August 2018

USER DASHBOARD FOR REUSABLE ANALYSIS PLATFORM

AUTHOR:

Sinclert Pérez

Digital Repositories (IT-CDA-DR)

SUPERVISOR:

Tibor Simko

Digital Repositories (IT-CDA-DR)

ABSTRACT

REANA is a reusable analysis platform which offers physicists the ability to structure their research data analysis and run their computational workflows in a containerized computing cloud.

The goal of this project was to develop a web interface dashboard in order to improve user interaction with the platform. The dashboard will include an overview of the previously submitted workflows, inspection of their running status and outputs, and job monitoring capabilities.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Over the last few years there has been a reproducibility crisis in most of the sciences. According to a 2016 Nature's paper [1]: *"More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half have failed to reproduce their own experiments".* This problem is far from being isolated due to its presence in a variety of fields:

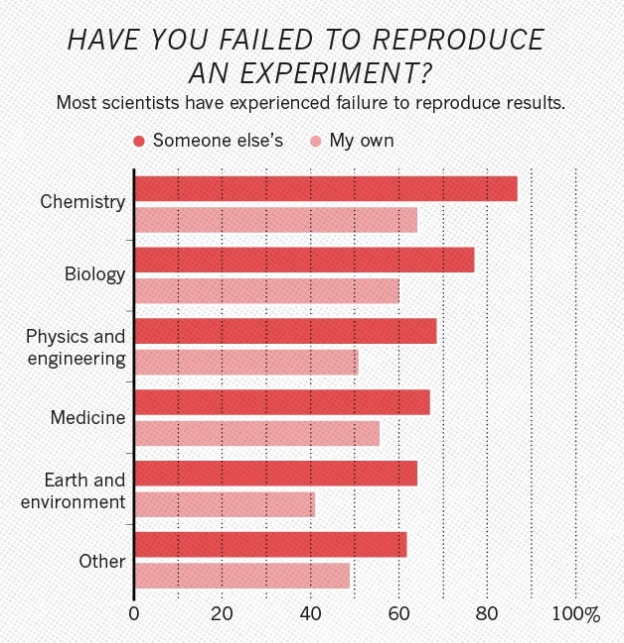

Figure 1: Reproducibility failures in sciences **[1]**

In order to solve this problem in the area of High Energy Physics (HEP), the Digital Repository section at CERN has developed REANA: a reusable and reproducible research data analysis platform that offers tools to particle physicists to structure their data analysis, so they can be instantiated and run on remote containerized compute clouds.

REANA is composed of a set of components, from which *REANA-Client* [2] is the one providing a Command Line Interface (CLI) in order to create and run data analysis workflows. In addition to the CLI, there was a clear need for workflow status visualization over the web.

This necessity set up the reason for the creation of a new module, *REANA-UI* [3], which will communicate through REST API with the already existing *REANA-Server* [4] module. My responsibility concerned the design and development of this new component.

2. REQUIREMENTS

The first step of the project was to define the basic user requirements that the User Interface (UI) needed to have. In order to do so, several wireframes were created and discussed allowing the team to define the interface views and structure.

From the discussion, we concluded with an Information Architecture including the following pages:

