MAKERERE UNIVERSITY

COLLEGE OF ENGINEERING, DESIGN, ART AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Bachelor of Science in Computer Engineering

DESIGN AND IMPLEMENTATION OF AN OPEN INTERNET LABORATORY PLATFORM

A Research Thesis

Submitted in Partial Fulfillment of the Requirements for the award of the

Degree in Bachelor of Science in Computer Engineering of Makerere University

Name:	RAYMOND NAMBAALE	11/U/10077/PS	211005342
Research Partner:	MIRIAM ACHENG	10/U/10065/PS	210005833

DECLARATION

I, Raymond Nambaale, declare that the contents of this thesis are original and have never been produced elsewhere by anybody or submitted to any College, University or Institution for any academic award.

Signed………………………..

Date…………………………..

We, the supervisors, have approved this thesis. It meets the examiners requirements for the award of Degree of Bachelor of Science in Computer Engineering of Makerere University.

Main Supervisor

Mr. Maximus Baguma Byamukama
Assistant Lecturer
Department of Electrical & Computer Engineering
College of Engineering, Design, Art & Technology
Makerere University

Signature: ....

Date: ....

Co-Supervisor

Mr. David Amitu
Assistant Lecturer
Department of Electrical & Computer Engineering
College of Engineering, Design, Art & Technology
Makerere University

Signature: ....

Date: ....

DEDICATION

To my parents, whose unconditional love has helped me achieve great and innumerable things.

ACKNOWLEDGEMENTS

I would like to thank first and foremost, my parents, for the gift of life. I bare my heart's sincere appreciation to them for the blessing of love, education, support and encouragement over the years. Most of all, I would like to thank them for having gone over and above their means to ensure I am what I am today. Also to my siblings who were invaluable in my journey, I am eternally in your debt.

My profound and sincere gratitude goes to my project supervisors, Mr. Maximus Byamukama and Mr. David Amitu for the guidance, mentorship and direction given during the course of the project, without which, the path would have been lost. May your guidance continue to illuminate the path of other students the way it did to ours.

Special appreciation goes to Mr. Frank Odongkara and Mr. Ephraim Malinga for all the technical help offered in areas where the project got obscure especially in understanding the iLabs Shared Architecture.

I would also like to thank my research partner, Ms. Miriam Acheng, whose co-operation, hard work, diligence, selflessness and sacrifice made the research outcome a reality. It was a great honor and blessing to work with you. Appreciation goes to the various iLabs@MAK research team members for their valuable contribution to the project.

Above all, I would like to thank the Almighty God for the endless blessings and favor that He has given unto me despite all my shortcomings, truly everything else pales in comparison.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ASP	Active Server Pages
CEDAT	College of Engineering, Design, Art and Technology
ELVIS	Education Laboratory Virtual Instrumentation Suite
IEEE	Institute of Electric and Electronic Engineers
iLab	Internet Laboratory
ISA	iLabs Shared Architecture
Lab	Laboratory
LabVIEW	Laboratory Virtual Instrumentation Experimental Workbench
MAK	Makerere University, Kampala
MIT	Massachusetts Institute of Technology
NI	National Instruments
SRS	Software Requirement Specification
URL	Uniform Resource Locator
USB	Universal Serial Bus
AC	Alternating Current
DC	Direct Current
RMS	Root Mean Square
SDK	Software Development Kit
HTML	Hypertext Mark-up Language
CSS	Cascading Style Sheets

JS	JavaScript
API	Application Programming Interface
CSV	Comma Separated Values

ABSTRACT

This thesis discusses the design and implementation of an open source platform for running internet laboratories. It was undertaken at the College of Engineering, Design, Art and Technology under the iLabs@MAK project, to address the lack of flexibility that the National Instruments resources bring on board.

The pervasive interconnectivity of the Internet provides the means for entirely new kinds of laboratory experiences for the science and engineering student. For example, students can remotely access laboratory equipment that previously would have been too expensive or too dangerous for a typical student laboratory. Large-scale collaborative experiments between students separated geographically are also possible. The efforts for this project were to develop open source laboratory and Internet technologies to support relatively easy scientific exploration by the student. One outcome has been a straightforward and general solution using the website developed for student data retrieval of physical laboratory experiments over the Internet. With this project in place, students, universities and other organizations will be able to create, test and run a wide range of laboratories using open source solutions.

Chapter One: INTRODUCTION

1.1 Background

It is no secret that we are living in a time of incredible technological change. With the advent of more robust and stable international networks, the Internet and World Wide Web (Web) have dramatically changed the way we think about and engage in science and engineering. From "live" to "real-time" experiments, Internet applications and Web-based tools are providing infinite opportunities for scientific collaboration, problem-solving, and research. Projects such as the MIT iLabs project typify the innovative ways that Web technologies are being used to harness the collective intelligence and creativity of engineers to address pressing societal problems, improve research productivity, and challenge traditional views of "doing" engineering.

Engineering has also been impacted dramatically by technological change. Everything from online classroom management systems, virtual classrooms, and social networking tools such as Twitter, Facebook, WhatsApp and YouTube to the use of remote instrumentation and online labs are providing alternatives to traditional classroom and laboratory learning experiences. Initiatives such as the Massachusetts Institute of Technology's (MIT) iLabs project and Makerere University iLabs project provide examples of how institutions have embraced Web-based technologies for transforming science education.

Online laboratories are experimental setups that can be accessed through the Internet from a regular web browser. The initial inspiration for the first iLab came from the frustration that MIT's courses on semiconductor devices did not contain a laboratory component. Traditionally, students in these courses were exposed only to theoretical device models presented in lectures and course texts. At the same time, an Agilent semiconductor parameter analyzer, an expensive piece of equipment bought under a research contract, was sitting in a graduate research lab with spare capacity available. While the underutilization of the Agilent equipment seemed to provide an opportunity to have this tool also used in education, there was no way to accommodate the students taking courses using a single piece of equipment in the crowded research lab.

