

Improving the Web Interface for the ILD Data Catalogue

Viacheslav Filimonov, Moscow Institute of Electronics and Mathematics

September 2011

Abstract

In this report the steps of improving the Web Interface for the ILD Data Catalogue will be described. These steps consist of database model choice, framework choice, creating public web interface with this framework. All the development decisions will be explained. Results will be reviewed.

Contents

1. Introduction	4
2. Database schema choice	4
3. Framework choice	5
4. Modification	6
5. Navigation buttons	6
6. View files' list buttons	7
7. Clear selection	8
8. Adding new request	8
9. Results	11
A. Appendix	14
A.1. Project steps	14
A.1.1. To install Django 1.3:	14
A.1.2. To remove action bar from change list:	14
A.1.3. To delete the row of buttons for delete and save:	14
A.1.4. Overriding the templates:	14
A.1.5. To remove delete checkboxes:	14
A.1.6. To remove add button:	14
A.1.7. To remove recent actions module:	15
A.1.8. To change 'change' to 'view' in the start menu:	15
A.1.9. Note	15
A.1.10. To remove the history button:	15
A.1.11. Integrating Django with a legacy database:	15
A.1.12. At first making admin page only with Gfile model:	15
A.1.13. To add filters and search bar:	15
A.1.14. To remove add button from the change list:	16
A.1.15. To remove add button from the start menu:	16
A.1.16. Working with Gfile:	16
A.1.17. Adding Rfile and Sfile:	16
A.1.18. Navigation:	16
A.1.19. To create view:	17
A.1.20. Adding a button:	17
A.1.21. To replace 'Site administration' in the first menu with 'Browsing MCDB':	18

A.1.22. To replace 'Select to change' in the change list with 'Select to view':	18
A.1.23. To move Filter bar to the left side of the change list:	18
A.1.24. A button and a corresponding view to display a list of all files was created:	18
A.1.25. To change the names of the objects on the admin page:	18
A.1.26. To remove Auth application from the main menu:	18
A.1.27. Not to hardcode urls in buttons:	19
A.1.28. To remove past 7 days from the filter box:	19
A.1.29. Adding a request:	19
A.1.30. 'Clear selection' button:	19

1. Introduction

ILD Data Catalogue is used for data production. The data concerning generation, simulation and reconstruction of events is kept in this Catalogue. The database of this Catalogue has a certain schema. This database schema provides effective Catalogue browsing.

The main aim of my project is to make a public Web Interface for the ILD Data Catalogue. This public Web Interface should be suitable for effective Catalogue browsing. And to be suitable for effective Catalogue browsing this public Web Interface should be created with taking into account the database schema. And it's also important to mention that it's necessary to create an appropriate design for the public page and to provide it with essential functions.

2. Database schema choice

In data production there are three main steps: generation, simulation and reconstruction of events.

Due to the chaining of the generation, simulation and reconstruction steps, it's not viable to put all the information in one place. So we need to add an interaction for effective browsing of data.

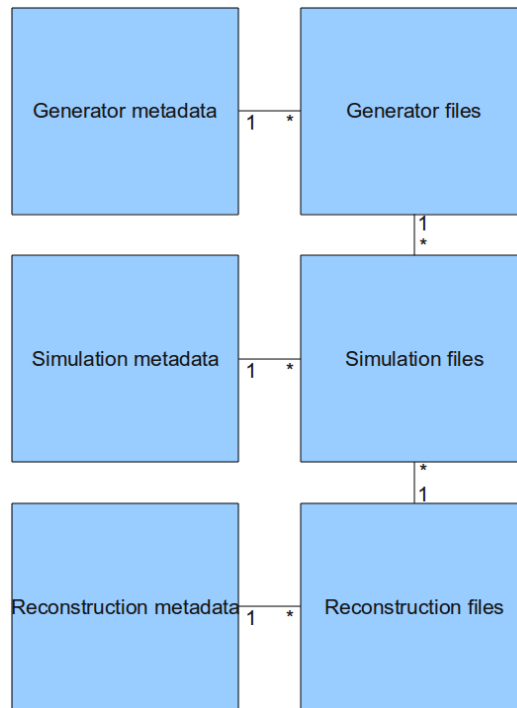


Figure 1: database schema

This database schema (fig. 1) provides this interaction. You can browse through this tables with the help of relations provided.

