Learning Management Systems as a Platform for Deployment of Remote and Virtual Laboratory Environments

Anastasiia Sapeha[1], Aleksandra Zlatkova[2], Marija Poposka[2], Filip Donchevski[2], Kirill Karpov[1], Zdravko Todorov[2], Danijela Efnusheva[2], Zhivko Kokolanski[2], Andrej Sarjas[3], Dusan Gleich[3], Marija Kalendar[2] and Eduard Siemens[1]

[1]*Department of Electrical, Mechanical and Industrial Engineering, Anhalt University of Applied Sciences 55 Bernburger Str., Köthen, Germany*

[2]*Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University 18 Rugjer Boshkovik Str., Skopje, R. N. Macedonia*

[3]*Faculty of Electrical Engineering and Computer Science, University of Maribor, 15 Smolskov trg., Maribor, Slovenia*
{anastasiia.sapeha, kirill.karpov}@hs-anhalt.de, {aleksandraz, fdoncevski, poposkam, todorovz,danijela,kokolanski, marijaka}@feit.ukim.edu.mk, eduard.siemens@hs-anhalt.de, { andrej.sarjas, dusan.gleich}@um.si

Keywords: LMS, Moodle, Open edX, WebLab-Deusto, RDP, EjS/EjsS, Distance Learning, Remote Laboratory.

Abstract: The constant development in ICT has not omitted the field of learning activities, digitalizing the concept of learning and making it more effective. This became even more obvious during the Covid-19 Pandemic when all educational activities were forced to be transferred remotely and online, with the help of existing Learning Management Systems. Motivated by this, the paper focuses on reviewing different LMSs and comparing their capabilities towards implementation of remote and virtual laboratories, providing analytical and empirical investigations. The goal is to assess the possibility of implementation and deployment of remote access, firstly to some existing on-campus laboratories hosting lab exercises based on real hardware; and secondly to virtual software platforms also usually available in the laboratories. Using such a set up enables the students to get the look and feel of what it means to be in presence in a real lab, nevertheless virtually. This immersive remote lab experience shall hereby be integrated into an already existing and widely used LMS for better operability and manageability. This work has been done in the course of the European project UbiLab ("A ubiquitous virtual laboratory framework"), conducted jointly between the Ss. Cyrill and Methodius University in Skopje, North Macedonia, Anhalt University, Koethen, Germany, and University of Maribor, Slovenia. In the course of the project a remote collaboration platform for hardware-in-the loop will be developed and deployed, and at the same time learning experience in using it in the course of the education process of electrical, system automation, and computer engineers shall be gathered. As an outcome of the work done, it was found that there is no ready-made solution for carrying out remote laboratory work that completely covers the goals of the UbiLab project. Thus, a custom solution based on the compilation of listed platforms components should be developed.

1 INTRODUCTION

The previous couple of years nave been very challenging in many aspects for the modern world. They have brought changes in the usual functioning of many areas of business, life, and education as well. Due to these challenges, the presence of students at the university campuses, and most of all, in the university laboratories has been significantly limited. We can argue that the most affected aspect of students' study experience is the limited access to the laboratory experiments that used to be conducted on site, in the lab facilities of the universities. Even though, the research of virtual and remote laboratories is not something that is emerging at the moment, today's challenges brought by the Covid-19 Pandemic, have pushed forward the need to intensively explore down this road and provide improved virtual and remote laboratories and implement novel ideas in this area. The main goal of the "A ubiquitous virtual laboratory framework - UbiLAB" project, supported by the Erasmus+ framework in the special "Coronavirus response: Extraordinary Erasmus+ calls to support digital education readiness and creative skills", [1], is exactly aiming in this direction. The UbiLAB project aims to design a complete Virtual Laboratory Framework enabling the process of designing and implementing more realistic remote and virtual laboratory exercises, and at the same time supporting the entire process of university study, as well as the collaborative learning experience of students. Thus, taking into account the learning experience that educators used to provide to their students, we have identified two most difficult challenges: a nearly realistic substitute for learning experiences in the actual physical laboratories; and a nearly realistic substitute for the social element of collaborative learning and "making friends" in the process [2, 3]. These challenges are the main focus of the UbiLAB project.

On the other hand, today it is almost impossible to imagine any kind of training without the support of LMS (Learning Management Systems) platforms, especially when everything takes place at a distance. Two of the most popular and widely used LMS platforms, that we decided to focus on, are Moodle and Open edX [3, 4, 5, 6]. These LMS platforms, despite the fact of being most widely used on one hand, are open-source on the other. Additionally, due to our focus on remote and virtual laboratories and the need to provide remote access to actual hardware, we add to the review the specialized RLMS (Remote Laboratory Management System) Weblab - Deusto [7]. Reviewing these platforms defines their strengths and weaknesses, and subsequently will enable building upon those previous experiences to produce a novel, strengthened and a more suited UbiLAB platform. This platform will be addressing current modern challenges reinforced by the ongoing Covid-19 pandemic, but will nevertheless be practical and usable in general for the currently developing era of digital living, work and education. The review of the platforms will focus on the elements required by the students and teachers, always having in mind the almost realistic, but nevertheless virtual, experiences in the educational, practical and social aspects.