The iLabs@MAK project can provide a means for the development of an open source online laboratory platform which will provide sufficient wide of laboratory applications that can be supported. The advantages of these online laboratories are numerous, not to mention a reduction in the cost of obtaining equipment and software for setting up and running the online laboratories.

1.1.1 The iLabs@MAK Project

In February 2004, the iLabs project was introduced to the then Faculty of Technology (now College of Engineering, Design, Art and Technology) in Makerere University (MAK) by Professor Jesus del Alamo during his maiden visit. Subsequently, a memorandum of understanding was signed between MAK and MIT, leading to the formation of the iLabs@MAK Project under the umbrella of the iLab-Africa Project.

To facilitate access to the online Microelectronics Device Characterization laboratory (WebLab) set up at MIT, an iLab Service Broker was set up at MAK. The MIT WebLab was then utilized by the first year students offering Bachelor of Science in Electrical and Telecommunications Engineering programmes over the next three years. In 2007, a team of three undergraduate developers embarked on research that led to the establishment of the comprehensive iLabs infrastructure at Makerere University. The project then embarked on continuous research and development to fully integrate iLabs into the College curricula, with formation of an agile research team as a key component of the process. [1]

The iLabs Shared architecture was developed at the Centre for Educational Computer Initiatives (CECI) at MIT, to enable real laboratory equipment to be accessed and operated remotely through the Internet. It removes physical and time restrictions of the laboratory, giving an almost limitless number of students and teachers all around the world the chance to use specialist equipment to conduct experiments in fields such as microelectronics, chemical, structural and electrical engineering, as well as signal processing. Using this software, universities around the world can develop systems that enable remote access to their scientific equipment by other universities, schools and individuals, 24 hours a day, 7 days a week.

As of June 2011, thirty iLabs based on the ISA and powered by diverse hardware, had been developed at Makerere University, as shown in Table 1, and used by over one thousand five hundred students and staff.

In conclusion, the practical experience students obtain from the laboratories would go a long way in imparting knowledge and improving their competence as they would be exposed to some of the modern techniques being employed in the field today**.**

Table 1: iLabs developed at Makerere University as of June 2011

Field	Labs developed and utilized	Hardware platform
Analogue Electronics	Basic Semiconductor Device Analysis: Diodes, BJTs, FETs Operational Amplifiers and Associated Circuits	NI ELVIS II+ with the Free Scale Prototyping Board
Digital Electronics	Logic Gate Characterization and Application, Combinational Logic Circuits, Memory Elements, Counters and Shift Registers	NI ELVIS II+ with the Free Scale Prototyping Board and the Field Programmable Gate Array (FPGA) Board
Communications Engineering	Amplitude Modulation, Frequency Modulation, Pulse Code Modulation, Radio Frequency Techniques, Fiber Optics Transmission	NI ELVIS II+ with the Emona DATEx and Emona FOTEx Board
Digital Signal Processing	Digital Filters and Sound Effects	NI SPEEDY
Control Systems Engineering	Motor Modeling and Speed Control	Quanser 010 DC Motor Control Trainer
Renewable Energy	Solar Tracking	NI cRIO

1.2 Problem Statement

Working with current iLabs architecture poses challenges in that it is software and hardware specific, in which case both the hardware and software required is not readily affordable by most of the stakeholders in the engineering and technology field. In addition, there are space constraints where the equipment is hosted. Students have to perform the experiments in numerous shifts amidst time constraints in the timetable or the experiment will be available to only the Third year students who are fewer in number and as a result, most students in lower years will miss out on this practical opportunity.

1.3 Objectives

1.3.1 General Objective

The main objective is to design, construct and test an open source laboratory platform that incorporates the specific attributes of the iLabs Shared Architecture (ISA), as well as support a wide range of open source hardware and software.

1.3.2 Specific Objectives

The specific objectives arising from the main objective included:

1. To determine the system functional and non-functional requirements for the open source platform
2. To design the data acquisition module and interface hardware modules.
3. To develop a web application with intuitive algorithms for data acquisition, analysis and display.
4. To test the application as well as get involved in an entire revamp of the current iLabs framework.

1.4 Justification

The Open Laboratory System will enhance engineering and research projects due to its ability to support a wide range of platforms at an affordable cost. This is because, as opposed to the current internet laboratories, this system is not tied to run with a set of expensive equipment in terms of hardware or software

1.5 Scope

The geographical scope: The platform is intended for students in the CEDAT; particularly in Department of Electrical and Computer Engineering

The application scope: The platform is intended for students studying courses with concepts in full and half wave rectification at CEDAT as shown in the table below.

Table 2: Target Courses at CEDAT

Course	Course Unit	Credit Units
BSc. Computer Engineering	CMP: 1101 Electronics 1	4
BSc. Telecommunications Engineering	ELE: 1102 Physical Electronics	4
BSc. Electrical Engineering	ELE: 1104 Physical Electronics	4

1.6 Summary of Methodology

The research work was divided into four major phases; requirements specification, system design, implementation of a prototype and documentation as summarized in Table 3.

