePageList

Request Type: POST

Parameter:

Field	Type	Explanation
pageIndex	int	page number
pageSize	int	display quantity per page

All users could get a paging list of files through this interface.

#### 4.4.16 Delete dataset files

Interface Address: /api/notice/getFilePageList

Request Type: GET

Parameter:

Field	Type	Explanation
id	int	id of file

All users could delete previously uploaded files through this interface.

#### 4.4.17 Save a analysis setup

Interface Address: /api/setup/create

Request Type: POST

Parameter:

Field	Type	Explanation
setupId	string	Id of the analysis setup
setupName	string	name of the analysis setup
fileIdList	array	a list of file id
selectAll	bool	a mark that whether select all files
measures	array	a list of measure codes

**User** and **Administrator** could save analysis setups for next use through this interface..

#### 4.4.18 Get the detail of a saved analysis setup

Interface Address: /api/setup/getDetail

Request Type: GET

Parameter:

Field	Type	Explanation
setupId	string	Id of the analysis setup

**User** and **Administrator** could get the detail of saved analysis setup through this interface.

#### 4.4.19 Delete a analysis setup

Interface Address: /api/setup/delete

Request Type: GET

Parameter:

Field	Type	Explanation
setupId	string	Id of the analysis setup

**User** and **Administrator** could delete previously saved analysis setups through this interface.

#### 4.4.20 Get a list of saved analysis setups in pagination

Interface Address: /api/setup/getSetupPageList

Request Type: POST

Parameter:

Field	Type	Explanation
pageIndex	int	page number
pageSize	int	display quantity per page

**User** and **Administrator** could get a paging list of previously saved analysis setups through this interface.

#### 4.4.21 Do analysis

Interface Address: /api/setup/analysis

Request Type: POST

Parameter:

Field	Type	Explanation
-------	------	-------------

selectAll	bool	a mark that whether select all files
fileIdList	array	a list of file id
measures	array	a list of selected measures

After users configured quality measures and datasets, they could get a analysis result by this interface.

#### 4.4.22 Save an analysis result

Interface Address: /api/result/save

Request Type: POST

Parameter:

Field	Type	Explanation
selectAll	bool	a mark that whether select all files
fileIdList	array	a list of file id
resultByDataset	array	the result format by dataset
fileId	int	id of the file
fileName	string	the original file name
measureList	array	a list of value for each quality measure
measure	string	key of quality measure
value	string	value of quality measure
resultByMeasure	object	the result format by measure
measure	string	key of quality measure
datasetList	array	a list of quality measure value for each file
fileId	int	id of file
fileName	string	The original file name
value	string	Value of quality measure

After users get the analysis result, they could use this interface to save the analysis result.

#### 4.4.23 Rename a saved analysis result

Interface Address: /api/result/rename

Request Type: POST

Parameter:

Field	Type	Explanation
resultId	string	the id of saved results
resultName	string	the name of saved results

Users could rename saved analysis results.

#### 4.4.24 Get the detail of a saved analysis result

Interface Address: /api/result/getDetail

Request Type: GET

Parameter:

Field	Type	Explanation
resultId	string	the id of saved results

Users could get the detail of a saved analysis result through this interface..

#### 4.4.25 Get a list of saved analysis results in pagination

Interface Address: /api/result/getAnalysisResultPageList

Request Type: POST

Parameter:

Field	Type	Explanation
pageIndex	int	page number
pageSize	int	display quantity per page

**User** and **Administrator** could get a paging list of previously saved analysis results through this interface.

#### 4.4.26 Delete a saved analysis result

Interface Address: /api/result/delete

Request Type: GET

Parameter:

Field	Type	Explanation
resultId	string	the id of saved results

**User** and **Administrator** could delete saved analysis results through this interface.

#### 4.4.27 Export a result as a csv file

Interface Address: /api/result/export

Request Type: GET

Parameter:

Field	Type	Explanation
-------	------	-------------

resultId	string	the id of saved results
----------	--------	-------------------------

**User** and **Administrator** could export analysis results as csv files through this interface.

#### 4.4.28 Export a setup as a csv file

Interface Address: /api/setup/export

Request Type: GET

Parameter:

Field	Type	Explanation
setupId	string	the id of saved analysis setups

**User** and **Administrator** could export analysis setups as csv files through this interface.

## 4.2 Frontend Router

### 4.2.1 Guide

**Path:** /Guide

This path is pointed to the page of user's guide. It is said that how to use this application from the view of three different roles, **Visitor**, **User** and **Administrator**

### 4.2.2 Account Setting

**Path:** /AccountSetting

This path is pointed to the page for users to view their account information and submit a application for access as **User**.

### 4.2.3 Analysis Setup

**Path:** /AnalysisSetup

This path is pointed to the page for users to manage their saved analysis setups.

### 4.2.4 Approve

**Path:** /Approve

This path is pointed to the page for **Administrator** to approve the application of



**Visitor.** They could choose to agree or reject the application.

#### 4.2.5 Manage Dataset

**Path:** /ManageDataset

This path is pointed to the page for users to manage their self-uploaded dataset files.

#### 4.2.6 Manage User

**Path:** /ManagerUser

This path is pointed to the page for **Administrator** to manage all accounts.

#### 4.2.7 Notice

**Path:** /Notice

#### 4.2.8 Saved Result

**Path:** /SaveResult

This path is pointed to the page for users for manage their self-saved analysis results.

#### 4.2.9 Login

**Path:** /login

This page is pointed to page for all users to login by entering username and password.

### 4.3 Database

Table name: **base\_member\_info**

Field	Type	Explanation
userId	int	user id
username	string	username
password	string	password
occupiedSpace	float	maximum space occupied by files

otherInfo	string	other relevant information
roleId	int	id for role
roleName	string	name for role
enableFlag	bool	a marker for delete

Table: **flow\_apply\_form**

Field	Type	Explanation
applyFormId	string	id for application
userId	int	id for user
username	string	account name for user
reason	string	reason for application
statusId	int	id for status of application
statusName	string	name for status of application
processUserId	int	id for administrator who approved the application
processUserName	string	name for administrator who approved the application
action	string	the action performed on the application, agreed or rejected
createDateTime	datetime	the date and time when the application was created
enableFlag	bool	a marker for delete

Table: **log\_notice**

Field	Type	Explanation
id	int	unique id, auto increment
userId	int	id for user who received this notice
username	string	name for user who received this notice
content	string	the text of notice content
readFlag	bool	a flag that whether user has read it
enableFlag	bool	a marker for delete

Table: **analysis\_setup**

Field	Type	Explanation
setupId	string	unique id for analysis setup
setupName	string	name for analysis setup
measures	string	quality selected, splited by ;

selectAll	bool	a flag for whether all dataset files are selected
filelds	string	id for dataset file selected, splited by ;
createUserId	int	id for user who created the analysis setup
createUserName	string	name for user who created the analysis setup
createDateTime	datetime	the date and time when the analysis setup was created
enableFlag	bool	a marker for delete

Table: **analysis\_file**

Field	Type	Explanation
id	int	unique id for the file
fileName	string	the original file name
filePath	string	the relative path of file saving
fileUrl	string	download URL for file
fileSize	float	the size of file
createUserId	int	id for user who uploaded the file
createUserName	string	name for user who uploaded the file
enableFlag	bool	a marker for delete

Table: **analysis\_result**

Field	Type	Explanation
resultId	string	unique id for the analysis result
resultName	string	name for the analysis result
resultContent	string	the content of the result with json format
createUserId	int	id for user who saved the analysis result
createUserName	string	name for user who saved the analysis result
createDateTime	datetime	the date and time when the analysis result was saved
enableFlag	bool	a marker for delete

## 4.4 Introduction to application

### 4.4.1 Login

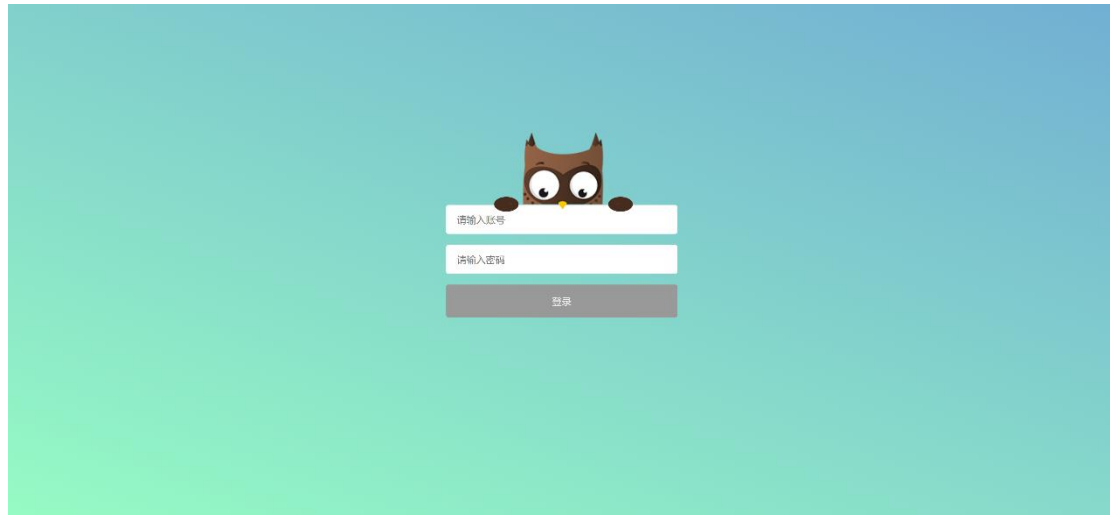


Fig 14. Page for Login

This is the login page. Users need to enter the username in the first line and password in the second line for login