The rest of the paper is organized as follows: The second section describes each reviewed platform from an architectural and feature viewpoint. The third section is focused on the possibilities to implement laboratory environments and the special tools and features for virtual and remote laboratory experiments. The fourth section compares all the reviewed platforms, especially focusing towards the features expected from the UbiLAB platform. Finally in the last section we summarize and conclude the paper.

2 ARCHITECTURE AND FEATURES OF THE REVIEWED PLATFORMS

Even though many LMS platforms exist and are widely used, this section will focus on the three chosen (R)LMS systems. The first two (Moodle and Open edX) are complete full featured LMS platforms and both have an active developer community that constantly improves the platform by adding new elements and modules (tasks of specific formats, new analytic tools, and similar). The third platform, Weblab-Deusto, is a specialized Remote Laboratory Management System focused on providing the elements for developing and implementing remote and virtual laboratories, but also enabling some of the features of a usual LMS system. This section will focus on all three platforms, explaining their architecture, main features, and their extensibility.

2.1 Moodle LMS

Moodle (Modular Object-Oriented Dynamic Learning Environment) is an open-source LMS mainly intended to improve online and remote learning[8]. The first version of Moodle was introduced in 2002 to provide students and teachers with the needed technology for interactive and cooperative distance learning. Moodle has evolved to offer access to a great variety of learning materials, opportunity to connect with peers, various tools to support online learning activities (discussion forums, chats, assignments, quizzes, grading books and a plethora of interesting community contributed add-ons). As such, Moodle contributes to improving the effectiveness of online learning [9]. As seen in the following sections, Moodle is not natively intended to support virtual and remote laboratories.

2.1.1 Moodle LMS Architecture

Moodle has been developed based on the very popular open-source LAMP architecture (Linux, Apache, MySQL, PHP), presented in figure 1. Nevertheless, a wide range of operating systems, database systems, web servers, and programming languages can be supported due to Moodle's open-source and modular nature.

As Figure 1 presents, the architecture of Moodle utilizes the most common elements of today's web applications: a request made by the user; consecutively passed to the web-server that calls the PHP module responsible for the call; the PHP module calls the database with an action that returns the requested data in the form of HTML code to the web-server. Finally, the information is displayed back to the user.

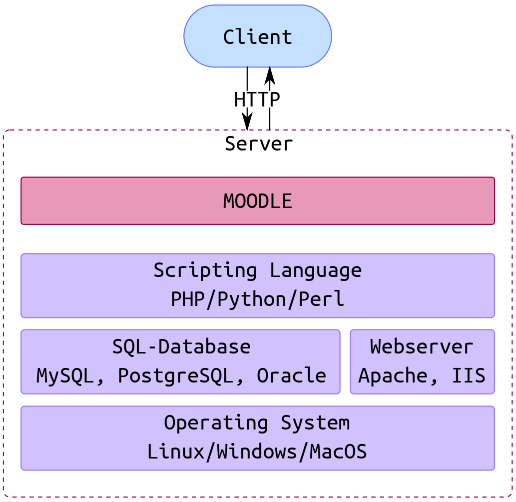


Figure 1: Simplified architecture of the Moodle platform.

2.1.2 Features of Moodle LMS

This section presents a selection of Moodle's features. Some of them are especially interesting for the possibilities to design virtual and remote laboratories; keeping track of students' progress; and the collaborative study experience:

- **Open source:** Moodle is full-feature open source project, so the source code can be downloaded and tailored to the needs.
- **Security and privacy:** A very important feature enabling complete control over the stored and used data.
- **Flexible learning:** Moodle provides its own building education platform that supports deep collaborative learning, through group activities, exploration and experimentation, Calendar, Notifications and Messaging.
- **Mobile learning:** Users can access all content, activities and assignments through a Moodle App, in order to improve the user experience on modern touch screen devices.
- **Scheduling:** Moodle provides extensive ability to set up timed schedules for resources and modules, as well as creating deadlines for tasks.
- **Course access control:** The administrator staff can configure the number of students who have access to the material, and can provide access to resources to defined groups and/or students.
- **User tracking:** Moodle has elaborate resources for tracking student activities in the course and in the separate activities, to enable different analytic and grading. Moodle has a quite elaborate grading system implemented.
- **Administration:** The Moodle platform uses modular and detailed administration tools on many levels: site, course, user administration.
- **Easy integration and extensibility:** Moodle can be integrated with various other programs and platforms to suit different needs. Moodle is a highly modular system. Thus besides the core components, Moodle comprises a wide variety of modules and plugins. Some are incorporated in the Moodle plugin database. However, many community-contributed plugins exist that enable extending Moodle in versatile ways. Moodle also supports the most popular standards in e-learning: IMS, AICC, and SCORM.