It's also important to mention that with the help of this database schema it's possible to reduce redundancy. For example, all generator files share the same Generator metadata. The same situation with simulation and reconstruction files. So if redundancy is low the database queries will be very fast. And in this situation it's essential due to the big amount of data.

3. Framework choice

To solve this interaction problem Django was chosen as a web framework. Because Django provides an admin page that has many nice navigation tools (fig. 2). The most important one is the possibility to browse the database. There are also filters, search bar, date hierarchy and possibility to sort tables' contents by column value. On figure 2 you can also see an example of view files' list and clear selection buttons that I had to add myself to improve the interface. But about improvements and modifications it will be said later.

The screenshot shows the 'MCDB Site' admin interface. At the top, there's a navigation bar with 'Home', 'Pub', and 'Generator files'. Below it, a 'Browsing MCDB' section contains a 'View gen files' list and a 'Clear selection' button. A search bar is also present. The main table displays a list of generator files with columns: Filter, Process, Generator, Energy, Luminosity, Cross section, Pol ep, Pol em, Series, Total events, Gen file, and Created. The table is filtered by 'By process' and shows several rows of data. The 'Created' column shows dates like 'Aug. 18, 2011, 2:35 p.m.'.

Filter	Process	Generator	Energy	Luminosity	Cross section	Pol ep	Pol em	Series	Total events	Gen file	Created
All	bbn3e3e2n2	Whizard	500.0	0.0	19.0238	1.0	-1.0	w19369_fixed2	19025	lfn:/grid/lc/mc-2008/generated/CMS_500/w19369_fixed2_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbcsdu	bbn3e3e2n2	Whizard	500.0	1000.06	19.0238	1.0	-1.0	w19369_fixed	19025	lfn:/grid/lc/mc-2008/generated/CMS_500/w19369_fixed_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbcsen1	bbn3e3e2n2	Whizard	500.0	1000.06	19.0238	1.0	-1.0	w19369	19025	lfn:/grid/lc/mc-2008/generated/CMS_500/w19369_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbcsen2	bbn3e3e2n2	Whizard	500.0	0.0	19.1229	1.0	-1.0	w19357_fixed2	19124	lfn:/grid/lc/mc-2008/generated/CMS_500/w19357_fixed2_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbcsen3	bbn3e3e2n2	Whizard	500.0	1000.06	19.1229	1.0	-1.0	w19357_fixed	19124	lfn:/grid/lc/mc-2008/generated/CMS_500/w19357_fixed_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbn1e1e2n2	bbn1e1e2n2	Whizard	500.0	1000.03	19.1375	1.0	-1.0	w17697	19138	lfn:/grid/lc/mc-2008/generated/CMS_500/w17697_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbn1e1e3n3	bbn1e1e3n3	Whizard	500.0	0.0	57.0304	1.0	-1.0	w19389_fixed2	57031	lfn:/grid/lc/mc-2008/generated/CMS_500/w19389_fixed2_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbn3e3e2n2	bbn3e3e2n2	Whizard	500.0	1000.01	57.0304	1.0	-1.0	w19389_fixed	57031	lfn:/grid/lc/mc-2008/generated/CMS_500/w19389_fixed_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbcsdu	bbcsdu	Whizard	500.0	1000.02	164.701	1.0	-1.0	w17785	29687	lfn:/grid/lc/mc-2008/generated/CMS_500/w17785_01.stdhep	Aug. 18, 2011, 2:35 p.m.
bbcsdu	bbcsdu	Whizard	500.0	1000.02	164.701	1.0	-1.0	w17785	67526	lfn:/grid/lc/mc-2008/generated/CMS_500/w17785_02.stdhep	Aug. 18, 2011, 2:35 p.m.
bbcsdu	bbcsdu	Whizard	500.0	1000.02	164.701	1.0	-1.0	w17785	67491	lfn:/grid/lc/mc-2008/generated/CMS_500/w17785_01.stdhep	Aug. 18, 2011, 2:35 p.m.

Figure 2: Change list form in admin page with search bar, date hierarchy and filters provided

And few words about Django in general. Django is an open source web application framework, written in Python, which follows the model-view-controller architectural pattern. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of DRY (Don't Repeat Yourself). Python is used throughout, even for settings, files, and data models.