### 4.4.2 Notice

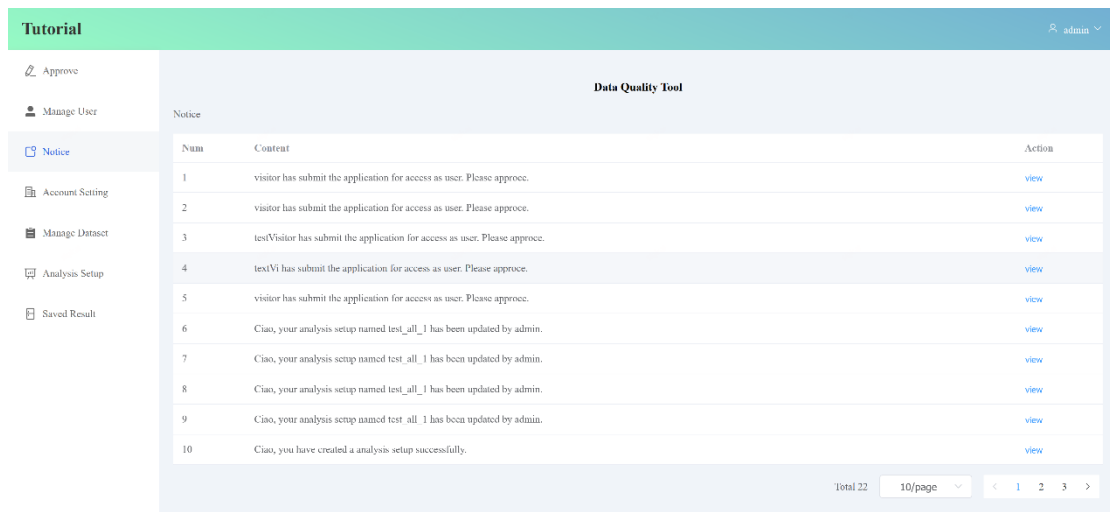


Fig 15. Page for Notice

This is the page for users to be notified when the information of their account is changed. Users will be notified in the following cases.

- The application for access as **User** is approved.
- The information in **Account Setting** is edited.
- The self-created analysis setup is edited.
- The self-saved analysis result is edited.

- The self-uploaded dataset file is edited.

### 4.4.3 Manage User

Num	User	Role	OtherInfo	Action
1	admin	Administrator	init admin account	<a href="#">view</a> <a href="#">delete</a>
2	user	User	I am a user now	<a href="#">view</a> <a href="#">delete</a>
3	testVi	Visitor	testVtestVi	<a href="#">view</a> <a href="#">delete</a>
4	visitor	Visitor	I am a visitor	<a href="#">view</a> <a href="#">delete</a>
5	visitor_siqicai	Visitor	v	<a href="#">view</a> <a href="#">delete</a>
6	test_visitor_1	Visitor	1	<a href="#">view</a> <a href="#">delete</a>

Fig 16. Page for Manager User

This module, **Manager User**, is only accessed by **Administrator**. **Administrator** could manage all account information here.

view item

\* user name

admin

\* password

admin

\* role

administrator

\* other info

init admin account

\* max space

1000

MB

Cancel

Confirm

Fig 17. Page for Account Detail

When **Administrator** want to create a new account, there are five properties, which should be set, including username, password, role, other info and max space. When **Administrator** want to remove an account, it just needs click the button “delete”, then field of mark for delete will be set false. This account data will be filtered out when getting a paging list. Actually, it is not physically from the database. It is just deleted in logical in order to restore data if wrong operation.

## 4.4.4 Approve

Tutorial						
Approve						
Data Quality Tool						
Approve						
Num	User	Reason	Status	Application Time	Action	
1	visitor	for study ...	process cad	2022-03-10 08:29:04	<a href="#">view</a>	
2	visitor	for thesis "Data analysis"	process end	2022-03-10 08:30:08	<a href="#">view</a>	
3	testVisitor	test account setting	process cad	2022-03-12 03:56:10	<a href="#">view</a>	
4	testVi	im testVi request~	process end	2022-03-12 03:59:56	<a href="#">view</a>	
5	visitor	d	process end	2022-03-14 06:44:42	<a href="#">view</a>	

Total 5 10/page < 1 >

Fig 18. Page for Approve

Like the last module, this module, **Approve**, is also only open for **Administrator**. Here they could approve the application of **Visitor** for access as **User**. When the application has been agreed, the **Visitor** applicant can get the access as **User**. It

Tutorial						
Approve						
Data Quality Tool						
Approve						
Num	User	Reason	Status	Application Time	Action	
1	visitor	for study ...	process cad	2022-03-10 08:29:04	<a href="#">view</a>	
2	visitor	for thesis "Data analysis"	process end	2022-03-10 08:30:08	<a href="#">view</a>	
3	testVisitor	test account setting	process cad	2022-03-12 03:56:10	<a href="#">view</a>	
4	testVi	im testVi request~	process end	2022-03-12 03:59:56	<a href="#">view</a>	
5	visitor	d	process end	2022-03-14 06:44:42	<a href="#">view</a>	

Total 5 10/page < 1 >

**Application Info**

Application	CreatedDateTime
visitor	2022-03-10 08:29:04
Reason	Result
for study ...	reject
ProcessUserName	ProcessDateTime
admin	2022-03-10 08:29:34

[close](#)

Fig 19. Page for Application Detail

also will receive a notice in the module **Notice**.

As shown in figure, there are four properties in the paging list, including **User**, **Reason**, **Status** and **Application Time**. There are two kinds of Status, **in process** and **process end**, which means the process status of the application.

When users click "**view**" to see the detail, it will show more properties of the application. If it is approved, users can see the processing time and who approved the application.

## 4.4.5 Account Setting

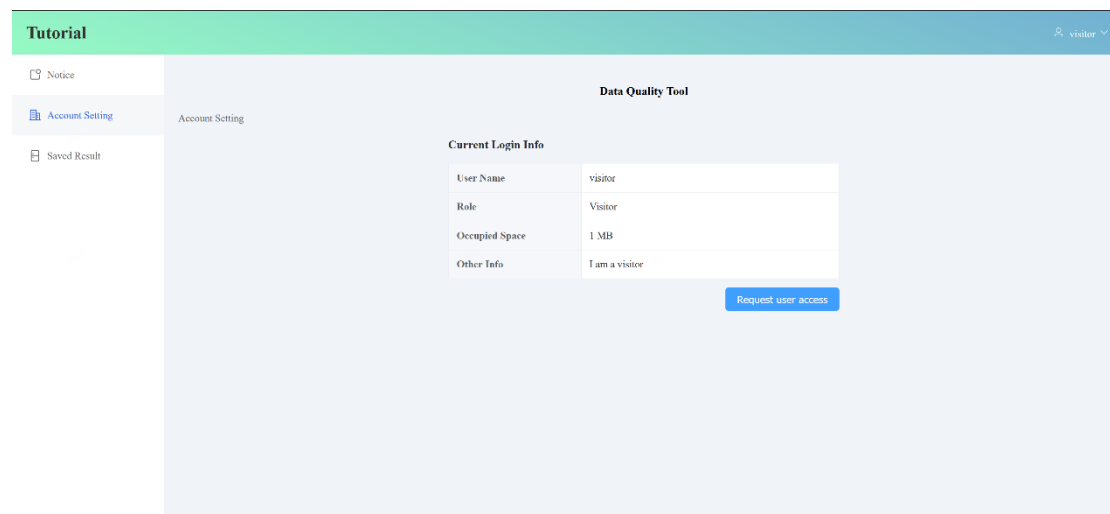


Fig 20. Page for Account Setting

This module is relatively simple. As shown in the figure, it show the account information. For **Visitor**, there is one more function, that **Visitor** could here submit the application for access as **User**.