All the features present Moodle as a full-featured LMS platform with all the needed elements for course creation and modeling. On the other hand, as seen from the usual features, Moodle lacks the elements for building and connecting virtual and remote laboratories. Nevertheless, Moodle's extensibility and modularity make it an obvious candidate for an easily extensible platform that can be modeled to incorporate needed features, even though they have not been implemented yet.

2.2 Open edX LMS

Open edX is an open-source educational platform that allows organizing online learning for various educational tasks: an online campus, instructor courses, group training programs and one-off training courses.

Open edX includes two main components - Studio for developing course content and LMS. Studio is a tool that manages xBlocks, individual pieces of content from which courses can be composed in the desired sequence. Each xBlock is a Sharable Content Object Reference Model (SCORM) compliant, so material in Open edX can be easily imported from different compatible systems (Camtasia, Articulate Storyline, Adobe Captivate), giving users more flexibility in composing and organizing content [6].

2.2.1 Open edX LMS Architecture

The Open edX platform consists of LMS components with which students interact directly, and Studio designed components used for learning management and course authoring.

Figure 2 shows a simplified architecture of the Open edX platform and its main components [10].

The LMS uses multiple data stores. For example, MongoDB is used to store courses, MySQL stores student data, and videos are hosted on Amazon S3. Studio is the course development environment which enables creating and updating educational material. Studio writes its courses to MongoDB. The server-side code in Open edX is written in Python. Django is used as a framework for web applications.

On the other hand, Open edX courses are made up of modules called XBlocks. Open edX provides the ability to write own XBlocks, which gives the instructors great flexibility in presenting course information and extending the platform.

However, there are several other ways to expand the functionality of the course. For example, course authors can embed Learning Tools Interoperability (LTI) components to integrate third-party learning tools into an Open edX course. JavaScript components can be integrated using JS Input. Furthermore, because the Open edX platform is built primarily using Python, the authors can use built-in Python code to present a problem or evaluate a student's response. The Python code written by the instructor is executed in the secure CodeJail environment. Elasticsearch was also successfully integrated into Open edX and provides a more flexible search for information on the platform.

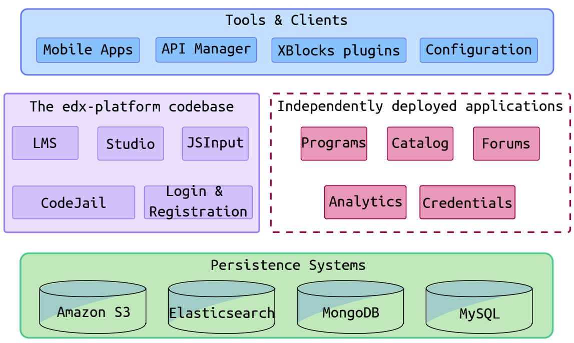


Figure 2: Simplified architecture of the Open edX platform.

2.2.2 Features of Open edX LMS

Open edX [11], like any other LMS, provides some standard functionality, such as tracking student progress, a platform for group discussion or sharing materials, as well as the availability of various types of quizzes to check and evaluate lecture material. The following platform features can be noted:

- **Open source and free.** Similar to Moodle, the source code can be downloaded and freely modified.
- **Scheduling:** Open edX enables setting up release schedules for certain Units, and creating deadlines for completing tasks.
- **Course access control:** The number of students who have access to the material can be limited, and access to lectures can be provided to specific groups, or even students.
- **User tracking:** Open edX also has extensive user tracking capabilities on course level and study modules level as well.
- **Administration:** The Open edX platform includes a specialized administration panel and module for course and user administration.
- **Intuitive interface:** The interface allows students to take classes easily, and tutors to create a course without even looking at the manual.
- **Certificates:** According to the results of the course, the student can receive a certificate. It can be generated from the default template of the Open edX platform, but lecturers have the ability to modify and upload their own templates as well.
- **Integration with Google:** Users can add content from both the Google Calendars and Google Drive apps to the course. Students can view the class schedule and files uploaded by the tutor.
- **Flexible search:** Elasticsearch allows users to search by keywords not only in the list of courses or teachers, but also in the comments database.
- **Extensible:** The Open edX platform is made up of XBlocks, that can be modified as needed, or written by course developers.

Considering the building components of the Open edX platform and the extension possibilities, it can be concluded that Open edX is a full-featured LMS system used typically as a self learning course environment with many possible extensions, but mostly in the virtual software (laboratory) learning domain.