Table 3: Summary of Methodology

Milestones	Work Packages	Activities	Deliverables
M.1 Analysis	WP 1.1: Review of Literature	Review Literature on the current existing iLabs architecture	System Requirements Specification
	WP 1.2 : Interaction with stakeholders	Visits to the server room where the current iLabs are set up from.	
	WP 1.3: Benchmarking	Best practices review with reference to the equipment	

		manufactured by National Instruments.	
	WP 1.4: System functional requirements determination	Specifying target Functionalities	
	WP 1.5: System quality attributes	Specifying system usability, portability, inter-operability	
M.2 System Design	WP 2.1 Design interface circuitry		Interface Designs
	WP 2.2 Design website user interface		
	WP 2.3 Develop data acquisition, processing and analysis algorithms	Choose which equipment are best for acquiring experiment readings, processing and Analysis	
	WP 2.3: Choose components and development environments	Specify which languages will be best for development of the user interfaces and which experiment to use for the proof of concept.	
M.3 Implementation	WP 3.1 Building Experiments	Using a breadboard and connecting components for the experiment and being able to control the experiment remotely	A solar PV System iLab Prototype
	W.P 3.2 Development of the website Application	Using PHP as the core backend language in addition to other web engineering languages such us HTML and CSS to come up with the user interface	
	W.P 3.3 Testing and Debugging the	Testing the prototype laboratory using the website interface to carry	

	Platform	out the chosen experiment.	
M.4 Documentation	W.P 4 Writing the necessary documentation	Writing the Thesis Report	Comprehensive Documentation
		Writing user manuals	

1.7 Report Outline

In this chapter, the Research Background, Problem Statement, Objectives, Scope, Summary of Methodology and Justification have been presented.

The other sections of the report have been presented under four major sections namely;

Chapter Two: Literature Review, which presents the theoretical principles of half wave rectification, factors affecting open internet laboratories lab. The inter model and its advantages are also presented.

Chapter Three: Methodology, gives a detailed account of the procedure that was followed to achieve the research objectives.

It includes the step by step guide to how the lab was implemented; from requirements specification to deriving system designs and then implementing the designs into a prototype.

Chapter Four: Conclusions, Challenges and Recommendations, presents a Summary of the report and research results, Contribution of the Research, Limitations, Challenges and Recommendations arising from the undertaken research.

Chapter Two: LITERATURE REVIEW

2.1 Introduction

In this chapter, theoretical concepts of half wave rectification are presented. The working principle, applications, influencing factors and limitations are discussed. The Open Internet Laboratories Architecture, being proposed by this thesis is also discussed.

2.2 Rectification

One of the most important applications of diodes is in the design of rectifier circuits. A diode rectifier forms an essential building block of the dc power supplies required to power electronic equipment. A block diagram of such a power supply is shown in the Figure 1 below. As indicated, the power supply if fed from the 120-V(rms) 60-Hz ac line, and it delivers a dc voltage V_o (usually in the range of 5 – 20 V) to an electronic circuit represented by the *load* block. The dc voltage V_o is required to be as constant as possible in spite of the variations in the ac line voltage and in the current drawn by the load.

The first block in a dc power supply is the **power transformer.** It consists of two separate coils wound around an iron core that magnetically couples the two windings. The **primary winding,** having N_1 turns, is connected to the 120-V ac supply, and the **secondary winding,** having N_2 turns, is connected to the circuit of the dc power supply [2].

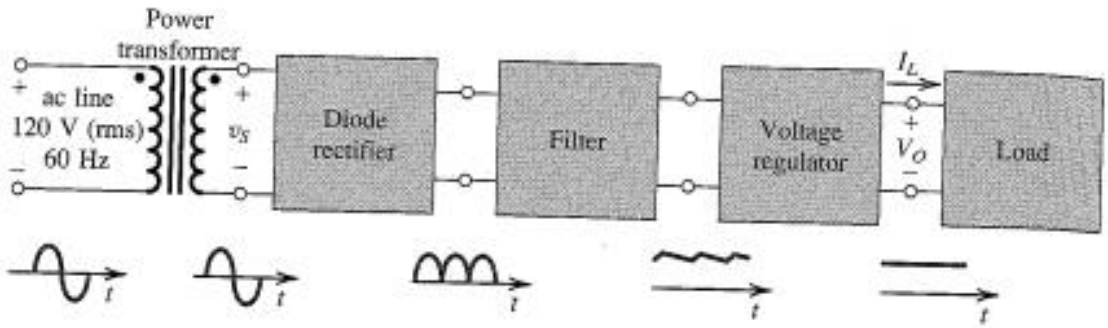


Figure 1: Block diagram of a dc power supply [2]

Thus an AC voltage v_s of $120(N_2/N_1)$ V (rms) develops between the two terminals of the secondary winding. By selecting an appropriate turns ratio (N_1/N_2) for the transformer, the designer can step the line voltage down to the value required to yield the particular dc voltage output of the supply. For instance, a secondary voltage of 8-V rms may be appropriate for a dc output of 5V. This can be achieved with a 15:1 turns ratio.

In addition to providing the appropriate sinusoidal amplitude for the dc power supply, the power transformer provides electrical isolation between the electronic equipment and the power-line circuit. This isolation minimizes the risk of electric shock to the equipment user.

The diode rectifier converts the input sinusoid v_s to a unipolar output, which can have the pulsating waveform indicated in Fig 1. Although this waveform has a nonzero average or a dc component, its pulsating nature makes it unsuitable as a dc source for electronic circuits, hence the need for a filter. The variations in the magnitude of the rectifier output are considerably reduced by the filter block in Fig 1. In the following section we shall study a number of rectifier circuits and a simple implementation of the output filter.

The output of the rectifier filter, though much more constant than without the filter, still contains a time dependent component, known as a **ripple**. To reduce the ripple and to stabilize the magnitude of the dc output voltage of the supply against variations caused by changes in load current, a voltage regulator is employed. Such a regulator can be implemented using a zener shunt regulator configuration. Alternatively, and much more commonly at present, an integrated-circuit regulator can be used [2].