## 4.4.6 Manage Dataset

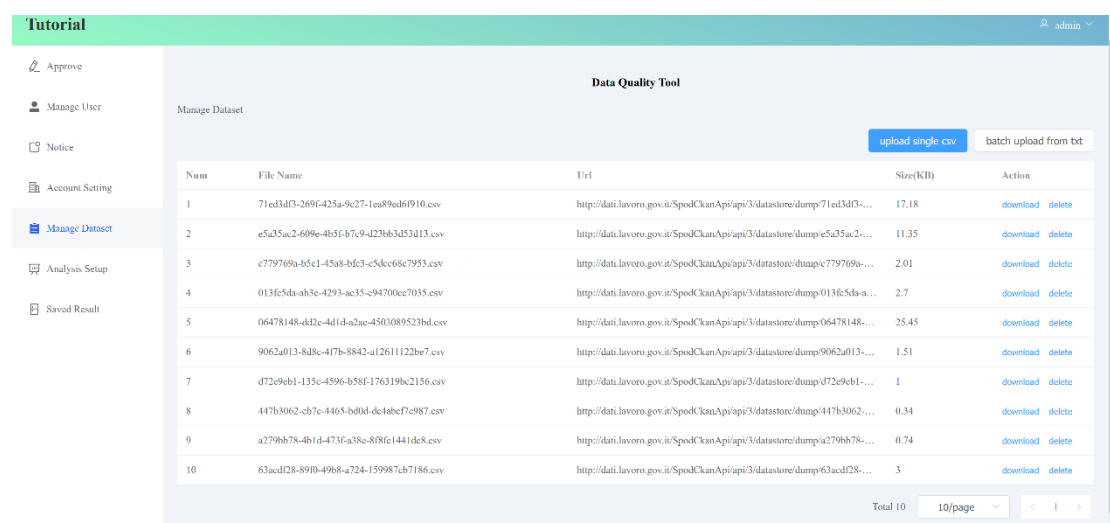


Fig 21. Page for Manager Dataset

This is an important module before data analysis. All dataset files for data analysis are managed here. Users could upload single csv file or batch upload files by upload a txt file including file URLs. In the paging list, it is shown, file name, file URL and file size. In the row **Action**, users could download files or delete it. All uploaded files here can be selected in **Analysis Setup**.

### 4.4.7 Analysis Setup

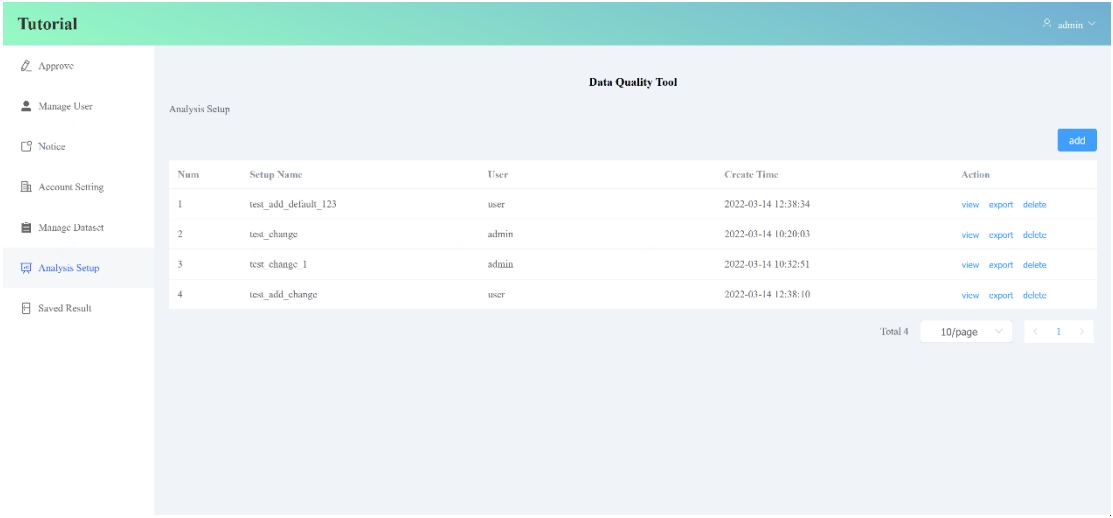


Fig 21. Page for Analysis Setup

As is shown in Fig 21, in a paging list of analysis setups, we can see the name of setup, who created it and when it is created. There are three actions, which can be performed, **view**, **export** and **delete**.

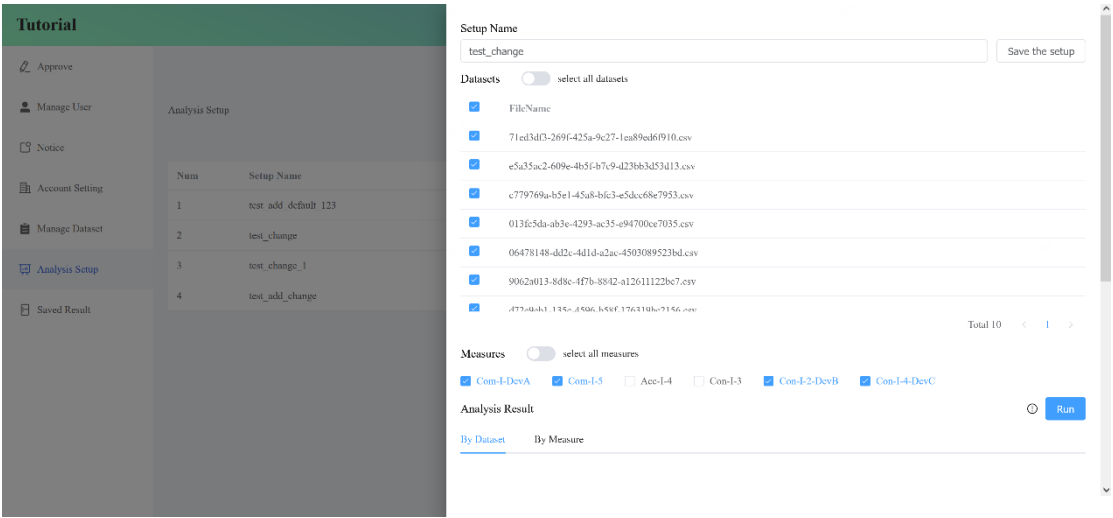


Fig 22. Page for Analysis Setup Detail

As is shown in Fig 22, we could select files that are uploaded before. And there are six quality measurement to select, including **Com-I-DevA**, **Com-I-5**, **Acc-I-4**, **Con-I-3**, **Con-I-2-DevB** and **Con-I-4-DevC**. Thesis quality measurements are explained in **Section 2.2.6**. The range of each quality measurement is from 0 to 1. There are two display modes, **By Dataset** and **By Measure**. In **By Dataset**, each table shows the value for different quality measurement on single dataset file. In **By Measure**, each table shows the value for different dataset file on single quality



measurement. They are shown in Fig 23 and Fig 24.

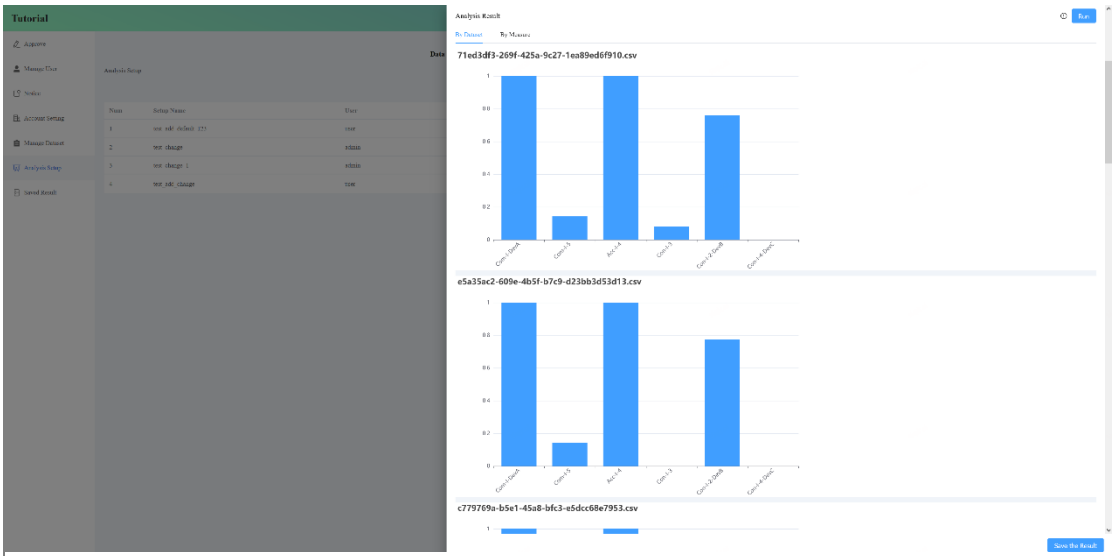


Fig 23. Display mode (By Dataset)