2.3 WebLab-Deusto - A Specialized RLMS

WebLab-Deusto is an open-source RLMS used to develop and manage remote (physical) and virtual laboratories [7]. This platform was developed at the University of Deusto in the early 2000s. At first, it was used mainly in classes for CPLDs and FPGAs, but its acceptance contributed to the development and implementation of various other laboratories [12]. The usefulness and usability of remote laboratories was researched, and the results show that the platform is functional and useful, and accordingly, students get a good feeling of control over the hardware/equipment [13, 14].

2.3.1 Architecture of Weblab-Deusto RLMS

The remote laboratories developed in Weblab-Deusto can be managed or unmanaged laboratories [13]. The architecture of Weblab-Deusto is presented in Figure 3.

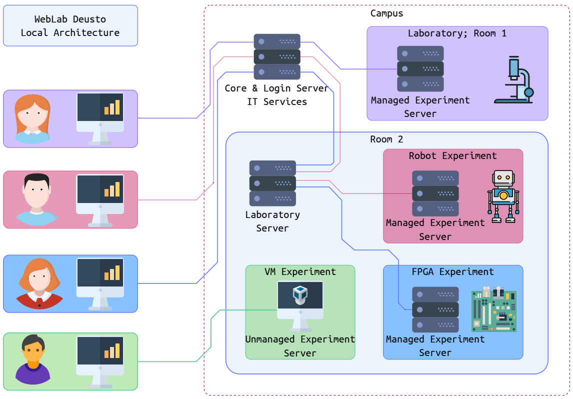


Figure 3: Simplified architecture of the Weblab-Deusto platform.

The managed laboratories are developed using the application programming interface (API) of Weblab-Deusto. This enables bypassing the development of communications, security, and information stored in databases regarding the experiment server because it is being handled through the API of the platform. Weblab-Deusto supports and provides API libraries for multiple programming languages: Python, Node.js, Java, .NET, C++ and C. Managed laboratories consist of two parts: a client and a server; the developer does not manage the communication; and the data is stored in a database that can be used for user tracking.

On the other hand, unmanaged laboratories are those where the communication does not go through WebLab-Deusto. An unmanaged laboratory handles the experiment server communications directly, running a custom server-client app without regard to Weblab-Deusto API. This enables the flexibility of using any programming language, framework, or protocols not supported by Weblab-Deusto (WebSockets, virtual machines, SSH/VNC/Remote Desktop). However, Weblab-Deusto is responsible for the authentication, authorization, scheduling and user tracking in the time slot during the establishment of the communication between the experiment and client. The client (student) using HTTP with JSON connects to the main server responsible for the authentication, authorization, scheduling, user tracking; then he chooses the laboratory, and the main server sends the request to the laboratory. If the authorization is verified, the student gets the privilege to access the laboratory. When another user uses the laboratory, the student will wait in a queue.

2.3.2 Features of Weblab-Deusto RLMS

Weblab-Deusto is a specialized platform for remote laboratories (virtual and physical). The benefits of this platform can be observed through the following useful features [13]:

- **Open Source:** Weblab-Deusto is also an open source platform, and it requires exclusively open source technologies. It also supports proprietary technologies for optional extension.
- **Extensible:** WebLab-Deusto is designed to ease the development of remote laboratories. It provides APIs and different approaches to include new or existing remote labs.
- **Administration:** WebLab-Deusto provides extensive web administration tools.
- **LMS integration:** WebLab-Deusto laboratories can be accessed from LMSs, for example Moodle, relying on the LMS for authentication and authorization.
- **LightWeight:** Weblab-Deusto can run at low-cost devices, but it must be supported by the requirements for the laboratory.
- **Federation:** Weblab-Deusto enables the educational institution to share the designed laboratory with other educational institutions.
- **Scheduling:** WebLab-Deusto guarantees exclusive access to the laboratories through a priority queue subsystem. It can be customized to support multiple concurrent (optionally collaborative) users to a single laboratory.

Considering the features, WebLab-Deusto is an open-source, extensible RLMS platform incorporating some of the usual LMS platforms features (scheduling, user tracking- in the course of using the laboratory), but mainly specialized in the design of remote and virtual laboratories.

3 ADD-ONS AND CAPABILITIES FOR DEPLOYMENT OF REMOTE AND VIRTUAL LABORATORIES

This section willl present the add-ons (modules) enabling development and implementation of remote and virtual laboratories by extending the functionalities of the usual full-featured LMS platforms lacking native support for such laboratories. Additionally, the native capabilities of WebLab-Deusto for building the remote and virtual laboratories will also emphasized. Finally, the section will present some example remote and virtual laboratories developed in each platform.