2.2.1 The Half-Wave Rectifier

The half-wave rectifier utilizes alternate half-cycles of the input sinusoid. Figure 2 shows the circuit of a half-wave rectifier. This circuit assumes an ideal diode. Using the more realistic battery-plus-resistance diode model, we obtain the equivalent circuit shown in Fig. 3, from which we can write

$$v_o = 0, v_s < v_{D0}$$

$$v_0 = \frac{R}{R + r_D} v_s - v_{D0} \frac{R}{R + r_D}, v_s \geq v_{D0}$$

The transfer characteristics represented by these equations is sketched in Fig. 4. In many applications, $r_D \ll R$ and the second equation can be simplified to

$$v_0 \approx v_s - v_{D0}$$

Where$v_{D0} = 0.7\ V\ or\ 0.8\ V$. Figure 5 shows the output voltage obtained when the input is a sinusoid.

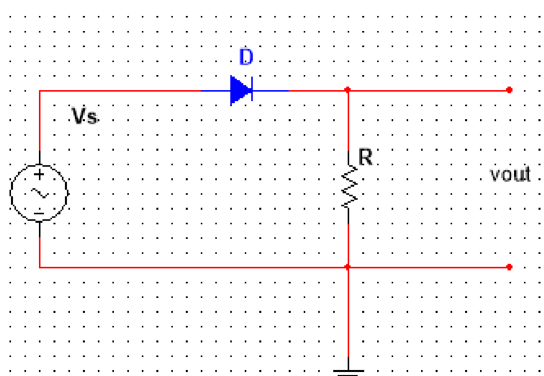


Figure 2: Half wave rectifier

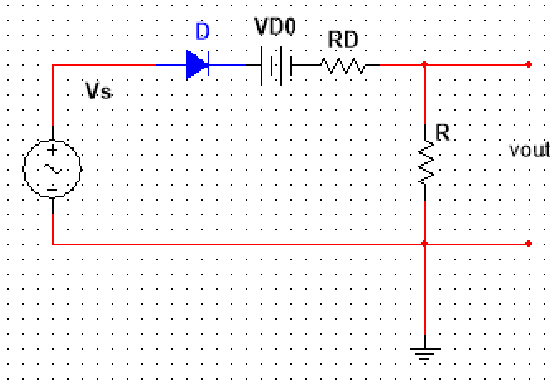


Figure 3: Equivalent circuit of the half-wave rectifier with the diode replaced with its battery-plus-resistance model

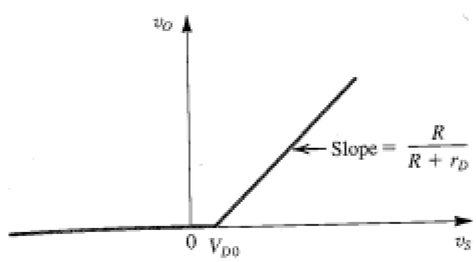


Figure 4: Transfer characteristic of the rectifier circuit

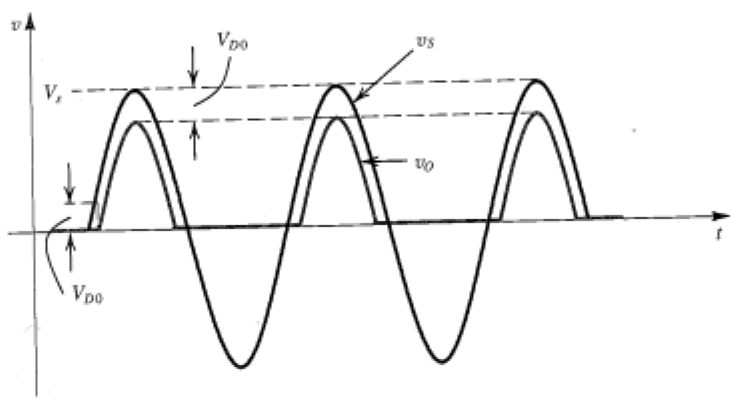


Figure 5: Input and output waveforms

In selecting diodes for rectifier design, two important parameters must be specified: the current-handling capability of the diode, determined by the largest current the diode is expected to conduct, and the **peak inverse voltage** (PIV) that the diode must be able to withstand without breakdown, determined by the largest reverse voltage that is expected to appear across the diode. In the rectifier of Fig. 2, we observe that when v_s is negative the diode will be cut off and v_0 will be zero. It follows that the PIV is equal to the peak of v_s,

$$PIV = V_s$$

It is usually prudent, however, to select a diode that has a reverse breakdown voltage at least 50% greater than the expected PIV.

In summary, there are two important notes about the half-wave rectifier. First, it is possible to use the diode exponential characteristic to determine the exact transfer characteristic of the rectifier. However, the amount of work involved is usually too great to be justified in practice. Of course, such analysis can easily done using a computer circuit analysis program.

Second, whether we analyze the circuit accurately or not, it should be obvious that this circuit does not function properly when the input signal is small. For instance, this circuit cannot be used to rectify an input sinusoid of 100-mV amplitude. For such an application one resorts to a so-called precision rectifier, a circuit utilizing diodes in conjunction with op amps [2].

2.2.2 Full-Wave Rectifier

The full-wave rectifier utilizes both halves of the input sinusoid. To provide a unipolar output, it inverts the negative halves of the sine wave. One possible implementation is shown in Fig. 6. Here the transformer secondary winding is **center-tapped** to provide two equal voltages v_s across the two halves of the secondary winding with the polarities indicated. Note that when the input line voltage (feeding the primary) is positive, both of the signals labelled v_s will be positive. In this case D_1 will conduct and D_2 will be reverse biased. The current through D_1 will flow through R and back to the center tap of the secondary. The circuit then behaves like a half-wave rectifier, and the output during the positive half-cycles when D_1 conducts will be identical to that produced by the half-wave rectifier.