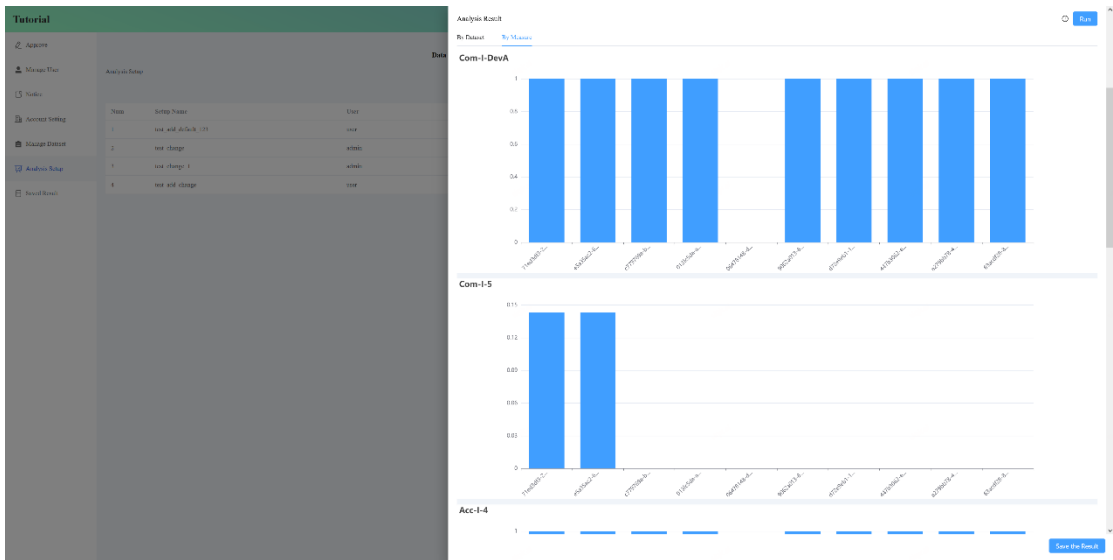


Fig 24. Display mode (By Measure)

#### 4.4.8 Analysis Result

In the last Module **Analysis Setup**, when users saved the analysis results, these saved analysis results are managed in the module, **Analysis Result**.

As is shown in Fig 25, we can see the name of result, who saved it and when it is saved. In the row of **Action**, there four buttons, rename, view, export and delete. Users could rename the saved result, view the detail of result, export the result as a csv file and delete it when it is not needed by thesis buttons.

When users see the detail of the analysis result, the display mode is the same as it in **Analysis Setup**. But we cannot edited the selected file and quality

Tutorial

Approve

Manage User

Notice

Account Setting

Manage Dataset

Analysis Setup

Saved Result

admin

Data Quality Tool

Saved Result

Num	Result Name	User	Create Time	Action
1	test_add_default	admin	2022-03-16 07:41:07	<a href="#">rename</a> <a href="#">view</a> <a href="#">export</a> <a href="#">delete</a>
2	test_all_1_123	admin	2022-03-14 06:42:35	<a href="#">rename</a> <a href="#">view</a> <a href="#">export</a> <a href="#">delete</a>
3	test add change	user	2022-03-14 12:39:08	<a href="#">rename</a> <a href="#">view</a> <a href="#">export</a> <a href="#">delete</a>
4	test_add_default_all_file	user	2022-03-14 12:41:12	<a href="#">rename</a> <a href="#">view</a> <a href="#">export</a> <a href="#">delete</a>
5	test_add_default	user	2022-03-14 12:39:40	<a href="#">rename</a> <a href="#">view</a> <a href="#">export</a> <a href="#">delete</a>
6	analysis result sample	test visitor 1	2022-03-14 07:23:35	<a href="#">rename</a> <a href="#">view</a> <a href="#">export</a> <a href="#">delete</a>

Total 6

10/page

< 1 >

Fig 25. Page for Saved Result

measurement because it is a saved analysis result. If we want to edit something, we need back to **Analysis Setup**.

## 4.5 Test case

### 4.5.1 What is open data

Open data is data that anyone can publicly access, utilize, edit and share for any purpose (even commercial purpose). Open data is licensed under an open license.

Some data should be provided to everyone free of charge to use and redistribute according to their wishes, free of copyright, patent or other control mechanisms. The goals of the open source data movement are similar to those of other open source movements, such as open source software, hardware, open content, open norms, open education, open educational resources, open government, open knowledge, open access, open science and open network. The development of open data movement is accompanied by the rise of intellectual property rights. With the rise of the Internet and the world wide web, especially with open data government programs, such as data gov、Data. Gov.uk and data gov.in.

One of the most important forms of open data is open government data (OGD), which is an open data form created by the ruling government agencies. The importance of open government data stems from its becoming a part of citizens' daily life, down to the most routine / ordinary tasks that seem to be far away from the government.

## 4.5.2 Examples of measurements on open data

### 4.5.2.1 Comune Genova

Now, let's show a case about analysis in open government data. First, we need upload datasets. I uploaded some dataset files from <http://dati.comune.genova.it>. This is government open data from Genova.

The contents are as follows:

Comune Genova

[http://dati.comune.genova.it/sites/default/files/SITGEO%20V\\_PEBA\\_AMBITI.csv](http://dati.comune.genova.it/sites/default/files/SITGEO%20V_PEBA_AMBITI.csv)

<http://dati.comune.genova.it/sites/default/files/bancadatiterra.csv>

<http://dati.comune.genova.it/sites/default/files/Consistenzadettagliosedefissa.csv>

[http://dati.comune.genova.it/sites/default/files/inquinamentob\\_2015.csv](http://dati.comune.genova.it/sites/default/files/inquinamentob_2015.csv)

[http://dati.comune.genova.it/sites/default/files/STORICO\\_CARR\\_ORTOFRUTTA\\_2016\\_0.csv](http://dati.comune.genova.it/sites/default/files/STORICO_CARR_ORTOFRUTTA_2016_0.csv)

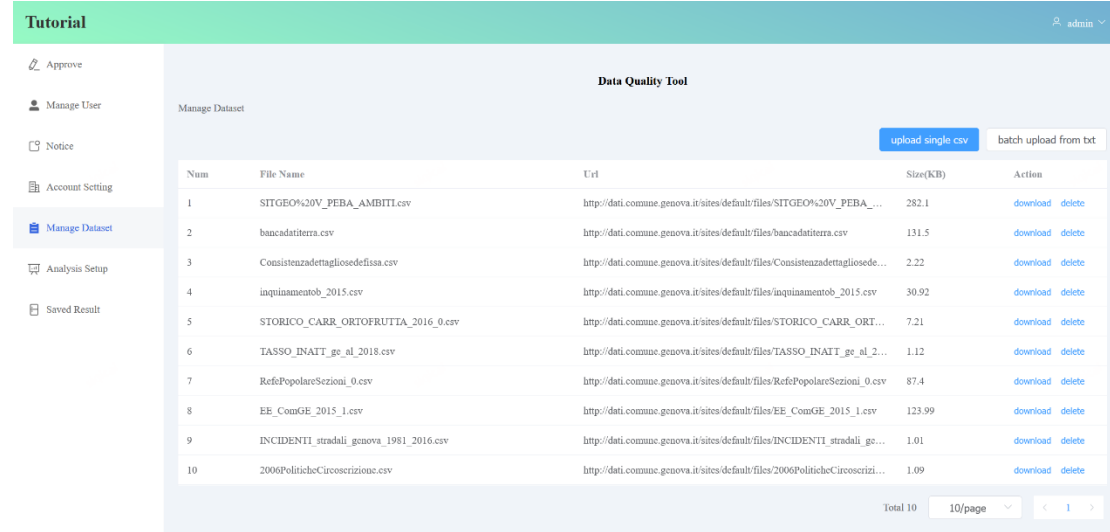
[http://dati.comune.genova.it/sites/default/files/TASSO\\_INATT\\_ge\\_al\\_2018.csv](http://dati.comune.genova.it/sites/default/files/TASSO_INATT_ge_al_2018.csv)

[http://dati.comune.genova.it/sites/default/files/RefePopolareSezioni\\_0.csv](http://dati.comune.genova.it/sites/default/files/RefePopolareSezioni_0.csv)

[http://dati.comune.genova.it/sites/default/files/EE\\_ComGE\\_2015\\_1.csv](http://dati.comune.genova.it/sites/default/files/EE_ComGE_2015_1.csv)

[http://dati.comune.genova.it/sites/default/files/INCIDENTI\\_stradali\\_genova\\_1981\\_2016.csv](http://dati.comune.genova.it/sites/default/files/INCIDENTI_stradali_genova_1981_2016.csv)