3.1 Easy Java/JavaScript Simulation in LMS

Easy Java/JavaScript Simulation (EjS/EjsS) is also a free and open-source standalone tool (not an LMS) used for creating virtual and remote laboratories[15]. This tool is being independently developed for more than a decade, and has been extensively used to create a great number of simulations and remote labs, mostly in the Physics domain (ComPADRE-OSP digital library [16]), as well as many virtual and remote labs in the automatic control field (UNILabs network [17, 18]). Eventhough most of the previous applications were based on Java and deployed as Java applets, EjsS now focuses on building Javascript simulations.

An example of a simple Physics experiment with mass and spring is shown in Figure 4.

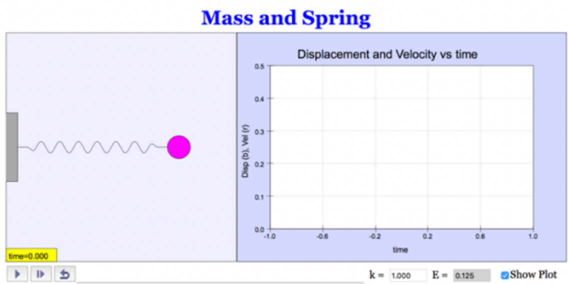


Figure 4: EjS Mass and Spring Simulation Example.

The EjsS is a tool that provides a full platform of elements and libraries to design and implement simulations conveying the laws of Physics in many different fields of study. The full set of already available features makes it simple, not only for scientists with computer skills, to build such realistic simulations comprising many predesigned graphic components specialized for a given type of visualization or user interaction, in a sophisticated HTML interface. Through these front-end elements, the user has full control over the simulation (reset to initial state, change the variables through input fields and/or slider elements, run, pause, reset or run the simulation step by step). The EjS/EjsS platform also enables elements for extending the simulations and accessing remote physical laboratories and communicating with real equipment [19, 20].

3.1.1 Moodle Extension for EjS/EjsS Simulations

Since the EjS/EjsS platform only provides the modules for designing simulations, virtual and remote laboratories, in order to enable a full set of classroom/laboratory experience as intended in the UbiLab Project, an additional LMS, such as Moodle, would be required. Since Moodle is the most widely used open-source LMS, and highly extensible by developing plugins or add-ons, the EjS/EjsS platform has utilized this possibility and has developed a specialized and fully featured Moodle plugin called EJSApp[21].

This plugin enables all EjS/EjsS applications to be embedded into the Moodle LMS. By embedding the applications in the Moodle LMS, they benefit from the integration with the usual and additional Moodle LMS features: integrating a booking system for controlling the access to the RL, multi-language support, saving data and image files from the virtual or RL application to users' file repository in the LMS, grading, monitoring the time spent by users working with the experiment, as well as backup and restore options.

Thus, preparing an EjS/EjsS application (simulation or remote lab) beforehand, and then implementing it into Moodle LMS using the EJSApp plugin [22] is a very good option that enables many possibilities for virtual and physical remote laboratory experiments, while accompanying them with the usual students' tracking and learning options provided by the LMS platform.

Figure 5 presents the architecture and features overview of the EJSApp plugin incorporated within Moodle LMS.

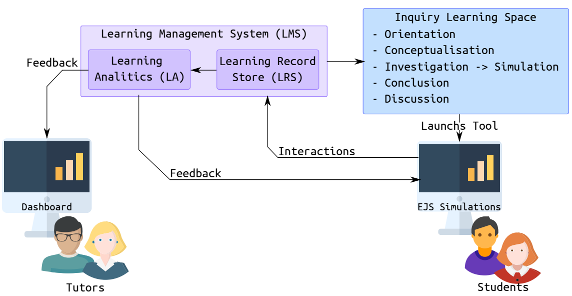
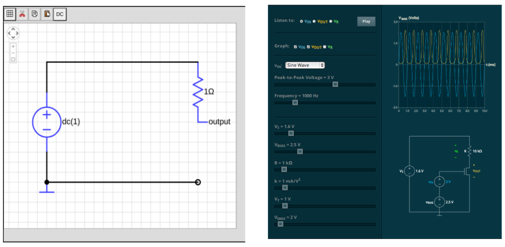


Figure 5: EJSApp Moodle LMS plugin and its features.

3.2 Open edX Laboratory Extensions

As already noted, the Open edX LMS is built upon the XBlocks framework that enables designing interactive tasks for students [10]. The XBlocks



(a) (b)

Figure 6: Examples of Using XBlocks to Create Labs. (a) The Circuit Schematic Builder, (b) The Conditional XBlock.

framework enables creating not only quizzes, but more complex tasks as well. XBlocks components support compilers for various programming languages. Some examples of using XBlocks in the labs are presented in (Figure 6).

The Circuit Schematic Builder (Figure 6(a)) allows students to create virtual circuits by placing elements such as voltage sources, capacitors, resistors, and transistors on an interactive grid. The system evaluates a permanent variable or enables transient circuit analysis. Flowchart-based Conditional XBlocks (Figure 6(b)) allows students to change the input while watching the output change.