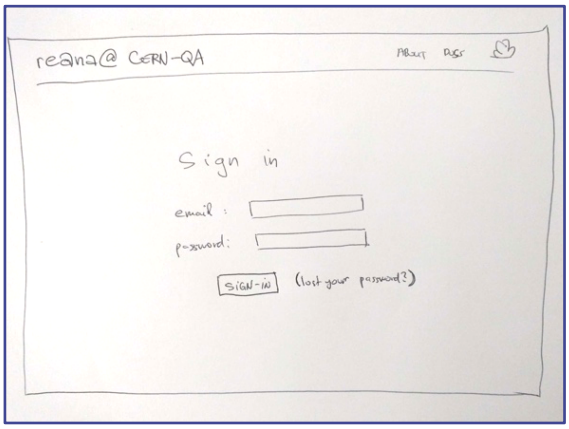

Figure 2: Login page wireframe

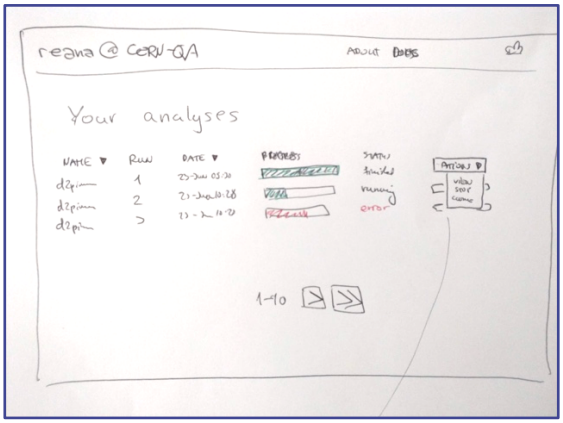

Figure 4: Workflow-list page wireframe

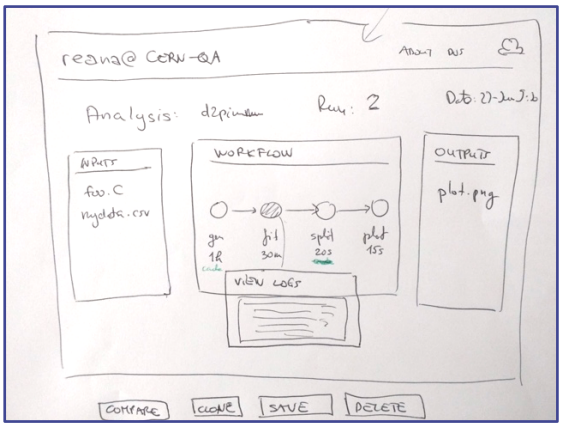

Figure 3: Workflow-details page wireframe

From those wireframes, a list of core UI elements was extracted:

- Progress bar.
- Sorting tables.
- Pagination.
- Graph visualization.
- Buttons with disability option.

3. TECHNOLOGY SELECTION

3.1. INTERFACE BUILDING

Regarding the interface building framework, we selected ***React*** over its competitors (*Angular* and *Vue*) due to the synergies and similarities with other CERN projects using this framework.

The preference for *React* is shared among several projects within the Digital Repositories section, including the CERN Analysis Preservation project with whom REANA shares the data analysis preservation and reusability goals.

3.2. INTERFACE STYLE

The UI style framework decision was important due to the long perspective life time and complexity of this project. The chosen framework needed to be mature enough, so it could provide most of the core requirements defined in section 2.

The frameworks to consider in this section were: *Grommet* [6], *Material-UI* [7] and *Semantic-UI* [8]. In order to select the most appropriate one, different comparisons were made.

The first comparison consisted on creating a UI elements table to discover the frameworks advantages and disadvantages:

Required element	Grommet	Material-UI	Semantic-UI
Disabled buttons	No	Yes	Yes
Progress bar labels	Yes	No	Yes
Progress bar colors	Yes	No	Yes
Nodes graphs	Yes	No	No
Table pagination	No	Yes	Yes
Table sorting	Yes	Yes	Yes
Documentation quality	Medium	Medium	Good
Examples quality	Medium	Good	Good

Additionally, several technical mock-ups were implemented using a platform called *StoryBooks* [5] for fast and easy prototyping. The next three figures are examples of how the *workflow-list* page technical mock-ups looked:

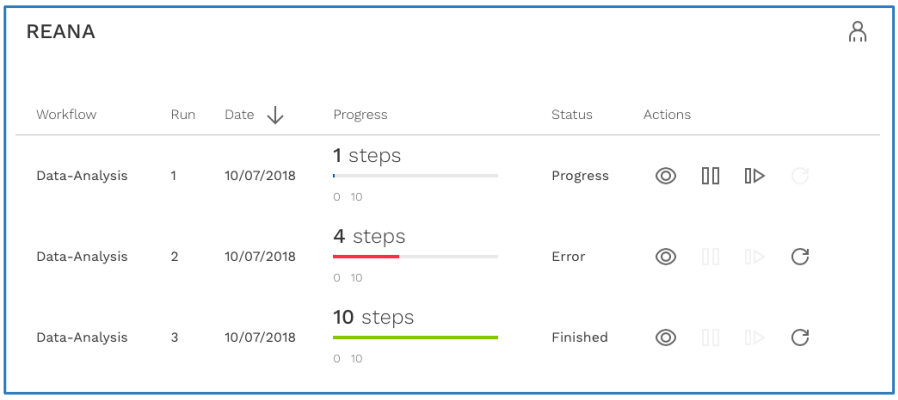

Figure 7: Grommet workflow-list page technical mock-up

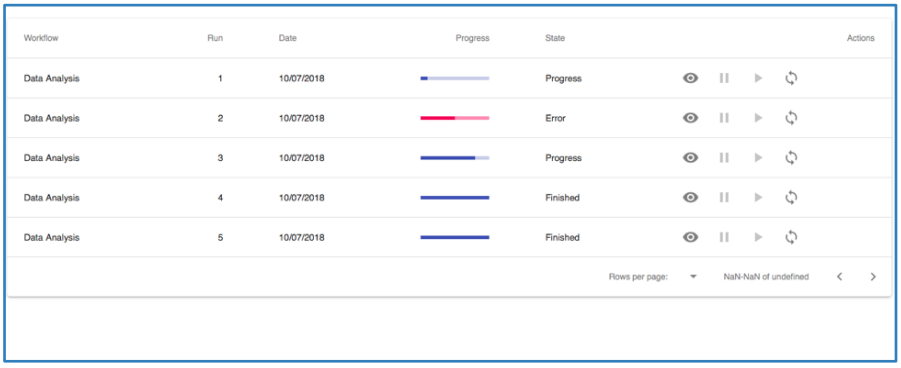

Figure 5: Material-UI workflow-list page technical mock-up

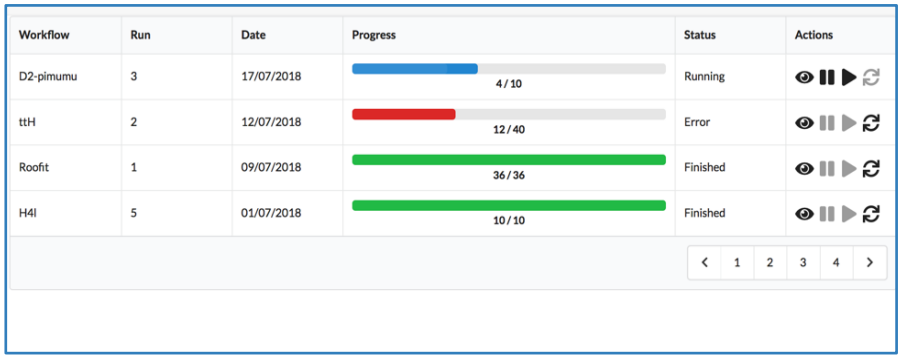

Figure 6: Semantic-UI workflow-list page technical mock-up

From the different frameworks, **Semantic-UI** was chosen due to the maturity of its development and the good quality of its documentation and examples.

3.3. WORKFLOW VISUALIZATION

In the case of the data analysis workflows, they are defined as directed acyclic graphs (DAG) [9]. REANA currently supports either the *Common Workflow Language* (CWL) format, or the *Yadage* format. These workflows may reach high complexity levels in real world use-cases.