<http://dati.comune.genova.it/sites/default/files/2006PoliticheCircoscrizione.csv>



Num	File Name	Url	Size(KB)	Action
1	SITGEO%20V_PEBA_AMBITI.csv	http://dati.comune.genova.it/sites/default/files/SITGEO%20V_PEBA_...	282.1	download delete
2	bancadatiterra.csv	http://dati.comune.genova.it/sites/default/files/bancadatiterra.csv	131.5	download delete
3	Consistenzadettagliosedefissa.csv	http://dati.comune.genova.it/sites/default/files/Consistenzadettagliosed...	2.22	download delete
4	inquinamentob_2015.csv	http://dati.comune.genova.it/sites/default/files/inquinamentob_2015.csv	30.92	download delete
5	STORICO_CARR_ORTOFRUTTA_2016_0.csv	http://dati.comune.genova.it/sites/default/files/STORICO_CARR_ORI...	7.21	download delete
6	TASSO_INATT_ge_al_2018.csv	http://dati.comune.genova.it/sites/default/files/TASSO_INATT_ge_al_2...	1.12	download delete
7	RefePopolareSezioni_0.csv	http://dati.comune.genova.it/sites/default/files/RefePopolareSezioni_0.csv	87.4	download delete
8	EE_ComGE_2015_1.csv	http://dati.comune.genova.it/sites/default/files/EE_ComGE_2015_1.csv	123.99	download delete
9	INCIDENTI_stradali_genova_1981_2016.csv	http://dati.comune.genova.it/sites/default/files/INCIDENTI_stradali_ge...	1.01	download delete
10	2006PoliticheCircoscrizione.csv	http://dati.comune.genova.it/sites/default/files/2006PoliticheCircoscrizi...	1.09	download delete

Fig 26. Result of file upload

As is shown in the following figure, we can see they are uploaded successfully. Then we need to create an analysis setup. I named it **Genova** because these open data are from Genova. I selected all files that were batch uploaded and all quality measurements. The final setup is shown in Fig 27.

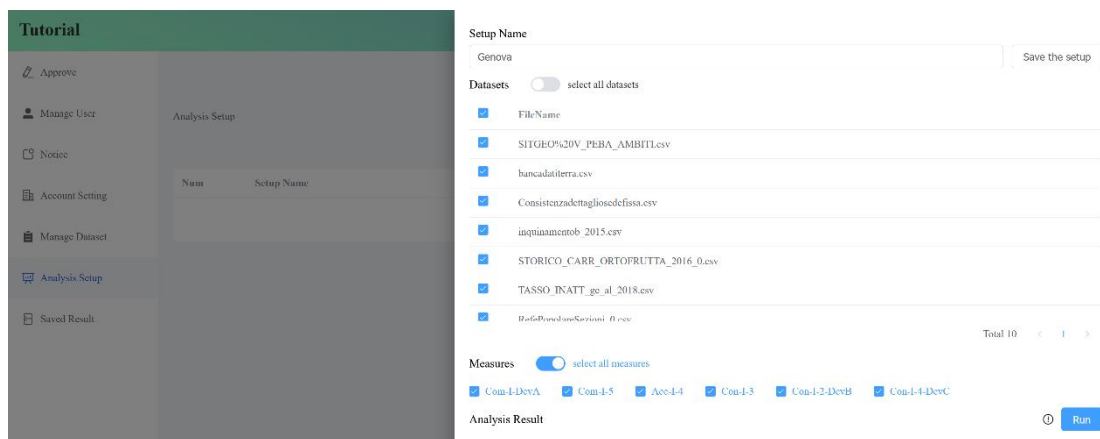


Fig 27. Setup of Genova

The analysis setup is finished. We just need to click the button, **Run**. The setup data will be sent to the backend sever to do analysis. It will take about several minutes, which is depend on the number of files.

	A	B	C	D	E	F	G
1	File Url	Com-I-DevA	Com-I-5	Acc-I-4	Con-I-3	Con-I-2-DevB	Con-I-4-DevC
2	http://dati.comune.genova.it/sites/default/files/SITGEO%20V_PEBA_AMBITL.csv	0	1	0.003	0.453	0	1
3	http://dati.comune.genova.it/sites/default/files/bancadatiterra.csv	0.06	1	0.084	0.96	0.017	0
4	http://dati.comune.genova.it/sites/default/files/Consistenzadettagliosedefissa.csv	0.125	1	0.032	0.875	0	1
5	http://dati.comune.genova.it/sites/default/files/inquinamentob_2015.csv	0	1	0.071	1.009	0	1
6	http://dati.comune.genova.it/sites/default/files/STORICO_CARR_ORTOFRUTTA_2016_0.csv	0	1	0.018	3.263	0.026	1
7	http://dati.comune.genova.it/sites/default/files/TASSO_INATT_ge_al_2018.csv	0	1	0.005	0.124	0	1
8	http://dati.comune.genova.it/sites/default/files/RefePopolareSezioni_0.csv	0	1	0.024	0.907	0	1
9	http://dati.comune.genova.it/sites/default/files/EE_CorGE_2015_1.csv	0	1	0.055	0.564	0	1
10	http://dati.comune.genova.it/sites/default/files/INCIDENTI_stradali_genova_1981_2016.csv	0	1	0.017	0.239	0	1
11	http://dati.comune.genova.it/sites/default/files/2006PoliticheCircoscrizione.csv	0	1	0.085	0.197	0	1

After I saved the result, I exported the analysis result to a csv file. The result is shown in the table. The above is a complete process, that do analysis on open data by using the application, Data Quality Tool.

From the result, we can find that most dataset files have a overall good result. First, as is shown is the table, the value of Com-I-DevA is low so we can know that almost all files have no null value, except the file in line 3 and 4. On the contrary, the value of Com-I-5 is 1, which means there is no empty records in these data files. For low value of Acc-I-4, we can find that there are few values to considered outlier in the datasets. The situation with Con-I-3 is complex, that almost values are range 0 to 1, even some value in line 6 is over 1. From the view of Con-I-3, I think there is a relatively high risk of having inconsistency due to duplication of data value. The performance of Con-I-2-DevB seems good, that there are only 2 datasets with a non-zero value, which means low ratio of the data are inconsistent with the data type of the column where they are stored. In the final, the values of Con-I-4-DevC in almost all files are 1, expect in the line 3, it is 0. Because all information about one element should be in one row but it is divided in 2 lines instead. Due to this reason, we can know that the standard architecture of CSV is not present in this dataset file.

#### 4.5.2.2 Ministero Lavoro

In this case, we will test the open government data from Lavoro. We could upload them by the batch upload feature.

The contents are as follows:

```
Ministero del Lavoro
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/615c83dc-620f-43c3-bed4-dcf63a33edc6.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/925bf200-46ad-4134-8e45-54b928709fac.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/024e1be0-75b4-431d-a403-c73882306494.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/bc702592-ed0f-4840-93ce-b604e355ffcb.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/d591fc58-7307-475b-a614-7deb987bebc4.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/198e1639-e65b-4477-8dde-1828d57f44dc.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/98082c03-0825-4d5f-b1bb-a0ea7a40c0fa.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/3bd911ca-ec88-49cc-bb2e-000e9e9d46d0.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/19a98306-534c-4324-95cb-4753cbe6a0e9.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/71ed3df3-269f-425a-9c27-1ea89ed6f910.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/e5a35ac2-609e-4b5f-b7c9-d23bb3d53d13.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/c779769a-b5e1-45a8-bfc3-e5dcc68e7953.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/013fc5da-ab3e-4293-ac35-e94700ce7035.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/06478148-dd2c-4d1d-a2ac-4503089523bd.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/9062a013-8d8c-4f7b-8842-a12611122be7.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/d72e9eb1-135c-4596-b58f-176319bc2156.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/6e510e62-f173-470b-9f55-d6f32fc85516.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/447b3062-eb7e-4465-bd0d-de4abcf7e987.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/a279bb78-4b1d-473f-a38e-8f8fe1441dc8.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/63acdf28-89f0-49b8-a724-159987cb7186.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/5dc97920-5274-4492-8d42-4c09d02f6005.csv
http://dati.lavoro.gov.it/SpodCkanApi/api/3/datastore/dump/541ec7f5-3d2f-47b5-a942-bf5efacf3505.csv
```

There is a total of 21 files above. The format of all files is CSV. After setup and clicking **Run**, we can get an analysis result of data quality measurements. In this case, we could view the result from another angle. We can see the values of single quality measurement for all files.

First, let's take a look at Com-I-DevA. Out of a total of 21 files, there null values in 11 files. In these files with null values, the maximum value is 0.333 and the minimum value is 0.111. We can find that the number of null values is high in the file of the third row. Because in this file, each line ends with “;”, where “;” represents delimiter. So, it was considered that each row has a null value and this dataset file has only two columns in total. Due to it, the value of Com-I-DevA for

the file in third row is 1/3.

