instituto
superior de
engenharia
de lisboa

Remote Lab

António Alves
Ângelo Azevedo

Advisor: Pedro Matutino

Report for Project and Seminar Class
Computer Science and Computer Engineering BSc

June 2025

Lisbon School of Engineering

Remote Lab

50539 António Alves

50565 Ângelo Azevedo

Advisor: Pedro Matutino

Report for Project and Seminar Class
Computer Science and Computer Engineering BSc

June 2025

Abstract

The design, development, implementation, and finally, the validation of digital systems require, in addition to simulators, the use of hardware to verify their implementations in real devices. In the current teaching paradigm, in which face-to-face time is reduced and remote and autonomous work is increased, it is necessary to create alternatives to the current model. The Remote Lab project aims to provide a virtual lab with access to remote hardware. This lab consists of a web application running on an embedded system. The web application, accessed via a website, aims to provide a dashboard where users can join a laboratory. This is where users can control the remote hardware. A hierarchy system will be implemented to provide different roles, each with their own permissions relative to how users can browse the information provided by the web application.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Motivation

In recent years, the need for remote access to laboratory resources has grown significantly, driven by the expansion of online education, research collaboration, and the increasing complexity of experimental setups. Traditional laboratories often require physical presence, which can limit accessibility and flexibility for students, researchers, and professionals. The **Remote Lab** project aims to address these challenges by providing a platform that enables secure, efficient, and user-friendly remote access to laboratory equipment and resources.

1.2 Objectives

The main objectives of the Remote Lab project are:

- To design and implement a scalable platform for remote laboratory access.
- To ensure secure authentication and authorization for different user roles.
- To provide an intuitive user interface for managing and scheduling laboratory sessions.
- To support integration with various types of laboratory hardware.

1.3 Scope

This project focuses on the development of the core platform, including backend services, user management, and basic hardware integration. Advanced features such as real-time data analytics, support for a wide range of laboratory devices, and extensive reporting capabilities are considered out of scope for the current phase.

1.4 Methodology

The project follows a modular and iterative development approach, leveraging modern software engineering practices. The backend is implemented using Kotlin and follows a layered

architecture, while the frontend is developed with Next.js to provide a responsive and accessible user experience.

1.5 Structure of the Document

The remainder of this report is organized as follows:

- **Chapter 2:** Related Work – Overview of existing solutions and technologies.
- **Chapter 3:** System Architecture – Description of the overall system design.
- **Chapter 4:** Implementation – Details of the main components and their interactions.
- **Chapter 5:** Evaluation – Assessment of the system's performance and usability.
- **Chapter 6:** Conclusions and Future Work – Summary of achievements and directions for future development.

Chapter 2

Placement

This chapter is organized into two sections, where we describe related work and some systems similar to the one developed in this project.

2.1 Related Work

In recent years, several initiatives have emerged to provide remote access to laboratory resources, especially in the context of higher education and research. Projects such as MIT's iLab and LabShare have demonstrated the feasibility and benefits of remote laboratories, enabling students and researchers to conduct experiments from anywhere in the world. These platforms typically focus on providing secure access, scheduling, and integration with a variety of laboratory equipment. The literature highlights the importance of usability, scalability, and security in the design of such systems, as well as the challenges associated with real-time interaction and hardware integration.

2.2 Similar Systems

There are several systems that offer functionalities similar to those of the Remote Lab project. For example, the iLab Shared Architecture (ISA) provides a framework for sharing laboratory equipment over the internet, supporting both batch and interactive experiments. LabShare is another notable example, offering a collaborative platform for remote experimentation and resource sharing among institutions. Other systems, such as WebLab-Deusto and VISIR, focus on specific domains like electronics and instrumentation, providing specialized interfaces and tools for remote experimentation. These systems serve as valuable references for the development of the Remote Lab platform, informing decisions related to architecture, user experience, and integration with laboratory hardware.

Chapter 3

Proposed Architecture

This chapter presents the proposed architecture for the Remote Lab platform, detailing its main components, their interactions, and the rationale behind the architectural choices.

3.1 System Overview

The Remote Lab platform is designed as a modular and scalable system, enabling secure and efficient remote access to laboratory equipment. The architecture follows a layered approach, separating concerns between the user interface, application logic, and hardware integration. This separation facilitates maintainability, extensibility, and the integration of new features or laboratory devices.

3.2 Main Components

The architecture consists of the following main components:

- **Frontend:** A web-based user interface developed with Next.js, providing users with access to laboratory resources, session scheduling, and experiment monitoring.
- **Backend:** Implemented in Kotlin, the backend exposes RESTful APIs for user management, authentication, authorization, and laboratory session control. It also handles business logic and enforces security policies.
- **Hardware Abstraction Layer:** This layer manages communication with laboratory equipment, abstracting hardware-specific details and providing a unified interface for the backend.
- **Database:** Stores user data, session information, access logs, and configuration settings. The database ensures data consistency and supports auditing requirements.
- **Authentication and Authorization:** Ensures secure access to the platform, supporting multiple user roles (e.g., students, professors, administrators) with different permissions.

3.3 Component Interactions

The components interact as follows:

- Users interact with the frontend to authenticate, schedule sessions, and access laboratory resources.
- The frontend communicates with the backend via secure API calls.
- The backend processes requests, applies business logic, and interacts with the database and hardware abstraction layer as needed.
- The hardware abstraction layer translates backend commands into device-specific instructions, enabling remote control of laboratory equipment.

3.4 Architecture Diagram

Figure 3.1 illustrates the high-level architecture of the Remote Lab platform.

Figure 3.1: High-level architecture of the Remote Lab platform.

3.5 Design Rationale

The architectural choices were guided by the need for scalability, security, and ease of integration with diverse laboratory equipment. The use of a layered architecture and standardized interfaces ensures that the platform can evolve to meet future requirements and support additional functionalities.

Chapter 4

Implemented Infrastructure

References