The following example shows a typical computational workflow used in the *BSM search* analyses:

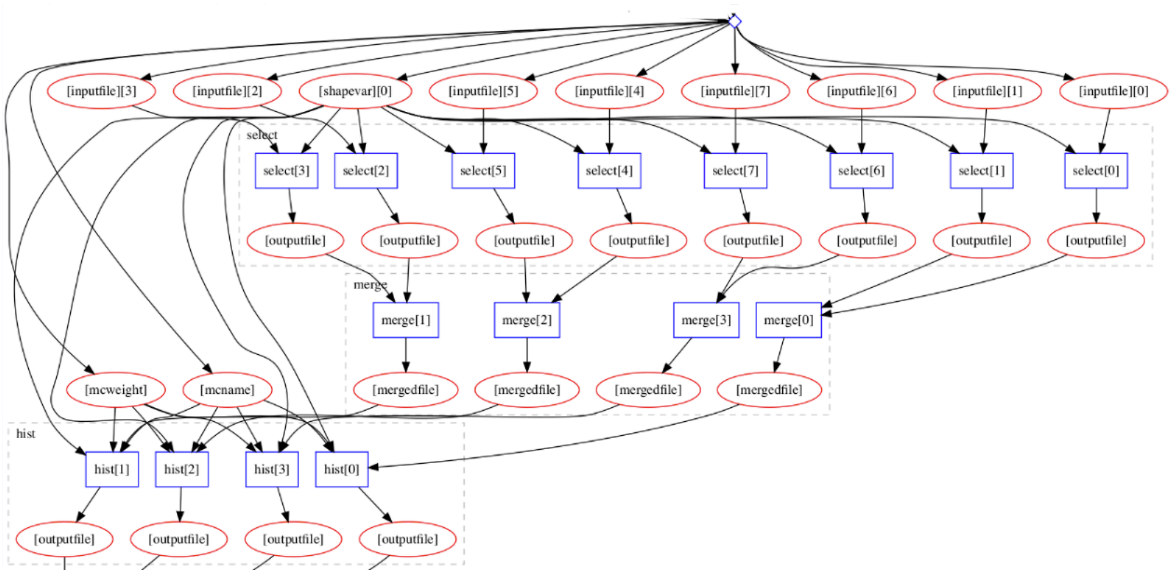

Figure 8: Part of a computational graph used in BSM search

Regarding the visualization of these workflows, **Vis.js [10]** was the selected library of choice due to its simplicity and easy integration with React. A simple example of how to define a graph in Vis.js can be seen in the following example figures:

```
graph: {
    nodes: [
       { id: 1, label: "Node 1" },
       { id: 2, label: "Node 2" },
       { id: 3, label: "Node 3" },
       { id: 4, label: "Node 4" },
       { id: 5, label: "Node 5" }
     ],
     edges: [
       { from: 1, to: 2 },
       { from: 1, to: 3 },
       { from: 2, to: 4 },
       { from: 2, to: 5 }
     ]
}
```

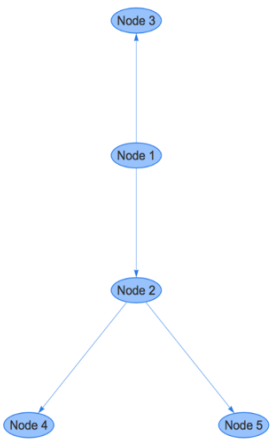

Figure 9: Example of a simple computational graph

3.4. ADDITIONAL TECHNOLOGIES

In addition to the main technology decisions, there were additional packages needed to support extended web-interface functionalities. These packages are:

- ***Axios:*** to perform HTTP requests to the deployed server. Axios handles promises in a better manner than the default *"fetch"* method from React.
- ***Lodash:*** to provide sorting functionality over an array of objects.
- ***Universal-cookies:*** to provide session persistency after a user has logged in.
- ***History***: to ensure browser navigation, and to create our own custom navigation between pages.

4. DEVELOPMENT

4.1. LOGIN PAGE

The first page of the user interface dashboard is the login page. Given that this project is in a private pilot state, users can only register via invitation: they will need to send an email to the project administrator (info@reana.io) and ask for a token.