Now, during the negative half cycle of the ac line voltage, both of the voltages labelled v_s will be negative. Thus *D1* will be cut off while *D2* will conduct. The current conducted by *D2* will flow through *R* and back to the center tap. It follows that during the negative half-cycles while *D2* conducts, the circuit, the circuit again behaves as a half-wave rectifier. The important point, however, is that the current through *R* always flows in the same direction, and thus v_0 will be unipolar, as indicated in Fig. 8. The output waveform shown is obtained by assuming that a conducting diode has a constant voltage dropv_D. Thus the transfer characteristic of the full-wave rectifier takes the shape shown in Fig. 7.

The full-wave rectifier obviously produces a more "energetic" waveform that that provided by the half-wave rectifier. In almost all rectifier applications, one opts for a full-wave type of some kind.

To find the PIV of the diode in the full-wave rectifier circuit, consider the simulation during the positive half-cycles. Diode *D1* is conducting, and *D2* is cut off. The voltage at the cathode of *D2* isv_0, and that at its anode is$-v_s$. Thus the reverse voltage across *D2* will be$(v_0 + v_s)$, which will reach its maximum when v_0 is at its peak value of $(v_s + v_{D0})$ and $v_s$is at its peak value ofV_s; thus,

$$PIV = 2V_s - V_D$$

Which is approximately twice that for the case of the half-wave rectifier [2].

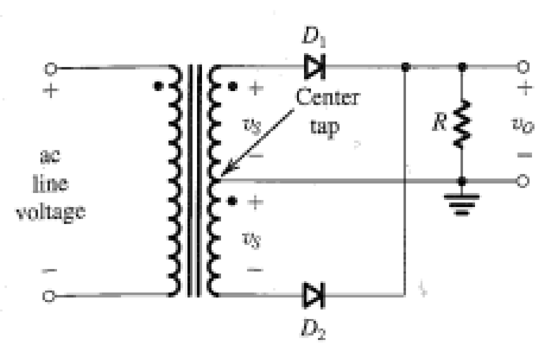


Figure 6: Full-wave rectifier circuit

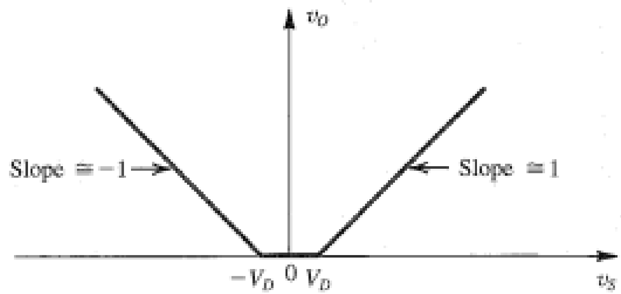


Figure 7: Transfer Characteristic assuming constant voltage drop model for the diodes

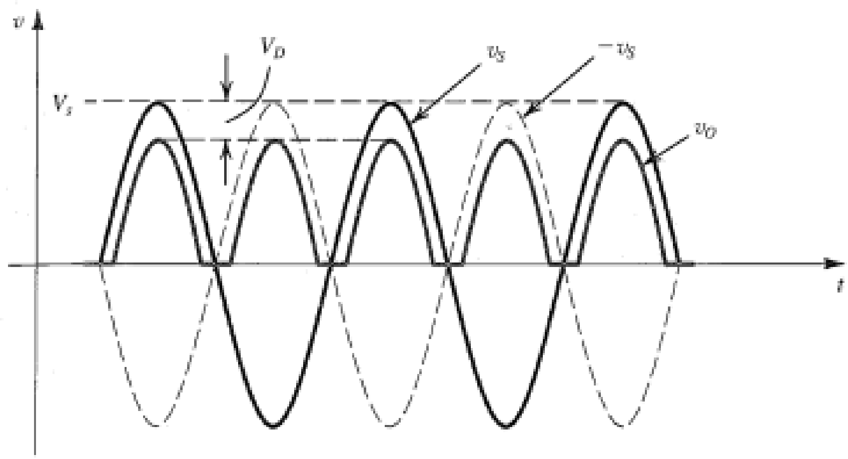


Figure 8: Input and Output waveforms [2]

2.2.3 The Bridge Rectifier

An alternative implementation of the full-wave rectifier is shown in Fig. 9. he circuit, known as the bridge rectifier because of the similarity of its configuration to that of the Wheatstone bridge, does not require a center-tapped transformer, a distinct advantage over the full-wave rectifier discussed above. The bridge rectifier, however, requires four diode as compared to two in the previous circuit. This is not much of a disadvantage, because diodes are inexpensive and one can buy a diode bridge in one package.

The bridge rectifier circuit operates as follows: During the positive half-cycles of the input voltage, v_s is positive, and thus current is conducted through diode D_1, resistor R, and diode D_2. Meanwhile, diodes D_3 and D_4 will be reverse biased. Observer that there are two diodes in series in the conduction path, and thus v_0 will be lower than v_s by two diode drops (compared to one drop in the circuit previously discussed). This is somewhat of a disadvantage of the bridge rectifier.

Next, consider the situation during the negative half-cycles of the input voltage. The secondary voltage v_s will be negative, and thus $-v_s$ will be positive, forcing current through D_3, R, and D_4. Meanwhile, the diodes D_1 and D_2 will be reverse biased. The important point to note, though, is that during both half-cycles, current flows through R in the same direction (from right to left), and thus v_0 will always be positive, as indicated in Fig. 10.

To determine the peak inverse voltage (PIV) of each diode, consider the circuit during the positive half-cycles. The reverse voltage across D_3 can be determined from the loop formed by D_3, R, and D_2 as

$$v_{D3}(reverse) = v_o + v_{D2}(foward)$$

Thus the maximum value of v_{D3} occurs at the peak of v_o and is given by

$$PIV = V_s - V_D$$

Observe that here the PIV is about half the value for the full-wave rectifier with a center-tapped transformer. This is another advantage of the bridge rectifier.

