

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
EP-CMX-DA

Priit Paluoja

**Database with web interface and search engine as a
diagnostics tool for electromagnetic calorimeter**

CERN Summer Student Report

Supervisors: Giacomo Cucciati
Davide Valsecchi

Geneva 2017

Database with web interface and search engine as a diagnostics tool for electromagnetic calorimeter

Abstract

During 2016 data collection, the Compact Muon Solenoid Data Acquisition (CMS DAQ) system has shown a very good reliability. Nevertheless, the high complexity of the hardware and the software involved is, by its nature, prone to some occasional problems. As CMS sub-detector, electromagnetic calorimeter (ECAL) is affected in the same way.

Some of the issues are not predictable and can appear during the year more than once such as components getting noisy, power shortcuts or failing communication between machines. The chain detection-diagnosis-intervention must be as fast as possible to minimise the downtime of the detector.

The aim of this project was to create a diagnostic software for the ECAL crew, which consists of database and its web interface that allows to search, add and edit the contents of the database.

1. Introduction

The high complexity of the hardware and the software involved in ECAL is sometimes prone to occasional problems. To minimise the downtime of the detector, author task was to create software solution that allows to store, access and edit ECAL related diagnostics data.

Project started with mapping of the use cases. Author mapped all the tasks with his supervisors.

Software solution had to have:

1. **Web** access from inside the CERN.
2. A **database** where all the details of the issues can be stored, edited and created with the user interface.
3. A **search engine** to query data from the database.
4. A **sorting system** to extract the information required.
5. A multimedia platform for **image hosting**.
6. A user **authentication**.

Author tried to develop software as generic as possible, trying not to limit its usage only to the ECAL team. Solution was tested by the author and his supervisors. Several problem statements (with attachments) were inserted and no bugs were detected.

The solution consists of 42 Java classes with total of 2260 lines, where 1564 are source code lines, 353 comments and 343 blank lines. There are 10 HTML files with total of 927 lines of code. Project uses 15 external back-end libraries.

Solution is accessible from <https://ecaldoctor.web.cern.ch/>.

2. Software architecture and technical solutions

Libraries

Author started implementation from choosing the suitable frameworks. Java based **Spring Boot** 1.5.5 was chosen for web framework with **Thymeleaf** template engine. **Bootstrap** with **Vue.js**, **jQuery** and **DataTables** were used for front-end. Different Spring related frameworks such as **Spring Data JPA**, **Spring Security** were also used. Project **Lombok** helped to reduce Java boilerplate code. Author chose **PostgreSQL** for database engine and **Hibernate Search** for search engine. Furthermore, **Caffeine** was used for server-side caching.

Web application architecture

Author designed the web application with four distinct layers, where each of the layer has specific task. It makes the software more modular.

Implemented web application consists of the following four layers:

1. Model (data holding)
2. View (HTML templates)
3. Controller (handles user input and calls service class methods)
4. Service (complex tasks such as search engine logic or saving and manipulating data)

To make software more modular, data transfer objects (DTO) were used to exchange data between service, controller and HTML templates.

Search engine

Author used Hibernate Search to implement database querying. Web application uses two different approaches for searching database. User input **autocompletion** (Figure 1) on search page uses **Levenshtein distance** to find similar results based on pre-defined database columns and its query time is limited in milliseconds to make it efficient. Actual result (Figure 3) set uses Hibernate Search keywords to query database columns and is not time limited.

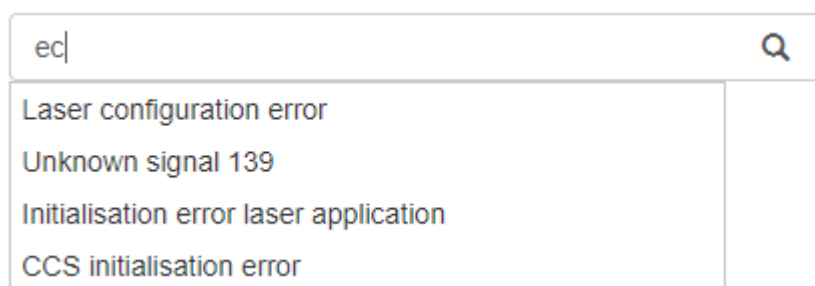


Figure 1 - Autocompletion results based on two tokens "ec".