The received token is used as a user account associated password. It is mainly used to obtain *JSON Web Tokens* [11] (RFC-7519) which provide additional security over common tokens due to their temporary nature (JWT tokens expire after a certain time).

The fully working prototype of the *login* page can be seen in the following figure:

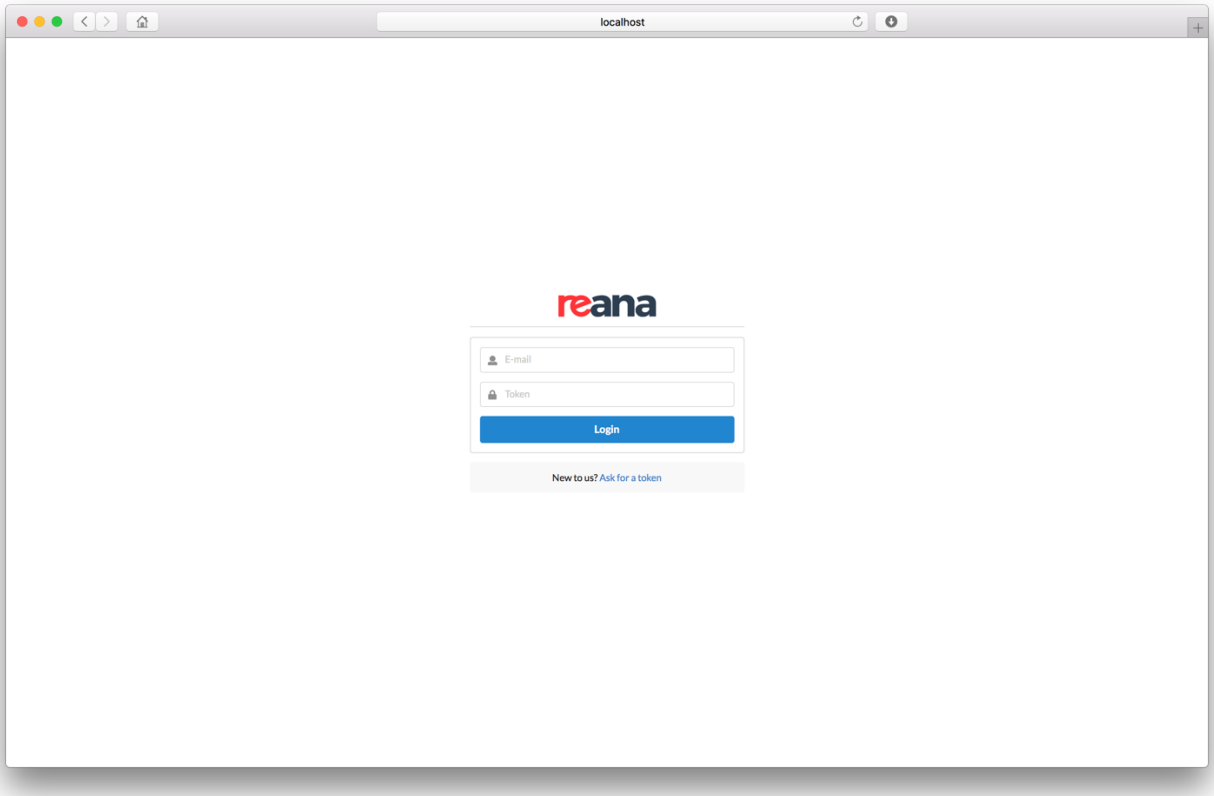

Figure 10: Screenshot of implemented login page

4.2. WORKFLOW LIST PAGE

This page will function as a home after a user has logged in. It contains information about the list of workflows that the user has created together with their status:

- The name and run.
- The creation date.
- The duration.
- The progress (updated every 5 secs).
- The current status.
- The list of available actions.