Django also provides an optional administrative CRUD (create, read, update and delete) interface that is generated dynamically through introspection and configured via admin models. And this interface was used as a basic step in my project.

4. Modification

But there are also two disadvantages: admin page does not have read-only user and provided navigation tools are only for browsing single tables. So it was necessary to modify the admin page and to turn it into a public page. To remove save, delete, add, history and other buttons that can't be in public page, change some labels to make them more suitable for public page it was necessary to override the corresponding templates. To make fields of the objects read-only a special django property read-only fields was used. But to use this property it was necessary to update django to version 1.3. And also some css files were modified to change the design of the page.

To solve the second problem it was necessary to add navigation buttons for establishing relationship between different tables. And also special buttons for viewing files' lists in copy-paste format were added. And special buttons for adding a request were added. I must also say that request includes registration of new user if it's necessary.

5. Navigation buttons

One of the main improvements are navigation buttons. Using this buttons you can view corresponding files or metadata. It means that if you want to see corresponding files or metadata for a certain file you will have to click an appropriate navigation button and you will view a page with already preselected corresponding files.

MCDB Site

Welcome, **slawof**. [Change password](#) / [Log out](#)

Home » Pub » Generator metadata » GenMetadata object

Browsing MCDB

Gen id:	15
Process:	bbn3e3e2n2
Generator:	Whizard
Energy:	500.0
Luminosity:	0.0
Cross section:	19.0238
Pol ep:	1.0
Pol em:	-1.0
Series:	w19369_fixed2
Cut:	(None)
Gen tevents:	19025
Tfiles:	1

[View corresponding generator files](#)

[View corresponding simulation files](#)

[View corresponding reconstruction files](#)

[View corresponding reconstruction metadata](#)

[View corresponding simulation metadata](#)

Figure 3: Navigation buttons for generator metadata object

It's also necessary to mention that navigation buttons provide interaction between tables. So you can browse the database through different tables using database schema above (fig. 1).

6. View files' list buttons

Another useful improvement are view files' list buttons. That you can use to view corresponding files' lists in copy paste format (fig. 4). And it's also necessary to mention that content of these lists depends on selected files. It means that if you selected some files by filters, date hierarchy or navigation buttons, the list will consist only of corresponding content. An example of view files' list button you can see in the upper left corner of the figure 2.

Log files

```
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0001-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0002-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0003-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0004-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0005-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0006-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0007-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0001-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0002-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0003-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0004-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0005-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0006-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0007-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw19389.fixed2.Pbbcselnl.e-1.p1-0001-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw19389.fixed2.Pbbcselnl.e-1.p1-0002-REC.tar.gz
lfn:/grid/l1c/generated/engels/prd/logs/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw19389.fixed2.Pbbcselnl.e-1.p1-0003-REC.tar.gz
```

Dst files

```
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0001-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0002-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0003-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0004-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0005-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0006-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0007-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0001-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0002-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0003-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0004-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0005-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0006-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0007-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw19389.fixed2.Pbbcselnl.e-1.p1-0001-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw19389.fixed2.Pbbcselnl.e-1.p1-0002-DST.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw19389.fixed2.Pbbcselnl.e-1.p1-0003-DST.slcio
```

Rec files

```
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0001-REC.000.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0002-REC.000.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0003-REC.000.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0004-REC.000.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0005-REC.000.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0006-REC.000.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17785.Pbbcsdu.e-1.p1-0007-REC.000.slcio
lfn:/grid/l1c/generated/engels/prd/data/v01-11/rec/rv01-11.sv01-11.mILD_00.LQGSP_BERT.E500.Iw17773.Pbbcselnl.e-1.p1-0001-REC.000.slcio
```

Figure 4: Files' lists

7. Clear selection

Example of clear selection button you can see in the upper left corner of the figure 2 next to the view files' list button. During browsing the database user can clear the results of selection made by filters, date hierarchy, search bar or navigation buttons with the help of this button. So it's easy to start selecting necessary files from scratch.

8. Adding new request

With the help of add new request button (fig. 5) user can add a request for reconstruction or simulation of events.