Caching

The main use case of the software is the querying of the database. To make it as seamless as possible, author activated server-side caching. It is assumed that all the edits of database content are made through the web application, which allows to cache time-consuming operations, which output is deterministic. If database contents are changed, then all the corresponding cache will be cleared. This makes to serve user queries effectively.

Furthermore, all static content such as Cascading Style Sheets (CSS) and JavaScript is minified and the amount of different file count has been minimized to two files per JavaScript and CSS, making the reading of the JavaScript and CSS more continuous. In Spring Boot configuration, the server sends headers for client to cache static content (CSS and JavaScript).

Database

Author chose PostgreSQL as database. Database is hosted by CERN Database on Demand Service. Database schema consists of tables problem, problem_question, question, type, attachment (Figure 4) and person (Figure 2).

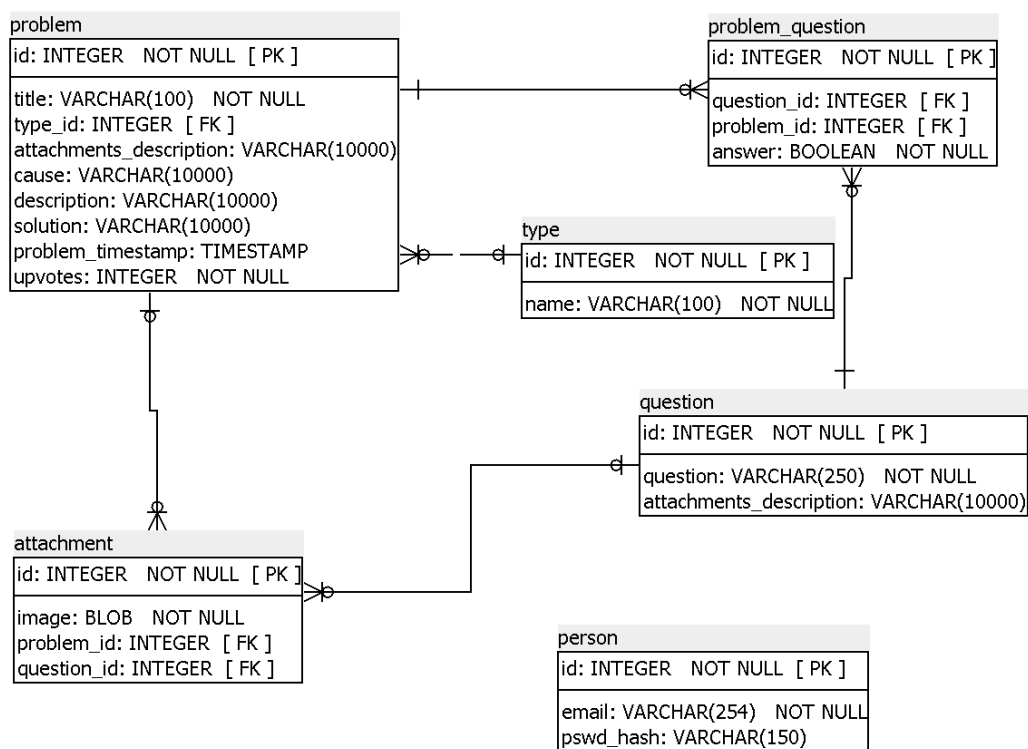


Figure 2 - Database schema of the solution.

Table person is used for **web application authentication** (to add and edit problems, user must be authenticated).

Each problem statement can be linked with a `problem_question` (which is linked with a question).

Questions are used for filtering search results. For example, user can create a problem statement “A” and assign a question “Is the screen red?” with answer “No” to the problem statement. Now searching for it, user can press “Filter” on results page. Interface will then present all the questions that are linked with the search results. If there is a question “Is this screen red?” and user answers “Yes”, then the problem statement “A” will be filtered out. Otherwise (answering “No” or providing no answer) the problem statement will be kept in the search results.