The fully working prototype *workflow-list* page can be seen in the following figure:

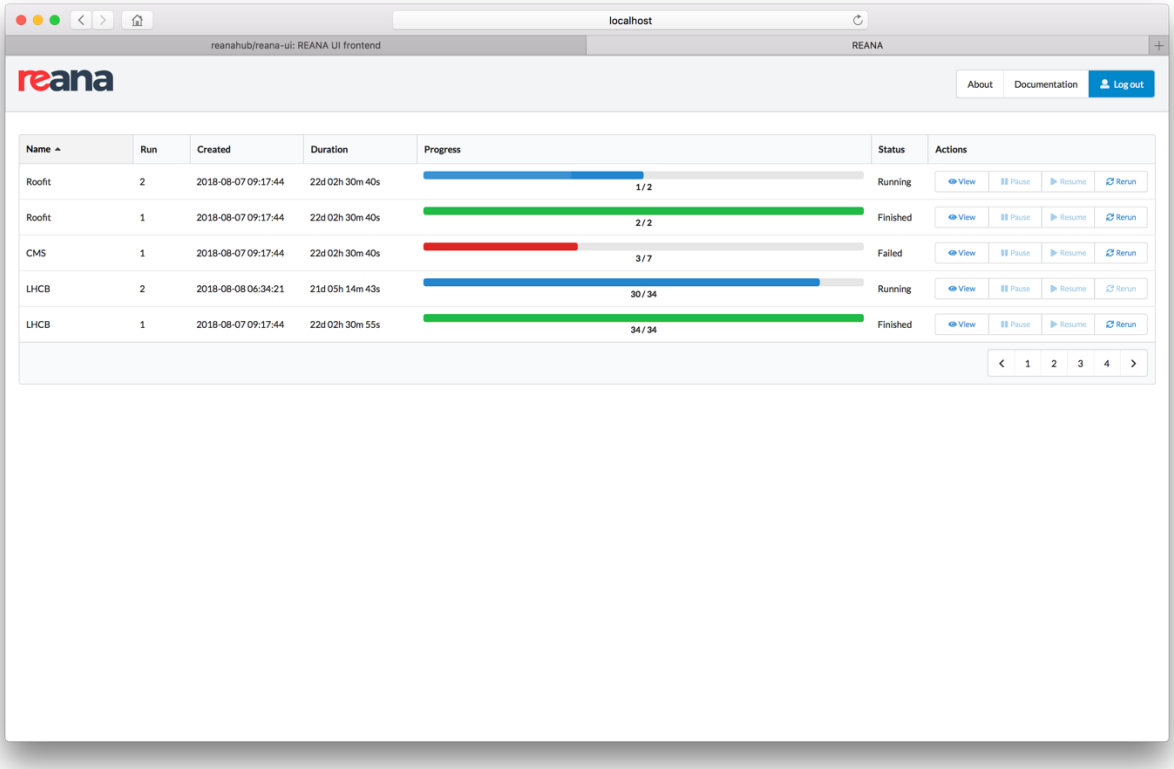

Figure 11: Screenshot of implemented workflow-list page

In terms of **further improvements** to this current prototype page:

1. The progress update could be done over web-sockets instead of simple pooling. It will require changes in the *REANA-Server* module.

2. The pagination at the footer of the list depends on server-side pagination future implementation (GitHub issue [12]).

4.3. WORKFLOW DETAILS PAGE

The workflow details page is the most complex one of the three. It contains the detailed workflow overview information, including:

- The workflow basic information:
  - The name.
  - The run.
  - The creation date.
  - The current status.
- The workflow steps graph.
- The workflow input files.
- The workflow output files.
- The workflow logs.