MCDDB Site
Welcome, **slawaf**. Change password / Log out

Home · Pub · Reconstruction metadata · RecMetadata object

Browsing MCDDB

Add new request

Id:	8
Gen id:	15
Sim id:	1
Rec id:	8
Process:	bbn3e3e2n2
Generator:	Whizard
Energy:	500.0
Luminosity:	0.0
Cross section:	19.0238
Pol ep:	1.0
Pol em:	-1.0
Series:	w19369_fixed2
Cut:	(None)
Ssw:	ilicsoft
Detector model:	ILD_00
Physics list:	QGSP_BERT

Figure 5: Change list form with add new request button

To add a request user has to enter his username and number of events in a special form (fig. 6). If the user already has a username, his request will be accepted, add to the database and emails with request information (username, type of request, number of events and file contents) will be sent to the database administrator and to the user. And this user will be redirected to the appropriate change list form to have a possibility to add a new request or continue browsing the catalogue.

User name:

Number of events:

If you are not registered please register here

Figure 6: Form for adding new request

But if this user doesn't have a username yet, he can click register here button to be redirected to the registration page (fig. 7). And also if a username is wrong this user will be redirected to the registration page.

You are not registered. Please register here.

First name:

Last name:

Email:

Institute:

Virtual organization:

Phone:

User name:

Submit

Figure 7: Registration page

Registration page contains such fields as: first name, last name, email, institute, virtual organization, phone, user name. If user fills this form correctly, he will receive a username and will be redirected to the add request form to add his request. And registration information will be add to the database.

9. Results

So now we have a public Web Interface for the ILD Data Catalogue that includes possibility to browse the database through different tables (with navigation buttons), filters, search bar, date hierarchy and possibility to sort the tables' contents by column value. There are also view files' list buttons that you can use to view corresponding files' lists in copy paste format.

With the help of clear selection buttons user will have a possibility to start selecting necessary files from scratch.

And another important improvement is possibility to add a new request. This possibility also includes possibility to register.

References

- [1] Python Web Development with Django (1st ed.), Addison-Wesley, *Forcier, Jeff; Bissex, Paul; Chun, Wesley*
- [2] The Definitive Guide to Django: Web Development Done Right (2nd ed.), Apress, *Holovaty, Adrian; Kaplan-Moss, Jacob*
- [3] Practical Django Projects (2nd ed.), Apress, *Bennett, James*

A. Appendix

A.1. Project steps

It's better to use Django 1.3 due to read-only fields property.

A.1.1. To install Django 1.3:

1. Install zlib module (to use setuptools)
2. Install libmysqlclient module (to use MySQLdb)
3. Install setuptools module (to use MySQLdb)
4. Install MySQLdb (python module)
5. Install Python compiled with zlib

A.1.2. To remove action bar from change list:

`AdminSite.disable_action(name)` should be used.

`admin.site.disable_action('delete_selected')`. You have to write it in `admin.py`.

A.1.3. To delete the row of buttons for delete and save:

`admin_modify.py`

should be used.

A.1.4. Overriding the templates:

It's necessary to write the path to the templates in `settings.py` and copy this template to the local admin dir.

A.1.5. To remove delete checkboxes:

You have to override and modify `tabular.html`.

A.1.6. To remove add button:

You have to delete string

```
{{ inline_admin_formset.formset.management_form }}
```

from the `tabular.html`.

A.1.7. To remove recent actions module:

It's necessary to override template index.html, removing

```
{% block sidebar %}
```

from it.

A.1.8. To change 'change' to 'view' in the start menu:

With the help of index.html.

A.1.9. Note

It is still possible to add and edit objects using Django API.

A.1.10. To remove the history button:

By using

```
change_form.html.
```

A.1.11. Integrating Django with a legacy database:

1. New project is in pub folder.
2. Editing settings.
3. Running inspectdb.
4. Editing models.py:
 - 1)Rearrange models' order
 - 2)Make sure each model has one field with

```
primary_key=True
```

A.1.12. At first making admin page only with Gfile model:

To display Gfile on the admin page don't forget to include pub app to the

```
INSTALLED_APPS
```

in settings.py

A.1.13. To add filters and search bar:

Use admin.py.

A.1.14. To remove add button from the change list:

You have to override and modify the
`change_list.html`
template.

A.1.15. To remove add button from the start menu:

You have to override and modify the `index.html` template.

