Design and development of process control and management application for pre-clinical laboratories

Ricardo Monteiro[1], Markus J. Maeurer[2,3], Pedro Minguéns Matutino[1], Nuno Domingues[1]

ricardo_luis19@hotmail.com, nasdlxpt@gmail.com, pedro.miguens@isel.pt, nuno.domingues@isel.pt

[1] Instituto Politécnico de Lisboa/ Instituto Superior de Engenharia de Lisboa
Rua Conselheiro Emidio Navarro, 1, 1959-007 Lisbon, Portugal
[2] ImmunoSurgery Unit, Champalimaud Centre for the Unknown, Lisbon, Portugal.
[3] I Medical Clinic, Johannes Gutenberg University of Mainz, Mainz, Germany

Abstract

We report here the design and development a laboratory information management system (LIMS) for a immunotherapy and immunosurgery laboratory. A web app was developed using ReactJS (Javascript), Django (Python) and PostgreSQL. A database schema was designed to allow the user of the app to track samples, and results associated with assays performed on each individual sample and sample derivaties. A role-base access control (RBAC) was used to allow different roles and priviledges in using the app. This approach allowed to build a practical and simple LIMS that is able to store and track result files for assays and is can be easily improved with more features in the future.

Keywords: LIMS (Laboratory Information Management System), Immunotherapy, React, Django, PostgreSQL

Introduction

The use of generic information recording and control software such as Excel is still prevalent in many biomedical research laboratory, pre-clinical and clinical practice. With the advancement of web technologies and databases, laboratory management and control systems (LIMS) based on these technologies have emerged in recent decades to facilitate laboratory management. This work proposes the development of a prototype web application to be applied in the management and control of the workflow of a laboratory dedicated to research and clinical practice in immunotherapy (ImmunoSurgery Champalimaud Foundation). The application was named coley, in honor of William B. Coley, one of the pioneers of immunotherapy. The platform was designed to allow easy access, yet also conform to EMA and FDA regulations using biological materials to develop biologically and clinically relevant processes.

Methods

The process of developing this application is rooted in the utilization of open-source tools, which bring about numerous advantages. To create an engaging and dynamic user interface, the decision was made to employ ReactJS, a highly regarded front-end JavaScript (Wirfs-Brock & Eich, 2020) library that was thoughtfully developed by the innovative minds at Facebook.(Rawat & Mahajan, 2020) This library's open-source nature not only encourages

I. Kazanidis, A. Tsinakos (Eds.), Proceedings of the 13th Pan-Hellenic and International Conference "ICT in Education", International Hellenic University, Kavala. Greece, 29 September – 1 October 2023. ISSN: 2529-0916, ISBN: 978-618-83186-8-7

collaboration but also ensures a robust and versatile foundation for crafting a visually appealing and user-friendly interface.

Turning our attention to the back-end, Django (Gore et al., 2021), an open-source web framework that thrives within the Python (K. R., 2017) programming ecosystem, was selected. This choice is grounded in the framework's remarkable ability to streamline and expedite the development process. By harnessing Python's elegance and versatility, Django empowers developers to create a seamless and efficient back-end structure, capable of handling intricate functionalities with grace.

In the realm of data management, PostgreSQL (Stonebraker & Rowe, 1986), a high-performance open-source relational database management system (RDBMS), was deemed the most fitting choice. Renowned for its reliability and robustness, PostgreSQL offers a secure and scalable foundation for storing and managing data. The open-source nature of PostgreSQL not only aligns with the ethos of collaborative development but also ensures the longevity and flexibility required to accommodate future expansions and optimizations.

In essence, the amalgamation of ReactJS, Django, and PostgreSQL symbolizes a strategic fusion of open-source tools, each meticulously chosen to cater to different aspects of the application's architecture. This harmonious blend not only exemplifies the prowess of community-driven development but also sets the stage for an application that is not only functional but also adaptable and ready to embrace the ever-evolving landscape of technology.

The database schema (Fig. 1) was designed in the web app DrawSQL. It's composed of 9 tables: user, patients, tumor_type, tissue_type, temperature, containers, sample, cut and analysis. The user table is the default table built in the Django authenticatin system.