Setup

Web application is hosted by Information Technology Department. They have deployed OpenShift Origin to facilitate deployment of web applications. Author deployed web application in OpenShift running on WildFly 9. Author configured OpenShift project in a way that web application is fetched, built and deployed from GitLab. It is achieved with the help of the Maven, which is used for project management. Furthermore, author configured OpenShift project to use HTTPS and allocated suitable amount of resources (1.5 GB of RAM and 900 ms of CPU) for building and running of the application.

In the idle state, the application uses 620 MiB of RAM and 0.005 cores of CPU.

3. Conclusion

Author created the **database** with the **web interface** as a diagnostics tool for electromagnetic calorimeter that is accessible at <https://ecaldoctor.web.cern.ch/>.

Completed software solution is optimized to be as fast as possible. To achieve this, author used server-side caching and front-end optimization. For server-side caching, **Caffeine** was used.

Software was based on **Spring Boot** web framework with Thymeleaf template engine and PostgreSQL as database engine. Bootstrap with Vue.js, jQuery and DataTables were used for front-end.

Hibernate Search was used for search engine.

The solution allows the user to insert, edit and search information about ECAL related problems and errors. Results can be filtered with questions that are linked to the problem statements. Solution has an **autocomplete** for searching that uses **Levenshtein distance** to find similar results. Main search results are queried by keywords.

Solution runs on WildFly 9 in OpenShift, where **HTTPS** is activated and new version of the application can be automatically fetched, built and deployed from the GitLab with the help of Maven build tool.

Author tried to develop software as generic as possible trying not to limit its usage only to the ECAL team. System can be used as a general search engine for similar problem statements.

Software was **tested** and no bugs were detected.

Appendix

Show	10	entries	Search:				
↓↑	Type/Tag	Title	Description	Solution	Date	Votes	Edit
▼	ECAL	ECAL DCC configuration error	One or more DCC Supervisors go in error during the configuration stopping ECAL. EcalFM reports s...	Power cycle the crate does the trick. Ask to experts to do that but provide them the crate numbe...	26.07.2017	0	Show
▼	ECAL	DCC configuration error	During configuration phase one of the ccs takes ages to configure and then one of the dcc goes in...	Contact Evgueni. In case exclude the corresponding FED.	28.07.2017	0	Show
▼	ECAL	FED blocking the run	A FED blocks the run. It is possible to see that in the DAQ-Expert page. XMAS DCC monitor shows...	If you are running and you cannot remove ecal from the run (like during beam), just remove the FE...	28.07.2017	0	Show
▼	ECAL/ES	Timeout error	The configuration phase takes ages (3 minutes) and then it goes in error. Just before of that yo...	1) Check in the TCDS applications list if there is any in error regarding ECAL or ES. If yes, tr...	28.07.2017	0	Show
▼	ECAL	TCC configuration error	During the configuration one or more TCC supervisors go in error moving the ecalFM in error state...	There is no real solution. Try a Red Recycle first and in the meanwhile check with Evgueni scrip...	26.07.2017	0	Show
▼	ECAL	Laser error during configuration	During configuration phase the ecalFM crashes with an error in the laser machine reporting: "Fa...	Call David Bailleux or laser expert and ask to restart the laser controller script.	28.07.2017	0	Show
▼	ECAL	CCS configuration error	During configuration phase the ecalFM crashes with all or some CCSs in error and reporting someth...	Check from the cmsonline page if the HV/LV is really off. Maybe it is a glitch or the cmsonline t...	28.07.2017	0	Show
▼	ECAL	Error configuring ttcci	Error of the ecalSupervisor crashing the ecalFM during the initialisation. It reports: "No appl...	Choose an other configuration. Is there a reason why you want to use such an old one? If you rea...	28.07.2017	0	Show
▼	ECAL	Sync loss	During the run we have sync loss in a partition of ECAL, you can see it in the daq-expert page: ...	Suggest immediately a stop&start of the run and in the meanwhile check the clock stability The S...	28.07.2017	0	Show
▼	ECAL	Laser configuration error	During the configuration phase the ecalFM crashes reporting an error in ecalod-laser-xdaq: "Unkn...	Call an expert (Evgueni/Jose) or the DCS expert/shifter to turn on the laser crate (S5D03b).	28.07.2017	0	Show

Showing 1 to 10 of 24 entries

[Previous](#)
[1](#)
[2](#)
[3](#)
[Next](#)

Figure 3 - Search results page showing general search results.