Yet one more advantage of the bridge rectifier circuit over that utilizing the center-tapped transformer is that only about half as many turns are required for the secondary winding of the transformer. Another way of looking at this point can be obtained by observing that each half of the secondary winding of the center-tapped transformer is utilized for only half the time. These advantages have made the bridge rectifier the most popular rectifier circuit configuration.

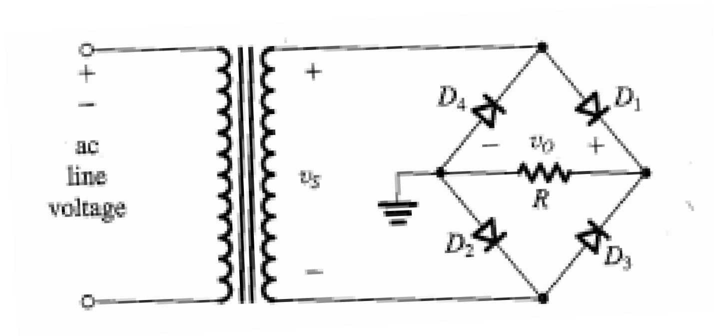


Figure 9: Bridge Rectifier

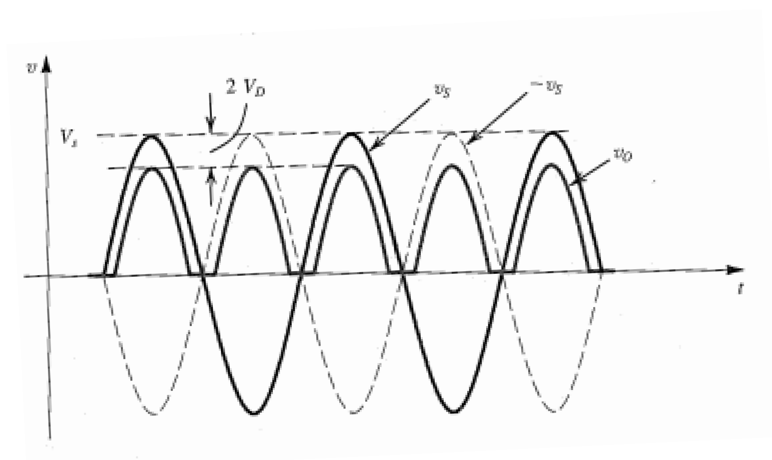


Figure 10: Input and Output Waveforms [2]

2.2.4 The Rectifier with a Filter Capacitor – The peak Filter

The pulsating nature of the output voltage produced by the filter circuits discussed above makes it unsuitable as a dc supply for electronic circuits. A simple way to reduce the variation of the output voltage is to place a capacitor across the load resistor. It will be shown that this **filter capacitor** serves to reduce substantially the variations in the rectifier output voltage [3].

To see how the rectifier circuit with a filter capacitor works, consider first the simple circuit shown in Fig. 11. Let the input v_i be sinusoidal with a peak value ofV_p, assume the diode to be ideal. As v_i goes positive, the diode conducts and the capacitor is charged so that$v_o = v_i$. The situation continues until v_i reaches its peak valueV_p. Beyond the peak, as v_i decreases the diode becomes reverse biased and the output voltage remains constant at the value V_p. In fact, theoretically speaking, the capacitor will retain its charge hence its voltage indefinitely, because there is no way for the capacitor to discharge. Thus the circuit provides a dc voltage output equal to the peak of the input sine wave. This is a very encouraging result in view of our desire to produce a dc signal.

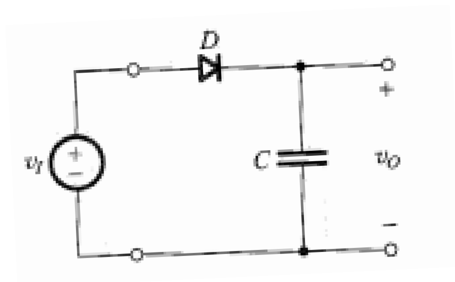


Figure 11: A simple circuit to illustrate the effect of a filter capacitor

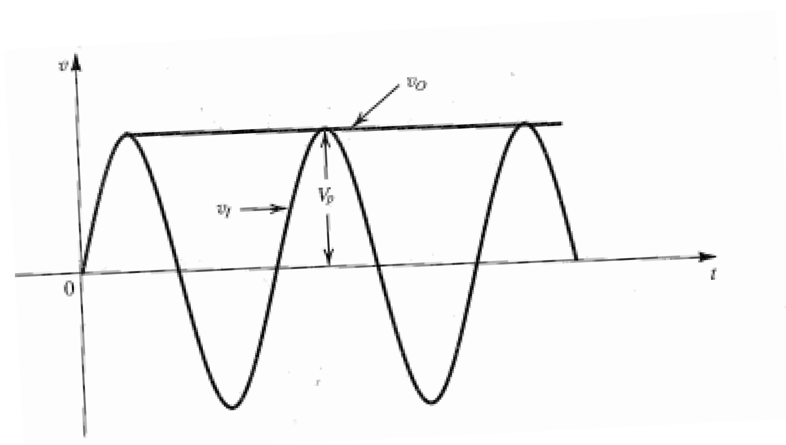


Figure 12: Input and Output Waveforms assuming an ideal diode

Next, we consider the more practical situation where a load resistance *R* is connected across the capacitor *C*, as depicted in Fig. 13. However, we will continue to assume the diode to ideal. As before, for a sinusoidal input, the capacitor charges to the peak of the input V_p. Then the diode turns off, and the capacitor discharges through the load resistance *R*. The capacitor discharge will continue for almost the entire cycle, until the time at which v_i exceeds the capacitor voltage. Then the diode turns on again and charges the capacitor up to the peak ofv_i, and thee process repeats itself. Observe that to keep the output voltage from decreasing too much during capacitor discharge, one selects a value for *C* so that the time constant *CR* is much greater than the discharge interval.

