# easyDMP Report

easyDMP is a data management plan (DMP) tool developed on Django Framework (https://www.djangoproject.com/). Django is a high-level Python web framework with Object-Relational Mapping (ORM) capabilities. This framework was chosen to have a state-of-the-art web app framework easily maintanable by researchers. easyDMP primary purpose is to help the laboratories integrate their DMPs within the overarching Digital ECOSystem (DECOS) architecture.

1. Main Features

The main features of easyDMP are:

1. Management the Data Management Plan of the experiments and the laboratories
2. FAIR Integration of the Data Management Plan in the metadata schema of the laboratories, including instrumentation and electronic notebook.

To achieve the first point, easyDMP is composed by two tools, a Core Data Management Plan form to create and update the DMP for the single laboratory, and a Sample Entry form to track samples and experiments.

Core Data Management Plan: this is a form page (example shown in Figure 1) that links the laboratory to the core data management plan tables, it could be created and edited by the user of the laboratory. It tracks the general aspects of metadata collection, publications and storage policies abided by the laboratory. While, by itself, it is a modifiable living document, the form questions are designed to be static, and are the same for every laboratory inside DECOS.

A screenshot of a computer

Description automatically generated

Figure 1: Core Data Management Plan Form example

Sample Entry: this form acquires the information when the sample is received (or it is created) by the laboratory. Successfully compiling this form create a specific DMP entry to manage the single experiment (identified with a service\_request unique code). As opposed to the Core Data Management Plan form that has the same items for every laboratory (albeit, of course, not the same answers), the “Sample Entry” questions are customized laboratory for laboratory, in collaboration with the researchers. The main objective, when designing the form, is to answer the question “What do I need, as a researcher, to know about my sample to do the experiment?”. In FIGURE 2 is shown a Sample Entry form for the Next Generation Sequencing laboratory (i.e., Laboratorio di Genomica ed Epigenomica, Area Science Park, Italy).

A form with a sample entry

Description automatically generated with medium confidence

Figure 2: Sample Entry Form partial example (Customized for Laboratory of Genetics and Epigenetics)

In addition, Sample Entry has a dedicated subsection that inquire about the instrumentation and the techniques required to manage the experiment.

Laboratory Report: the site has a dedicated search functionality to review and generate printable reports of past experiment DMPs, as shown in FIGURE 3.

A close-up of a report

Description automatically generated

Figure 3: Report page example

Switch Lab, and Selected Lab features: a User could be associated with multiple laboratories, to clearly specify the laboratory scope easyDMP implements a functionality to switch between different laboratories, if they are within the User roles.

1. FAIR data management architecture:

### Databases

The back end of the Django Framework was modified to include a database routing function, the rationale is to separate the service database (named “easydmp\_main”), used by the front-end web-app, from the shared metadata storage database (named “decos\_metadata\_db”), that is the core of the PRP metadata integration.

decos\_metadata\_db: this is the shared DECOS database created with an ad-hoc Schema implemented by the PRP ontology. In Figure 4 is shown the first version of the database schema. This database is made to be accessed by other scripts, instrumentation, software or application within the DECOS, and has two different scopes: store information on DMPs, file metadata and map this information in a well thought (and upgradable) singular ontology (named PRP-CDM, PRP Common Data Model). This database is made to satisfy the Findable, Accessible and Interoperable aspects of the FAIR principles.

A diagram of a computer program

Description automatically generated

Figure 4: decos\_metadata\_db entity relationship diagram, version 1.0.

### easyDMP Apps

easyDMP is composed by two apps with different scopes:

Home - this app hosts the front-end of the easyDMP, it was built using the Wagtail Content Management System (Wagtail CMS, https://wagtail.org/) and its primary function is to display and serve the views and the various pages dedicated to the users and their laboratories.

The Wagtail CMS has a well-defined administration dashboard (shown in Figure 5), and it give users (with elevated privileges) the capability to customize some aspects of the frontend of the web-app, and, if needed, create pages with specific information to be shared by other users.

Wagtail CMS could also manage user roles, linked also with the various laboratory. A single user could be associated in groups of various laboratories, and the visibility of specific pages could be managed by the admin dashboard, without the intervention of software developers.

PRP-CDM-App - this app is based on the native Django Framework. The main function of this app is to manage the overarching ontology of the PRP project, and it hosts the Object–relational mapping of the entire PRP schema. A developer could easily modify the model of this app to implement a new version of the overarching ontology, writing table and relations in Django ORM, while the database management software of Django will adapt the old database with the new schema during the migrations. This app also includes an object-oriented schema of the personalized *Sample Entry* forms, the interaction between PRP-CDM-App and Home are shown in Figure 6.

A screenshot of a computer

Description automatically generated

Figure 5: easyDMP Wagtail Admin page, home page customization tool

A diagram of a data flow

Description automatically generated with medium confidence

Figure 6: PRP-CDM-App partial class diagram, detailing the interaction with Home and its FormFactory pattern to create customized Sample Entry forms mapped into the PRP-CDM schema. FormOrchestrator checks the user selected laboratory and instructs the form factory to create all the form fields needed defined in a laboratory specific subclass of FormDefiniton. FormDefiniton defines the tables used in a form of a specific laboratory.

## Additional features:

Electronic Notebook REST API Integration (with elabFTW electronic laboratory notebook, https://www.elabftw.net/): if a user registers the elabFTW API keys (with write and read permissions) on *Home*, it is possible to integrate the Sample Entry page with experiment creation on elabftw, specifically on prp-electronic-lab.areasciencepark.it, making these tools interoperable.

Single Sign-On: easyDMP uses the module AllAuth (https://docs.allauth.org/en/latest/) to integrate the login/logout process with Authentik (https://goauthentik.io/).

Form customization: To improve usability the dynamic forms linked to the decos\_metadata\_db could be personalized using simple HTML tags created by the researchers as shown in Figure 7, the system will bypass the automatic form creator and delivers a personalized page (Figure 2).

*A screen shot of a computer program

Description automatically generated*

Figure 7: partial personalized HTML form code

File Management: easyDMP could manage the upload and download of files with a dedicated and customizable File Management System compatible also with S3 Buckets. If the form requires file upload (usually small csv, pdf or xls), easyDMP creates ad-hoc and DMP compatible folders. The path to the file is stored into decos\_metadata\_db in the laboratory specific table.