The fully working prototype *workflow-details* page can be seen in the following figure:

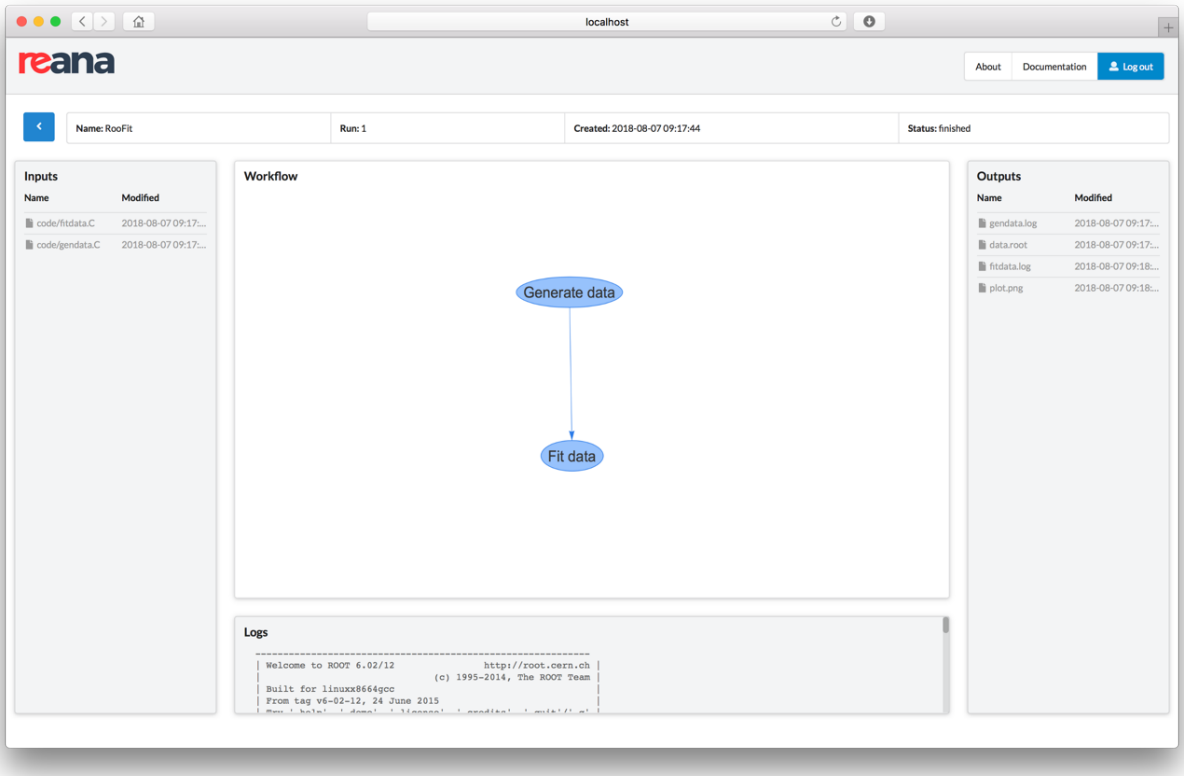

Figure 12: Screenshot of implemented workflow-details page

Additionally, there exists an online file visualization feature. Once a file is selected, the user can see its content and download it to their local computers.

The online file visualizer can be seen in the next figure:

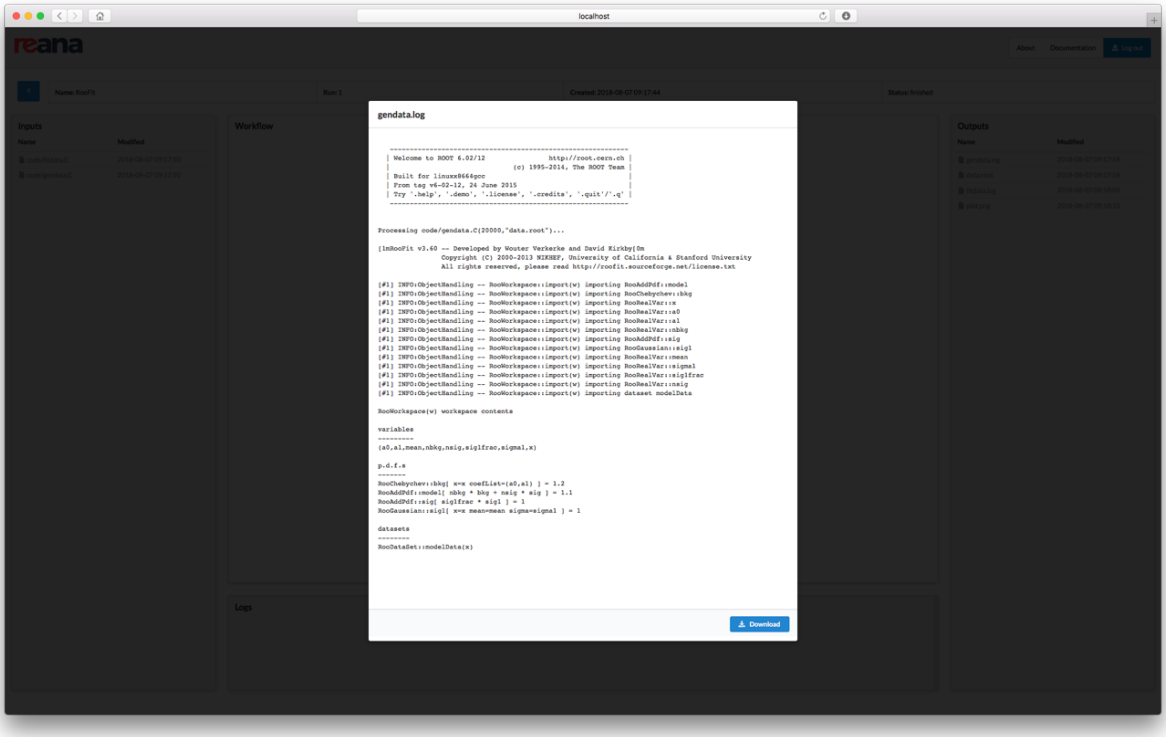

Figure 13: Screenshot of online file visualization

In terms of **further improvements** to this current prototype page:

1. The ability to filter both the list of files and the logs depending on the selected workflow step. It will require changes in the *REANA-Server* module.
2. The ability to expand and collapse nodes from the workflow graph visualization. It would be really useful for complex graphs.

5. CONCLUSION

The present work delivered a first fully functional prototype of the REANA user dashboard. The newly developed *REANA-UI* module, open sourced on GitHub **[3]**, offers users an easy interactive way to inspect and visualize their running workflows over a web interface.

The developed component sets the basic architecture of the REANA web frontend, validating the technology choices and creating a robust foundation for implementing further improvements and future user interface needs.

6. REFERENCES

1. Baker, Monya. *"1,500 scientists lift the lid on reproducibility"*. 25 May 2016. https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970
2. Reusable Analysis platform, *REANA-Client* component. https://github.com/reanahub/reana-client
3. Reusable Analysis platform, *REANA-UI* component. https://github.com/reanahub/reana-ui
4. Reusable Analysis platform, *REANA-Server* component. https://github.com/reanahub/reana-server
5. Storybook community. *StoryBooks* framework. https://github.com/storybooks/storybook
6. Grommet. *Grommet* framework. http://grommet.io
7. Material-UI Organization. *Material-UI* framework. https://material-ui.com
8. Semantic Organization. *Semantic-UI* framework. https://react.semantic-ui.com
9. Wikipedia, Wikimedia Foundation. "Directed Acyclic Graph". https://en.wikipedia.org/wiki/Directed_acyclic_graph
10. Almende. *Vis.js* visualization framework http://visjs.org
11. Auth0. JSON Web Tokens (RFC-7519). https://jwt.io
12. Reusable Analysis platform. *"reana-ui"* repository, issue number 12. *"implementation: pagination"*. https://github.com/reanahub/reana-ui/issues/12