We are now ready to analyze the circuit in detail. Figure 14 shows the steady-state input and output voltage waveforms under the assumption that$CR \gg T$, where *T* is the period of the input sinusoid. The waveforms of the load current

$$i_L = \frac{v_o}{R}$$

and of the diode current (when it is conducting)

$i_D = i_C + i_L$

$= C\frac{dv_i}{dt} + i_L$

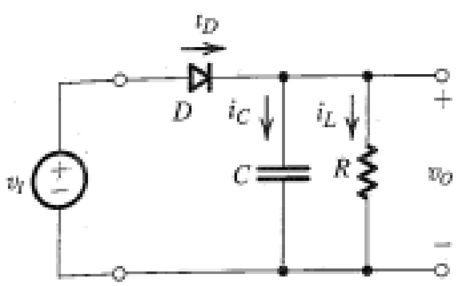


Figure 13: A practical circuit to illustrate the effect of a filter capacitor

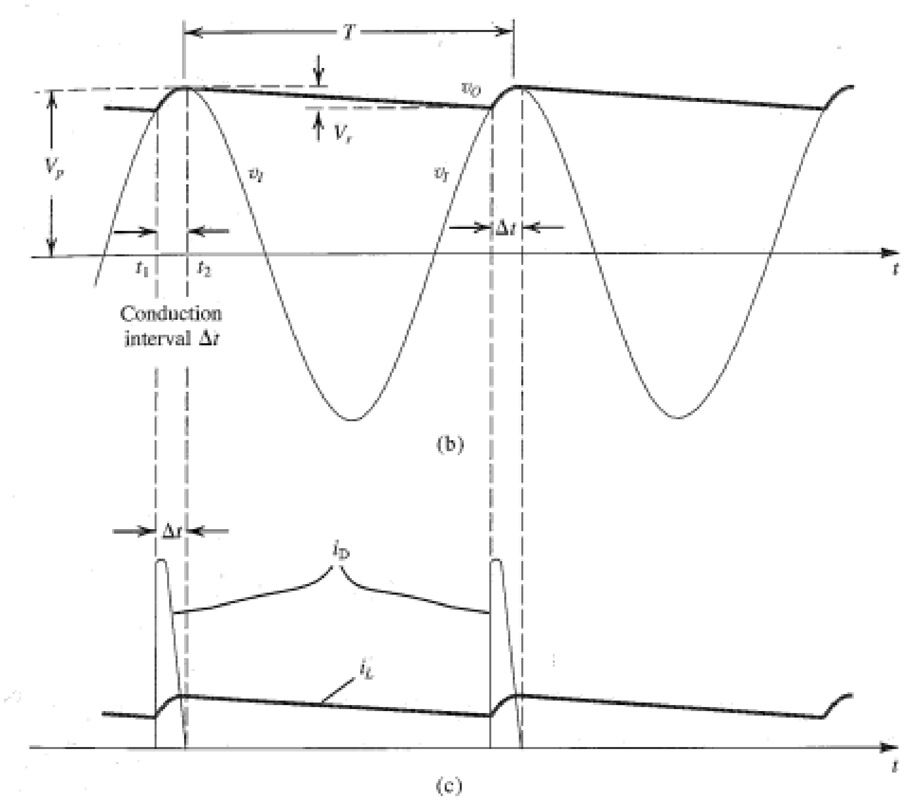


Figure 14: Input and Output waveforms [2]

2.3 The iLabs Shared Architecture (ISA)

2.3.1 Introduction

iLabs stands for Internet laboratories. iLabs are remote online laboratories that allow users to perform experiments through the internet. They allow students to complement their theoretical calculations and results with real data; providing them with a better understanding of concepts in science and engineering.

The purpose of the ISA is to support internet-accessible laboratories and promote their sharing among schools and universities on a worldwide scale. iLabs were originally just simulations hosted on the internet. Today, scientific instrumentation and experimentation has become more expensive and distance education has become more common. The primary goal was to provide students with access to real laboratory equipment no matter where the equipment was physically hosted so as to share experimentation resources worldwide.

The ISA is a web service infrastructure that was developed to provide a unifying software framework that can support access to a wide variety of online laboratories to different people globally over the internet. Users access these remote laboratories through a single sign-on session and a simple standard administrative interface. [4] The ISA basically provides platform-independent lab development, scalable access for students, and efficient management for lab providers.

Three tiers make up the communication framework for the ISA middleware as shown in Figure 12 namely;

- The first tier is the student's client application that either runs as a Java applet or a LabVIEW front panel embedded in the user's browser.
- The middle tier, called the Service Broker, provides the shared common services. It is backed by a standard relational database such as SQL Server or MySQL. The student's client communicates solely with the service broker, which forwards experiment specifications to the final third tier that includes the lab equipment.
- The third tier is the Lab Server, which interfaces with the instruments that execute the specified experiments. The Lab Server notifies the service broker when the results are ready to be retrieved.

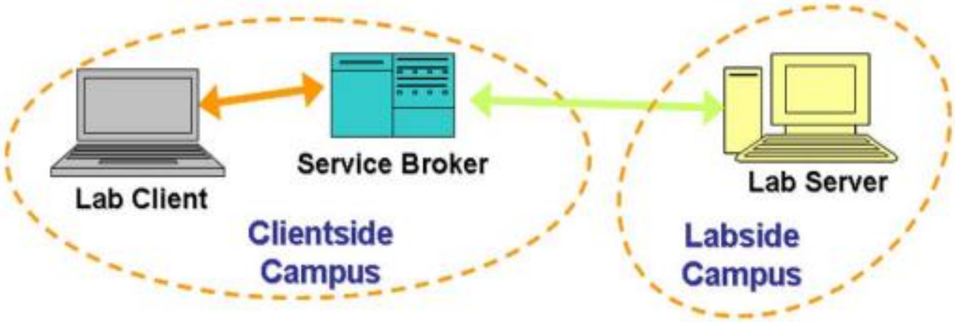


Figure 15: The standard ISA model

The ISA supports three modes of experiments i.e. Batched, Interactive and Sensor experiments. The iLabs@MAK project has adopted the Batched and Interactive experiment modes therefore only these will be discussed.