A.1.16. Working with Gfile:

- 1.To display Gfile's fields in a change list use list display.
- 2.Use

`date_hierarchy, list_filter, search_fields`

to add date hierarchy, filters and search bar.

- 3.Use

`readonly_fields`

to make fields read-only in change form.

A.1.17. Adding Rfile and Sfile:

The same situation as with Gfile.

A.1.18. Navigation:

1. To add rec metadata view to replace the software id in the rec table with the name and version of software.
2. To add sim metadata view to replace the software id in the sim table with the name and version of software.
3. To add gen metadata view to make names of the objects similar.
4. Afterwards to add all this models to the admin page.

A.1.19. To create view:

1. Write a sql statement in the

`db_model/mcdb-objects.sql`

2. Run a command:

```
mysql mcdb < db_model/mcdb-objects.sql
```

3. Refresh the models2.py by running command:

```
python manage.py inspectdb > pub/models2.py
```

from the mcdb directory

4. Not to forget about

```
primary keys = true
```

A.1.20. Adding a button:

Main aim is to add button so that it would be possible to use this button to view all the corresponding files for a certain file.

Add a link button to the

`change_form.html`

template:

Have to make a dynamic link.

To connect gen metadata and sfile we have to add generator id to the sfile view (using mcdb objects). So it will be possible to use a button in gen metadata change form to enter an sfile change list with corresponding lines.

Now it's necessary to find gen id in a change form page to enter an sfile change list with corresponding lines. It's possible by parsing the request url (split()).

Now when button is clicked user can see the corresponding sfiles that are chosen by filter.

The same operation for all other files.

It was necessary to make if constructions to avoid wrong buttons in every form.

It works with the

```
{% if opts.object_name.lower == "genmetadata" %}
```

condition, for example.

Button activates the view (via urls.py) and the view redirects user to the page of pre-elected corresponding files.

Gen id is taken from the object and object id is taken from the url (splitting).

If constructions were used in in views.py to match the corresp objects; objects' names were taken from the url (splitting).

A.1.21. To replace 'Site administration' in the first menu with 'Browsing MCDB':

Using site.py (found by grep).

A.1.22. To replace 'Select to change' in the change list with 'Select to view':

Using views/main.py (found by grep).

A.1.23. To move Filter bar to the left side of the change list:

With the help of changelists.css.

A.1.24. A button and a corresponding view to display a list of all files was created:

This button was created on the change list for sfiles (if construction).

A list of all files was modified to the copy paste format.

For if construction:

```
cl.opts.verbose_name_plural
```

View for all files was modified to display only corresponding files (files that were pre-selected by the filters, date hierarchy, navigation buttons).

Problem with filtering objects was solved with `**d` argument for passing dictionaries to the function to match it's parameters. Parameters were got by parsing the request url (`split()`).

The same buttons were added for gfiles and rfiles.

Problems with 'by created' filter: when the information from this filter (it's url) is used in the views.py to match the corresponding objects, the format of the date in the url and in the model fields is different. Was solved by catching an error.

A.1.25. To change the names of the objects on the admin page:

With the help of

```
verbose_name_plural
```

in the models.py class meta.

It's necessary to modify if constructions in

change_list.html template like this

```
{% if cl.opts.verbose_name_plural == "Reconstruction files" %}
```

A.1.26. To remove Auth application from the main menu:

By commenting the auth application in settings.py.

A.1.27. Not to hardcode urls in buttons:

Use

`/../../`

in html templates and url parsing in views.

A.1.28. To remove past 7 days from the filter box:

Use `filterspecs.py`.

A.1.29. Adding a request:

A button for requests was added to the change form.

Button redirects user to request form filled by user. Afterwards information from request is passed to the database or user is redirected to the registration page. Then email is sent to the database administrator automatically.

When user fills request form he is redirected to the appropriate change list form. It's possible because information about chosen object is passed through all the views (parsing `url + request.Post`) and html templates (type hidden).

Forms for registration and adding request are written in html. They are available via `views + urls.py`.

Emails are sent with the help of "send mail" function provided by django. Information from request and registration form is passed to the database with the help of sql procedures, that can be called from django views with the help of `cursor.execute()`.

A.1.30. 'Clear selection' button:

Parses url and redirects user to the page without filtering.