One of the most customizable blocks is the iFrame. This comprises HTML code used to embed interactive media, as well as third-party pages/code into a web-page. iFrame creates a separate window in the html document, which is located inside a regular document, and it allows the user to load other independent documents, videos and interactive media files into the page in an area of given size.

These functionalities of the XBlocks framework present a quite flexible possibility for implementing different types of virtual and remote laboratories and presenting various and intuitive user interfaces.

3.3 Weblab-Deusto Remote Laboratories

The WebLab-Deusto platform, as presented before, is responsible for the educational, organizational and technological objectives [13] of building remote laboratories. It provides defined API modules to ease the process of laboratory development. Consequently, WebLab-Deusto does not require additional extensions for enabling remote and vrtual laboratory development. In addition, there are many ready-made laboratories implemented on this platform[13], [23] that prove the flexibility and modularity of Weblab-Deusto for building additional and diverse laboratories. For example, one of the most popular implemented physical experiments in the laboratory is Arduino Robot (Figure 7) where students can practice their robot programming skills remotely, but at the same time see the results through a live streaming camera. The task is to program the robot to follow a line or avoid walls.

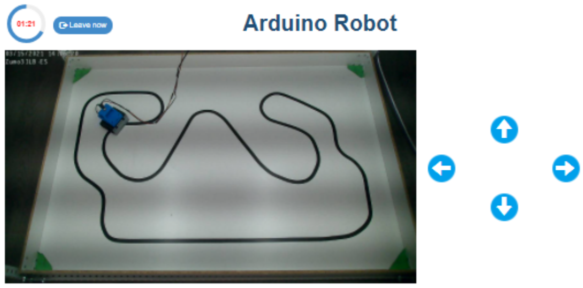


Figure 7: Line Follower Arduino robot.

The development of a new laboratory in WebLab-Deusto is mostly left to the creativity of the developers, since there is no specific way of developing a WebLab-Deusto laboratory. The process consists of developing a client-side web based application using common technologies (JavaScript, CSS and HTML). Next, the communication between the client and the laboratory needs to be implemented by utilizing the already available Weblab-Deusto API functions. In order to enable interaction between the student and the laboratory hardware, it is also necessary to implement a server side software using the existing callback functions from the Weblab-Deusto API.

3.4 Apache Guacamole as an LMS Extension

Apache Guacamole is a client-side remote desktop gateway that supports standard protocols such as Virtual Network Computing (VNC) protocol, Remote Desktop Protocol (RDP), and Secure Shell (SSH) protocol. It accesses remote desktops through a web browser, which means no plugins or additional software is required. It should be noted that this tool can also be used to extend the capabilities of other platforms. The architecture of Apache Guacamole is presented in Figure 8.

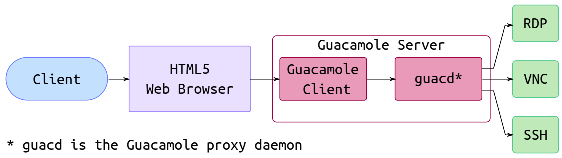


Figure 8: The architecture of Apache Guacamole.

Apache Guacamole supports LDAP database integration, which provides students with a smoother connection to remote machines. This tool is especially useful when using an iFrame in an LMS, and as such it can be noted as a universal extension for LMS platforms such as Moodle and Open edX when remote laboratory work needs to be carried out. Figure 9 shows a diagram of a remote desktop connection using Open edX LMS, an iFrame tool, Apache Guacamole, and a student account (StA) in LDAP. Figure 10 shows an example of using Apache Guacamole to perform remote lab work in a Matlab application.

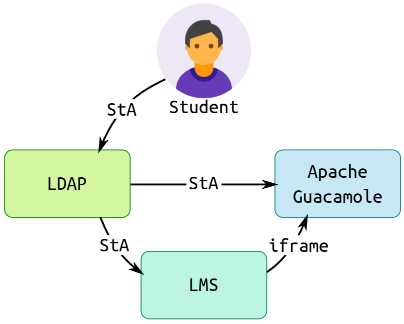


Figure 9: Learning system authentication blocks.

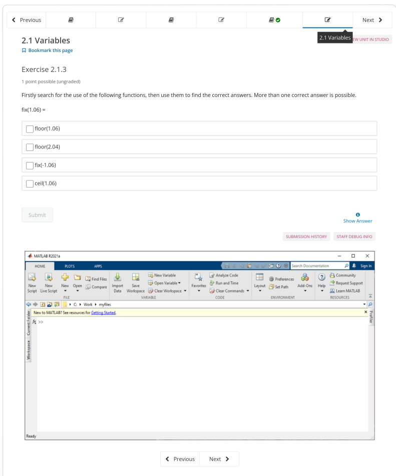


Figure 10: Open edX assignment with remote desktop.