**Com-I-DevA**

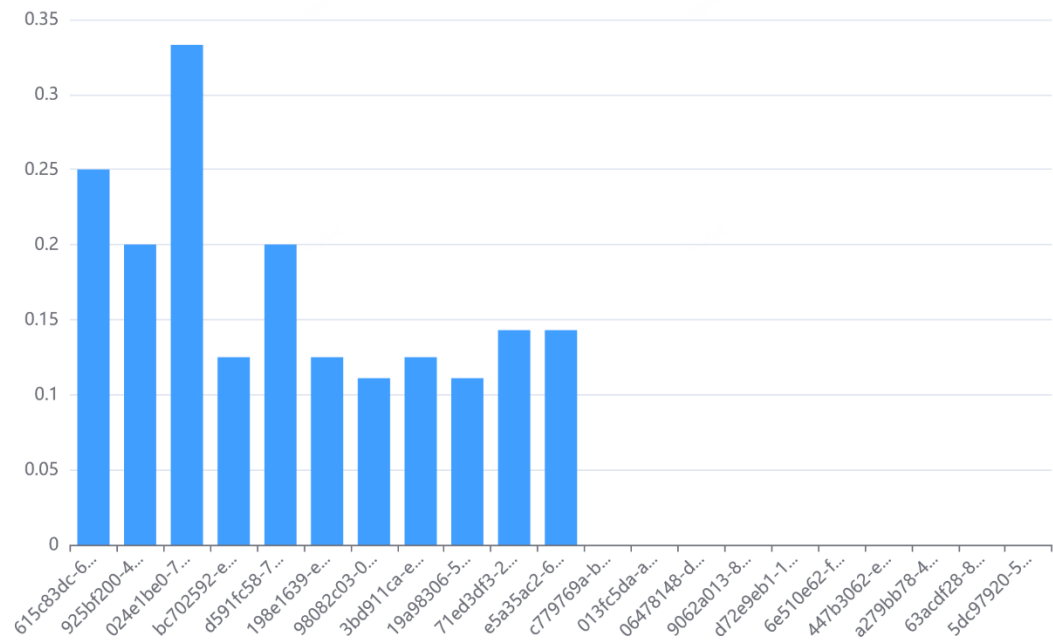


Fig 28. Value of Com-I-DevA

From the Fig 29, we have an overall view of the Com-I-5. The values of Com-I-5 for all file is 1, which means there is no row that are no completely non-null in the dataset. It is a positive result.

**Com-I-5**

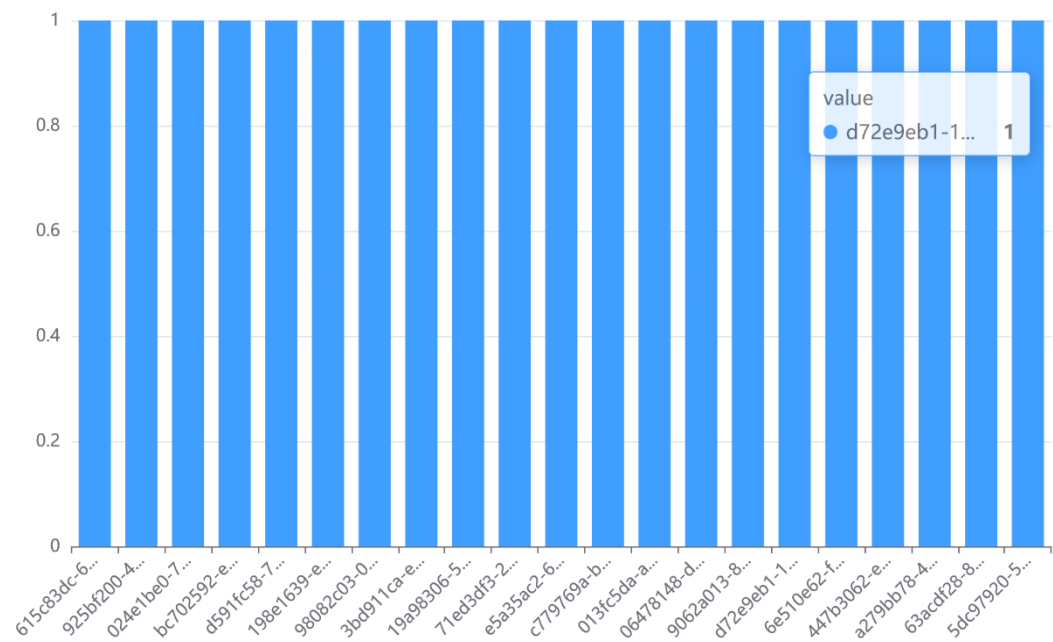


Fig 29. Value of Com-I-5

Then let's see Acc-I-4. Most of datasets have a good result in this metric. From the figure 30, we can find the file in the third row has a highest value, 0.167.

Because one of three rows of data in this file is considered null value, which also has a bad performance in Com-I-DevA due to the same reason.

**Acc-I-4**

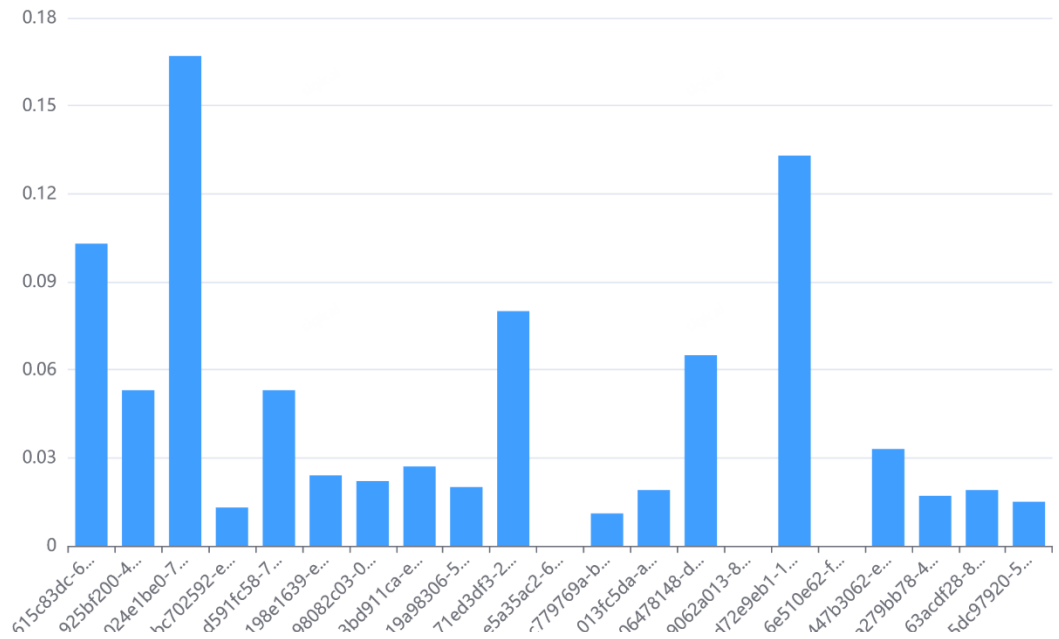


Fig 30. Value of Acc-I-4

Con-I-3 depends on the number of duplication values on the single attribute and by grouping 2 column. As is shown in figure 31, we can find there is a high percentage of duplication values, even more than 1. Because, for example, in the file in last row, there are two attributes, “Rapporte annule” and “Anno”, that have

**Con-I-3**

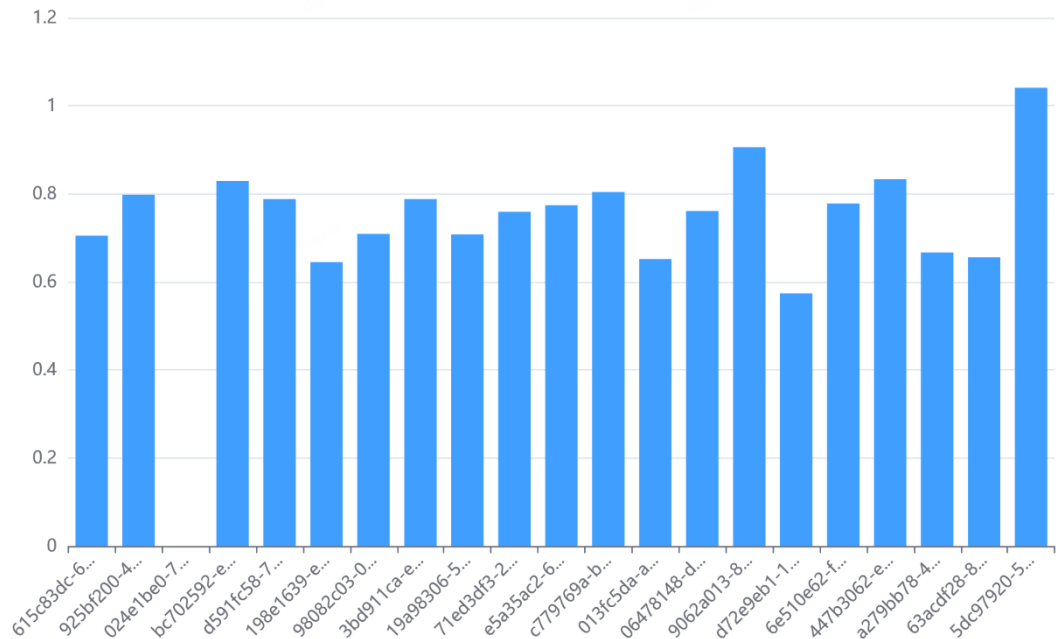


Fig 31. Value of Con-I-3

the same values, 2021 and 2020. For the attribute “Genere”, it also has a lot of repeated values, “Maschi”, “Femmine” and “Totale”. These attributes cause a larger number of values of duplication. Some boolean attributes are considered duplicated.