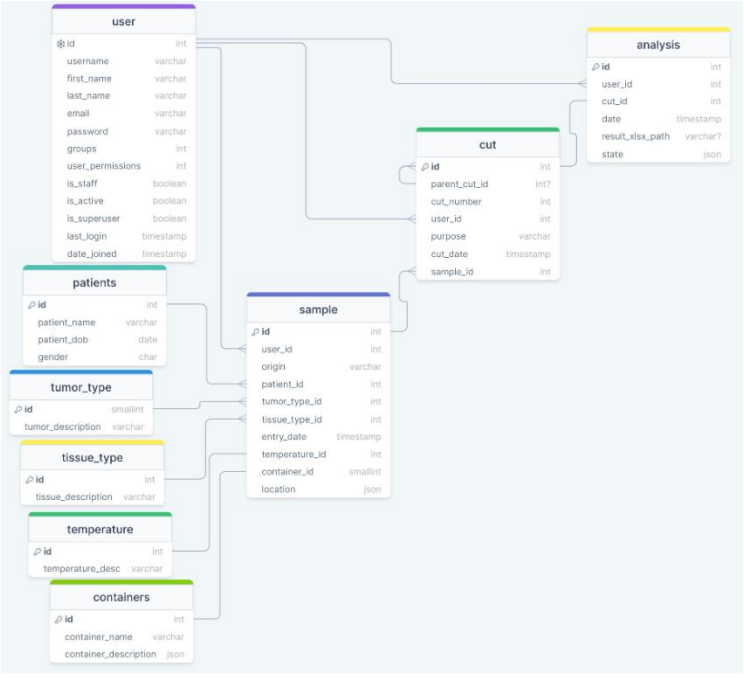

Figure 1. Database schema for coley

Discussion

The use of ReactJS, Django and PostreSQL allowed for the development of a functional application that can be implemented and used in a immunotherapt laboratory setting. One of the core and fundamental functionalities and features of this system is its inherent ability and capacity to effectively and efficiently manage and handle the registration, enrollment, and inclusion of a wide array and assortment of users, each possessing and holding distinct, varying, and unique roles, responsibilities, and authorizations within and throughout the entire framework and structure of the system itself, while capturing and documenting the chain of custody of the material. This dynamic and flexible user management component ensures that the system is capable of accommodating and adapting to the intricate and complex organizational and hierarchical structure that is often characteristic and synonymous with laboratory environments, particularly if these processes may be used to develop more structured supervised and quality-controlled steps which feed into the development of GMP-grade products. The ability of uploading result files for a given sample is a key feature for the laboratory workflow, that migrates from the use of a high number, often difficult-to-manage excel sheets to a single, centralized system that manages, registers and allows the access to the results in a single place. This is a substantial improvement on the laboratory workflow, since it facilitates the users job and its supervision, dimishes registration associated errors and eases the access to results. The first version of this application can be greatly improved. Features such as study creation, sample and cuts visualization, patient/study associated results, data visualization directly in the application, app user monitoring and other can be of great value for a laboratory dedicated to preclinical development. These features are included in many modern LIMS and would represent a valuable enhancement of our current system version. A safer and more comprehensive documentation will also allow for a improved depth and richness in data analysis since clinical samples often undergo so-called exploratory assays in addition to highly quality-controlled and validated assays (performed in standard clinical laboratories). The increased data quality using the current platform is – in part – an answer to the newly promulgated. In Vitro Diagnostic Regulation (IVDR) legislation, which replaces older legislation and sets a higher bar for data quality and safety for in vitro diagnostic medical devices (IVD) in the European Union (EU). The data are classified based on a risk-based system (A-D) guided by the purpose of the data acquisition, their potential harm for the patient, assay performance and evaluation of the validity test platforms. (ref: MDCG 2022-6 - Guidance on significant changes regarding the transitional provision under Article 110(3) of the IVDR (europa.eu). This cannot be achieved with the current version of the laboratory documentation system presented here, yet it presents an improvement of the often utilized excel platform in preclinical development towards a safer, use-friendly sample analysis/documentation system.

Conclusion

We developed a LIMS prototype, using ReactJS, Django and PostgreSQL to respond to the basic needs of an immunotherapy laboratory, replacing the standard Excel data management approach. Functionalities such as data evaluation, assay creation, user performance control,

chain of custody and risk-based data classification according to the IVDR updated guidelines, could be added in future versions.

References

Gore, H., Singh, R. K., Singh, A., Singh, A. P., Shabaz, M., Singh, B. K., & Jagota, V. (2021). Django: Web development simple & fast. *Annals of the Romanian Society for Cell Biology*, *25*(6), 4576–4585.

K. R., S. (2017). Python -The Fastest Growing Programming Language. *International Research Journal of Engineering and Technology*, *4*(12), 354–357.

Rawat, P., & Mahajan, A. N. (2020). ReactJS: A Modern Web Development Framework. *International Journal of Innovative Science and Research Technology*, *5*(11), 698–702.

Stonebraker, M., & Rowe, L. A. (1986). The Design of POSTGRES. *ACM SIGMOD Record*, *15*(2), 340–355. https://doi.org/10.1145/16856.16888.

Wirfs-Brock, A., & Eich, B. (2020). JavaScript: The first 20 years. *Proceedings of the ACM on Programming Languages*, *4*(HOPL). https://doi.org/10.1145/3386327.