4 PLATFORMS REVIEW REGARDING VIRTUAL AND REMOTE LABORATORY EXTENSIONS

This section presents the comparison of the different reviewed platforms features regarding the possibilities for implementing virtual and remote physical laboratories, as well as the possibilities for extending the platforms with additional own collaborative learning social elements. Table 1 presents a summary of the evaluation of the platforms' features.

Table 1: General comparison.

	Moodle	Open edX	WLDeusto
Virtual Simulations	Yes	Yes, through XBlocks	Limited
Real Time Simulations	Yes	Yes, through XBlocks	Yes
Scheduling System	Yes with plugin	Yes	Yes
User tracking	Limited	Yes	Yes
Grading System	Yes	Yes	No
Programming Language	PHP/JS	Python	Python
No. of concurrent laboratory users	Limited by lab type	Limited by lab type	30
Video call integration	Limited	Yes, through XBlocks	Limited
Mobile learning	Yes	Yes	Yes
Integration System	Moodle dependent	Stand-alone	Stand-alone
No. of concurrent platform users	Limited by hardware	Limited by hardware	Limited by hardware
Remote access to desktop device	N/A	Limited by lab type	Limited
Laboratory sharing across universities	Yes	Yes	Yes

As presented, the platforms are being evaluated according to the following features as they represent the most important elements that would provide insight for the future building of the unified UbiLAB virtual and remote laboratories platform. Each feature is elaborated according to the importance for building the virtual and remote laboratories.

1) **Virtual Simulations** The user can run and observe virtual (software) simulations.
2) **Real-time Simulations** The user can run and observe real-time (physical) simulations.
3) **Scheduling System** The user can check if a laboratory is available and schedule the laboratory.
4) **User tracking** The platform tracks the user progress with the laboratory exercise.
5) **Grading System** Incorporation of student grading in the system.
6) **Programming Language** The programming language used for the development of simulations.
7) **No. of concurrent laboratory users** The maximum number of students (incl. the professor) in a laboratory.
8) **Video call integration** Incorporation of video call system during laboratory exercises within the platform.
9) **Mobile learning** Support for learning from mobile devices (tablets and smartphones)
10) **Integration System** The system on which the LMS platform runs.

11) **No. of concurrent platform users** The maximum number of concurrent users using the LMS.
12) **Remote access to desktop device** The lowest system level of access of the user to the platform.
13) **Laboratory sharing across universities** Sharing laboratory resources between universities.

The features: **ease of new simulation development** and **ease of new simulation implementation**, quantify the difficulty level for a new user to develop or implement a new simulation. For all reviewed platforms we concluded that for an average new user, the difficulty level for development is medium, and for implementation is easy.

According to the features and comparison of the reviewed platforms we can argue that all of the platforms mainly possess the basic features for enabling a complete full-featured LMS system focused towards development and implementation of virtual and remote laboratories and collaborative learning experience. Nevertheless, each platform presents some unique features making it stronger on one side or the other.

The WebLab-Deusto platform is primarily intended for the development of laboratories with flexibility in used programming languages and the design of the experiment. The tools provided (API interfaces for communication, administration, scheduling, reservation) are mainly aiding the process of laboratory development. The features from a full-featured LMS system are generally lacking.

On the other hand, Open edX has an advantage for self-paced learning and for offering flexible extension capabilities by adding or developing specialized XBlocks and/or iFrame modules. On the downside, the Open edX standard course creation toolkit does not natively include virtual and remote lab development tools.

Finally, Moodle LMS, as a long existing and constantly developing LMS, even though does not provide native tools for laboratory development, has the biggest number and greatest modularity of additionally developed tools, effortlessly integrated in the platform. Due to these extension capabilities, Moodle can be easily upgraded with existing and novel laboratory development tools, as well as innovative collaborative learning modules.

5 CONCLUSIONS

Different LMSs differ not only in functionality, but also in what problems they can solve. Therefore, there is no universal solution to the LMS market and to the goals of the UbiLAB Framework, as well. By choosing to focus on open-source platforms, we have intentionally sought the flexibility of upgrading the platforms (improving the elements and modules that lack or do not have the full features required by the envisioned UbiLAB framework).

Thus, all platforms are good candidates for being the core of the UbiLAB Framework, since they present strengths in different elements. WebLab-Deusto is scored on the top for having full set of ready-made elements for developing new and versatile virtual and physical laboratories. Moodle is put on the top for including most elements for student interaction and success tracking, student materials, scheduling tasks, and student collaboration. Open edX is top-scored for its self-paced learning process and great possibilities for designing modular, versatile, and interactive tasks for the students.