The performance of quality measurement, Con-I-3-DevB, are good as shown in figure 32. In average, there are only about 3% of the data, which are inconsistent with the value type of the attribute. The values of some files seem high because there are a lot of “-” as a symbol of null value, which will be considered a string. But in the same attribute, other values are numeric, which are counted as type inconsistency. We can also find it in the result of Com-I-DevA. Those files with no null value happen to have type inconsistency, which means actually there are null values in them but the symbol “-” of null value is considered string, not null. Then they seem have no null value by the result of Com-I-DevA.

#### Con-I-2-DevB

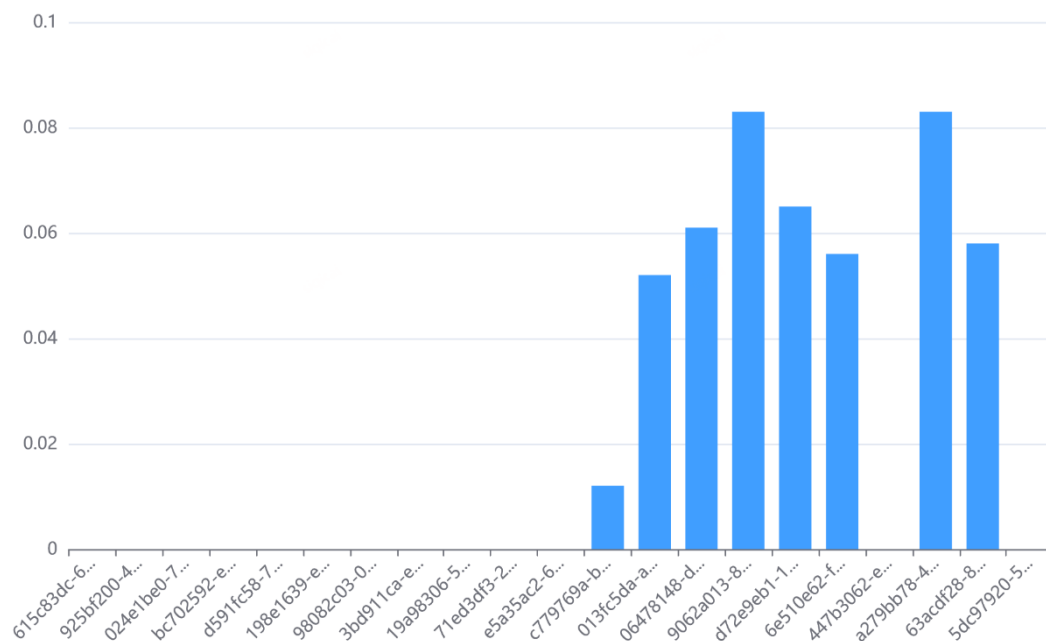


Fig 32. Value of Con-I-2-DevB

About the quality measurement, Con-I-4-DevC, as it can be seen from the figure 33, the performance is generally good. The quality measurement, Con-I-2-DevB, for all files have a value of 1, which there is no problem on the data structure. The number of rows that respect the data structure matches the number of rows contained in the data files.



### Con-I-4-DevC

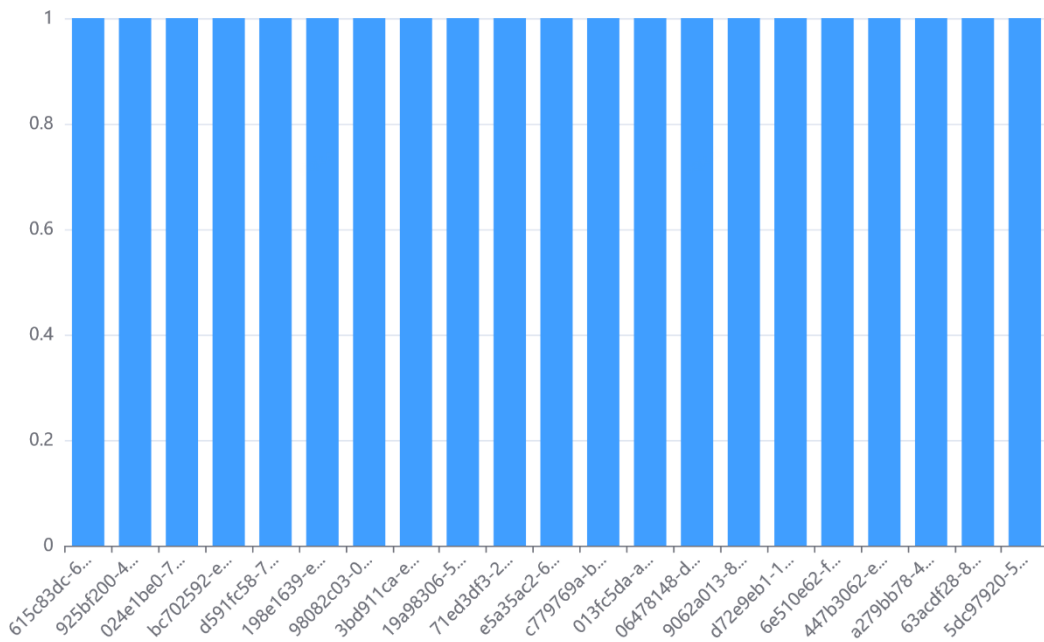


Fig 33. Value of Con-I-4-DevC

#### 4.5.2.3 Regione Sardegna

In this case, I will show the quality measurements by each dataset to make comments. The content of uploaded files is as follows:

Regione Sardegna

[http://opendata.sardegnaurismocloud.it/IT/turismo/domanda/ricettivita/movimenti-turistici/2016/movimenti\\_macrotipologia\\_2016.csv](http://opendata.sardegnaurismocloud.it/IT/turismo/domanda/ricettivita/movimenti-turistici/2016/movimenti_macrotipologia_2016.csv)

[http://opendata.sardegnaurismocloud.it/IT/turismo/domanda/telefonica/covisite/2015-2016/covisite\\_aree\\_turistiche\\_vodafone\\_2015-2016.csv](http://opendata.sardegnaurismocloud.it/IT/turismo/domanda/telefonica/covisite/2015-2016/covisite_aree_turistiche_vodafone_2015-2016.csv)

[http://www.sardegnaurismocloud.it/documenti/6\\_477\\_20180108092955.csv](http://www.sardegnaurismocloud.it/documenti/6_477_20180108092955.csv)

[https://www.sardegnaurismocloud.it/sites/default/files/opendata/anagrafica\\_enti\\_aggregati.csv](https://www.sardegnaurismocloud.it/sites/default/files/opendata/anagrafica_enti_aggregati.csv)

There are four files in total, that are from Sardegna. Then let's see the first dataset. The first file is **movimenti\_macrotipologia\_2016.csv**, which has 7 rows. The value of Com-I-DevA is 0, which means there is no null value in it. This point can also be confirmed by the value of Com-I-5. There is no rows that are not full of null. From the metric, Acc-I-4, which is 0.079, we can know there is a small ratio of the number of outliers over the dataset. The measurement, Con-I-3, is really high, because there are some columns like anno, provincia, and mese, which will cause duplication to be unavoidable. The value of Con-I-2-DevB is small, which means there is no data, that are inconsistent with the data type of the column where they are stored. For Con-I-4-DevC, it has a good result, which means this file has a

