

## Technical Assignment: Software Engineer

**Problem:** Folder structure application.

Capabilities:

- User can add a folder to the application and the application reads the content and convert them into an internal python data structure. The folder could contain: a sub-folder, text files, images, and videos.
- User can view the history of previously added folders.
- User can view the properties of the folders and their contents:
  - o If a file, the user can view the name, size, type, and upload date of the file.
  - o If a folder, the user can view the name, the upload date and the content of the folder. If the folder contains another folder, it should be shown in a tree format.
- User can filter the content of the folder in the application based on certain filter. The filtering results should hold the folder structure. We leave you to decide what are the properties that can be used as filter parameter.
- User can delete an added folder or a part of the folder.

Additional info:

- The application should be capable of handling big folders with thousands of files/folders.
- The application can be implemented as a server or as a desktop application.
- The application can be either GUI-based or CLI-based.
- The application must be implemented on Python.
- Do not use `pathlib` (or other directory library) to represent and maintain your internal data structure. Such directory utility library can only be used to read the folder content for `add` command.

**Deliverables:**

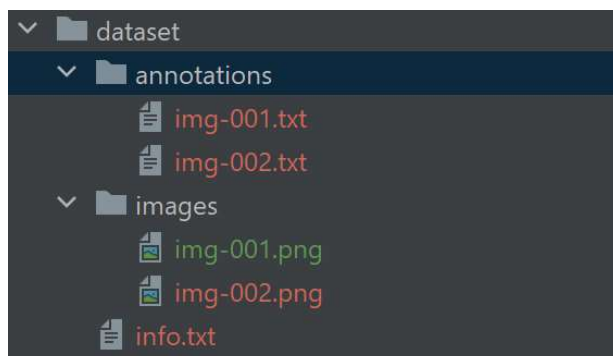
- Code base. You can either send a zip file or put the code in a github repo.
- ReadMe explaining how to run the code (including motivation of the choices in code design)

**Evaluation Points:**

- Feature completeness
- Design of data structure: robustness and flexibility.
- Code quality and readability

**Testing Scenario:**

Assume the user has a folder with the following structure on their machine:



User runs the application, for example: `python main.py`

In this example the application is CLI-based, so the application waits for user command. The command types are as follows:

- **Add**

The user can add a folder to the application through the `add` command by giving the folder path, for example:

```
>> add /path/to/dataset
```

- **View**

The view command lets the user view the list of all the added folders (and additional info, such as added time). For example, if two folders exist in the application:

```
>> view
```

Folder		upload time
-----		
Dataset		11:11:11 10/10/2021
Folder 2		11:11:12 10/10/2021

The view command can also specify a particular folder in the application. When specified, the application should show the content and the structure of the folder, along with their additional info (details up to you). For example:

```
>> view dataset
```

```
dataset
|- annotations
  |- img-001.txt (1kb, text)
  |- img-002.txt (0.5kb, text)
|- images
  |- img-001.png (1mb, image)
  |- img-002.png (2mb, image)
|- info.txt (2kb, text)
```

- **Filter**

The filter command is used to filter the content and show the content in the same view as the view command. The filtering parameters supported by the application are up to you. Here, we show an example of filtering parameters type and name.

For example:

```
>> filter dataset -type text
```

```
dataset
|- annotations
  |- img-001.txt (1kb, text)
  |- img-002.txt (0.5kb, text)
|- info.txt (2kb, text)
```

```
=====
```

```
>> filter dataset -name img*
```

```
dataset
|- annotations
|  |- img-001.txt (1kb, text)
|  |- img-002.txt (0.5kb, text)
|- images
|  |- img-001.png (1mb, image)
|  |- img-002.png (2mb, image)
```

#### - Delete

The delete command can delete the content of a folder. For example:

```
>> delete dataset/annotations/img-001.txt
```

```
>> view dataset
```

```
dataset
|- annotations
|  |- img-002.txt (0.5kb, text)
|- images
|  |- img-001.png (1mb, image)
|  |- img-002.png (2mb, image)
|- info.txt (2kb, text)
```

The delete command can also be used to delete an added folder from the application. For example:

```
>> delete dataset
```

```
>> view
```

Folder		upload time
Folder 2		11:11:12 10/10/2021

**Note that all visualizations and command formats shown in the testing scenario are for example purposes only, they are not a part of the requirement.** You are free to decide on how the data visualizations and the commands are. The capabilities mentioned above are the minimum capabilities; you are free to add any capability which you might think interesting for such an application.