Due to these conclusions, we can propose basing our UbiLAB framework on either Moodle or Open edX platform (as a solid basis for a complete LMS system) and upgrading and developing the additional modules envisioned in the UbiLAB framework. On the other hand, since WebLab-Deusto is the best choice for developing virtual and remote laboratories - an important part of the UbiLAB framework, and because WebLab-Deusto already has a Moodle integration developed, we would focus in this direction, too. Nevertheless, regarding the virtual software environments and interaction with the students, Open edX seems superior. Thus we would also consider integrating its advantages into the final full-featured UbiLAB framework, mostly for enabling the envisioned virtual software laboratories.

6 ACKNOWLEDGMENTS

This work has been supported by the European Erasmus+ project ”A ubiquitous virtual laboratory framework” (UbiLAB, ERASMUS+ Key Action 2, ref: 2020-1-MK01-KA226-HE-094548).

REFERENCES

[1] Coronavirus response: Extraordinary erasmus+ calls to support digital education readiness and creative skills. [Online]. Available: https://erasmus-plus.ec.europa.eu/news/coronavirus-response-extraordinary-erasmus-calls-to-support-digital-education-readiness-and-creative-skills-0.

[2] I. Gustavsson et al., “On objectives of instructional laboratories, individual assessment, and use of

collaborative remote laboratories," IEEE Transactions on Learning Technologies, vol. 2, no. 4, pp. 263–274, 2009.

[3] D. Galan et al., "Automated assessment of computer programming practices: The 8-years uned experience," IEEE Access, vol. 7, pp. 130 113–130 119, 2019.

[4] F. J. G. Clemente et al., "Development of learning analytics moodle extension for easy javascript simulation (ejss) virtual laboratories," 2019.

[5] N. Cavus, "Distance learning and learning management systems," Procedia - Social and Behavioral Sciences, vol. 191, pp. 872–877, 2015, the Proceedings of 6th World Conference on educational Sciences. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877042815028712.

[6] Y. Chaabi et al., "Development of a learning analytics extension in open edx," in 2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI), 2021, pp. 1–6.

[7] P. Orduña et al., "The weblab-deusto remote laboratory management system architecture: Achieving scalability, interoperability, and federation of remote experimentation," in Cyber-Physical Laboratories in Engineering and Science Education, M. Auer et al., Eds. Springer, Cham, 2018, ch. 2, pp. 17–42. [Online]. Available: https://doi.org/10.1007/978-3-319-76935-6 2.

[8] N. Simanullang et al., "Learning management system (lms) based on moodle to improve students learning activity," Journal of Physics: Conference Series, vol. 1462, p. 012067, 02 2020.

[9] N. H. S. Simanullang et al., "Learning management system (LMS) based on moodle to improve students learning activity," Journal of Physics: Conference Series, vol. 1462, no. 1, p. 012067, feb 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1462/1/012067.

[10] "Open edX Developer's Guide," Circuit Sub PBLLC, Developer's Guide, January 2022. [Online]. Available: https://openedx.org/.

[11] "Open edX Learner's Guide," Circuit Sub PBLLC, Tech. Rep., February 2022. [Online]. Available: https://openedx.org/.

[12] J. Garcia-Zubia et al., "Acceptance, usability and usefulness of weblab-deusto from students point of view," in 2008 Third International Conference on Digital Information Management, 2008, pp. 899–904.

[13] J. Garcia-Zubia et al., "Addressing software impact in the design of remote labs," IEEE Transactions on Industrial Electronics, vol. 56, p. 4757 – 4767, 12 2009.

[14] J. Garcia-Zubia et al., "Application and user perceptions of using the weblab-deusto-pld in technical education," in 2011 First Global Online Laboratory Consortium Remote Laboratories Workshop, 2011, pp. 1–6.

[15] Esquembre et al., "Easy java simulations: A software tool to create scientific simulations in java," Computer Physics Communications, vol. 156, pp. 199–204, 01 2004.

[16] F. J. García Clemente, "Ejss: A javascript library which makes computational-physics education simpler," 08 2014.

[17] J. Sáenz et al., "Open and low-cost virtual and remote labs on control engineering," IEEE Access, vol. 3, pp. 805–814, 2015.

[18] D. Chaos et al., "Virtual and remote robotic laboratory using ejs, matlab and labview," Sensors (Basel, Switzerland), vol. 13, pp. 2595–612, 02 2013.

[19] R. Pastor Vargas, "Web-based virtual lab and remote experimentation using easy java simulations," vol. 38, 07 2005, pp. 2289–2289.

[20] J. Chacón et al., "Enhancing ejss with extension plugins," Electronics, vol. 10, p. 242, 01 2021.

[21] L. de la Torre Cubillo et al., "Providing collaborative support to virtual and remote laboratories," IEEE Transactions on Learning Technologies, 10 2013.

[22] L. de la Torre et al., "Easy creation and deployment of javascript remote labs with ejss and moodle," in 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV), 2016, pp. 260–261.

[23] Garcia-Zubia et al., "An integrated solution for basics digital electronics: Boole-deusto and weblab-deusto," 02 2013, p. 1–5.