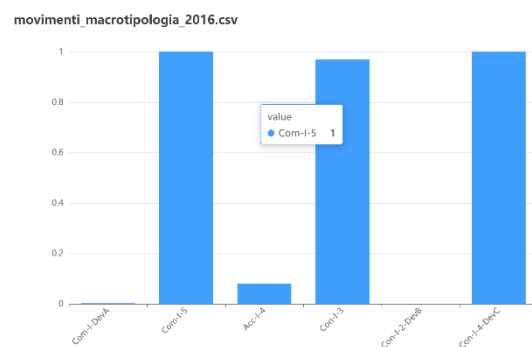


Figure 34. movimenti\_macrotipologia\_2016.csv

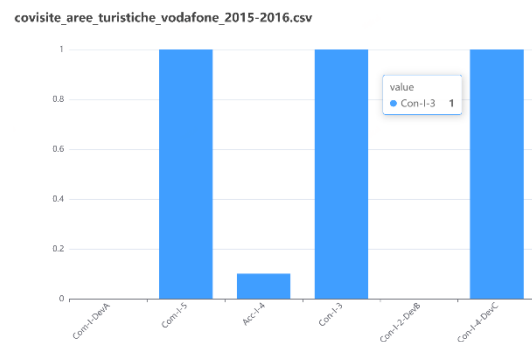


Figure 35. covisite\_aree\_turistiche\_vodafone\_2015-

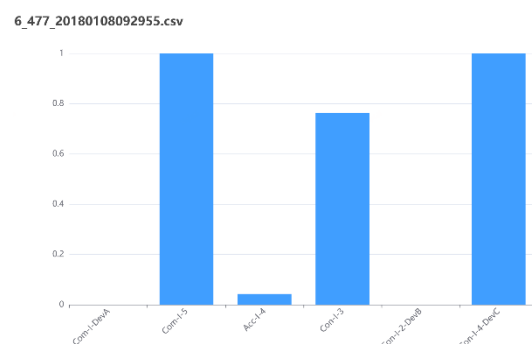


Figure 36. 6\_477\_20180108092955.csv

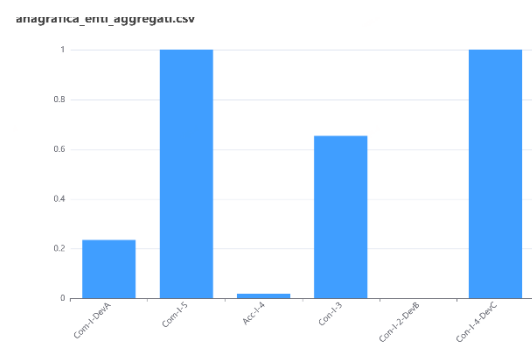


Figure 37. anagrafica\_enti\_aggregati.csv

standard csv structure. Similar situation happened in another file, **covisite\_aree\_turistiche\_vodafone\_2015-2016.csv**. Compared with the first two files, the third file, **6\_477\_20180108092955.csv**, has a lower value of Acc-I-4 and Con-I-3. For the last dataset, **anagrafica\_enti\_aggregati.csv**, it has a better result than other datasets apart from Com-I-DevA, which has a highest value, 0.235. It can be considered that there is a ratio of null values over this file. The overall analysis is as follows:

	A	B	C	D	E	F	G
1	File Url	Com-I-DevA	Com-I-5	Acc-I-4	Con-I-3	Con-I-2-DevB	Con-I-4-DevC
2	<a href="http://opendata.sardegnaturismocloud.it/IT/turismo/domanda/ricattivita/movimenti-turistici/2016/movimenti_macrotipologia_2016.csv">http://opendata.sardegnaturismocloud.it/IT/turismo/domanda/ricattivita/movimenti-turistici/2016/movimenti_macrotipologia_2016.csv</a>	0.002	1	0.079	0.969	0	1
3	<a href="http://opendata.sardegnaturismocloud.it/IT/turismo/domanda/telefonata/covisite/2015-2016/covisite_aree_turistiche_vodafone_2015-2016.csv">http://opendata.sardegnaturismocloud.it/IT/turismo/domanda/telefonata/covisite/2015-2016/covisite_aree_turistiche_vodafone_2015-2016.csv</a>	0	1	0.101	1	0	1
4	<a href="http://www.sardegnaturismo.it/documenti/6_477_20180108092955.csv">http://www.sardegnaturismo.it/documenti/6_477_20180108092955.csv</a>	0	1	0.042	0.762	0	1
5	<a href="https://www.sardegnaturismo.it/sites/default/files/opendata/anagrafica_enti_aggregati.csv">https://www.sardegnaturismo.it/sites/default/files/opendata/anagrafica_enti_aggregati.csv</a>	0.235	1	0.018	0.654	0	1

# 5. Conclusions

## 5.1 Conclusion

In my point of view, data quality is a broad concept. But its importance is known by government. In beginning of thesis, we have known what is data quality tool and those professional and commercial data quality tool on the market, which has complex functions. So, the aim of thesis is to design and develop an easy-to-use application to analysis on Italian Open Government Data.

In the chapter 2, we have an understanding for the overall needs and modules of application design, which gave us a general view of the application. The application is divided into 7 main modules, **Approve**, **Manage User**, **Notice**, **Account Setting**, **Manage Dataset**, **Analysis Setup** and **Saved Result**.

In the chapter 3, I introduced the technology stacks, that are applied in development. I explained the solutions for back-end, front-end, database and deployment. **Flask** is a micro web framework, which provides a full ability for API while **Vue** is a beginner friendly framework. Especially using docker-compose to deploy, it makes deployment much easier without worrying about the problems of environment. We just need a file **docker-compose.yml** to manage docker containers.

In the chapter 4, I explained the implementation of the application in detail, including back-end APIs, front-end router paths and database structure in practice. We were also shown the graphical user interface diagram of the application and how to use this data quality by offering test cases. It is important that we can define the analysis setup and view the analysis results in two display modes, which means we can save and see data quality measurements online and don't need to save the file of data quality measurements. Of course, we can also export data quality measurements to a csv files if necessary. From test cases, some comments were made for some strange values in order to find the causes, like different symbols for expressing null values, which should be considered, in the future, to avoid affecting the evaluation of data quality.

## 5.2 Future Implementation

As is said in the section 4.5. When there are a large number of files to analysis, it will take a long time. I think the future improve could make the algorithm more efficient. For now, there are two display modes, **by dataset** and **by measure**, that

both display quality measurements for single files. But we usually analysis datasets from the same region one time. It might be considered that average values for all datasets could be displayed and be grouped by quality measurement. This allows users to intuitively feel the differences between different regions to make the overall evaluation on quality measurements for regions.

From [dati.gov.it](https://dati.gov.it), we can find that csv is the most common data format in government data. There are also other big data themes, like environment , population and society, which needs to be paid attention on data quality. In environment data, there is not only csv but also other data formats, SHP and TIFF. It is a challenge to analysis this type of datasets. But I think it is available to make the algorithm applicable to population and society data. In population and society data, the most common data format is xlsx, which is similar with csv. By some technologies, the xlsx files could be converted to csv, which is not difficult to be realized.

## 6. List of references

- [1] Jack Vaughan, Data quality,  
<https://www.techtarget.com/searchdatamanagement/definition/data-quality>
- [2] F Gualo, M Rodríguez, Verdugo J, et al. Data Quality Certification using ISO/IEC 25012: Industrial Experiences[J]. 2021.
- [3] ISO/IEC, ISO/IEC 25012. Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model, ISO/IEC, International Standard, 2008.
- [4] ISO/IEC, ISO/IEC 25024. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of data quality, ISO/IEC, International Standard, 2015.
- [5] Davide Vitaletti, Quality analysis of the Italian open government data through a generalized algorithm
- [6] <https://opensenselabs.com/blog/articles/frontend-backend/frontend-backend>
- [7] <https://stackoverflow.com/questions/2964140/what-is-a-software-framework>
- [8] <https://www.ietf.org/rfc/rfc2068.txt>
- [9] <https://flask.palletsprojects.com/en/2.1.x/quickstart/#apis-with-json>
- [10] <https://vuejs.org/guide/introduction.html>
- [11] Fan W, Geerts F. Foundations of data quality management[J]. Synthesis Lectures on Data Management, 2012, 4(5): 1-217.
- [12] Grinberg M. Flask web development: developing web applications with python[M]. " O'Reilly Media, Inc.", 2018.
- [13] Song J, Zhang M. Design and Implementation of a Vue. js-Based College Teaching System[J]. International Journal of Emerging Technologies in Learning, 2019, 14(13).
- [14] Rad B B, Bhatti H J, Ahmadi M. An introduction to docker and analysis of its performance[J]. International Journal of Computer Science and Network Security (IJCSNS), 2017, 17(3): 228.
- [15] List M. Using docker compose for the simple deployment of an integrated drug target screening platform[J]. Journal of Integrative Bioinformatics, 2017, 14(2).