2.3.2 Batched iLabs Shared Architecture

Batched experiments are those in which the entire course of the experiment is specified before the experiment begins. Experiment execution takes place in machine time. The client communicates with the Lab Server through the service broker as shown in Figure 13. The client is where the user specifies parameters that will be used in the experiment. It is a Java Applet that is launched from the Service Broker as shown in Figure 14.

2.3.3 Interactive iLabs Shared Architecture

Interactive experiments are those in which the user monitors and controls one or more aspects of the experiment during its execution. Experiment execution takes place in human time.

The Service Broker serves to authenticate the user and schedule a time slot in which he/she can perform the experiment. It establishes a relationship between the client and the Lab Server. Once this relationship is established, the client and Lab Server then communicate directly. This interaction is done via a LabVIEW VI embedded in the client's web browser.

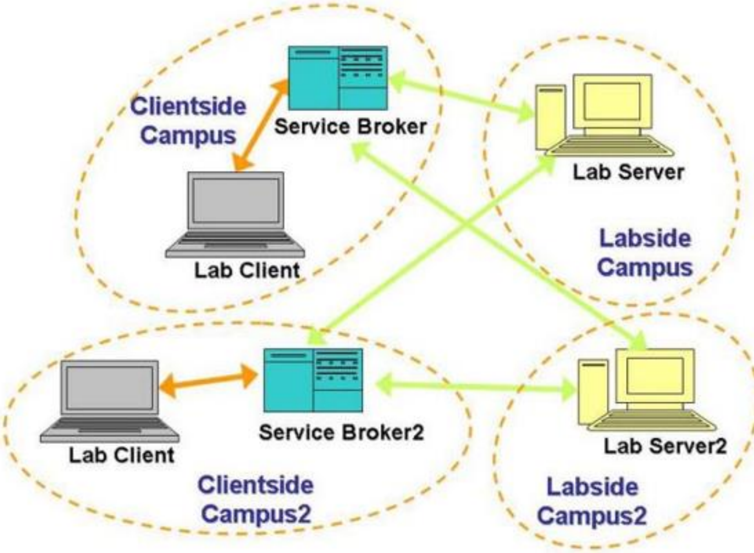


Figure 16: Model of the Batched ISA

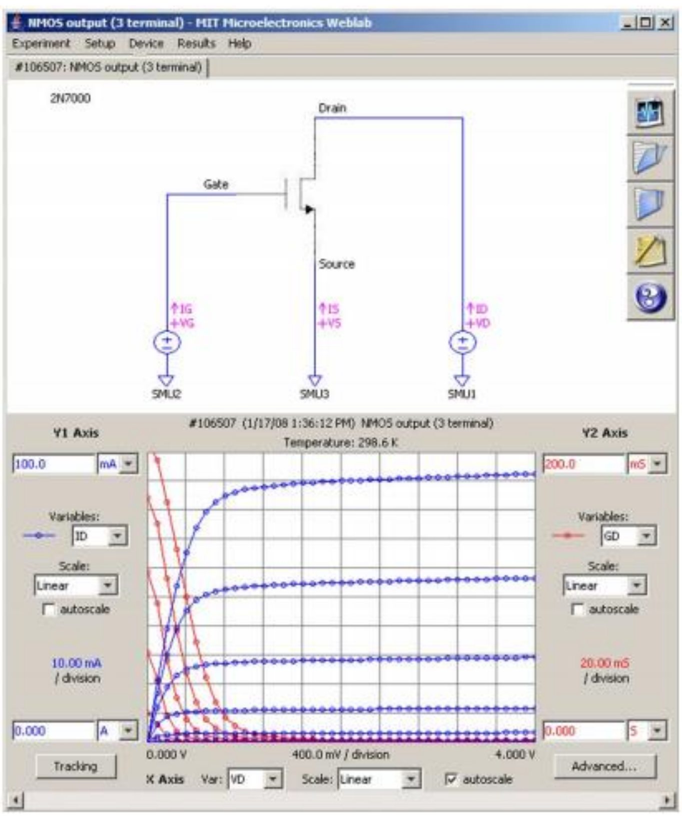


Figure 17: The batched ISA Java Applet client interface

The interactive iLabs Shared Architecture middleware consists of a host of five web services that perform different roles. These include;

iLab Service Broker (ISB) - This is responsible for user authentication and authorization as well as launching the User Side Scheduling service.

User-side Scheduling Server (USS) – This is a scheduling service that enables the user to store and redeem time reservations, within which they will have dedicated control of the laboratory hardware.

Lab-side Scheduling Server (LSS) - This is also a scheduling service that performs the same function as the USS on the Lab Server side. Decoupling the LSS and USS allows the development of each to proceed independently. For example, a university that wants to implement an innovative user scheduling policy can do so without needing to modify the scheduling policy for the lab server whose LSS may be located at another university and controlled by different staff. [4]

Experiment Storage Server (ESS) - This is an independent web service that handles the potentially high-bandwidth traffic from the client and Lab Server during experiment execution. It stores both XML and binary data from a particular experiment (experiment data).

Interactive Lab Server (ILS) - This web service is responsible for processing experiment execution requests. Once a request is validated, the ILS initializes the experiment, feeds data back to the user interface in the client, stores experimental data on the ESS, and closes down the experiment after the reservation has expired as shown in Figure 15.

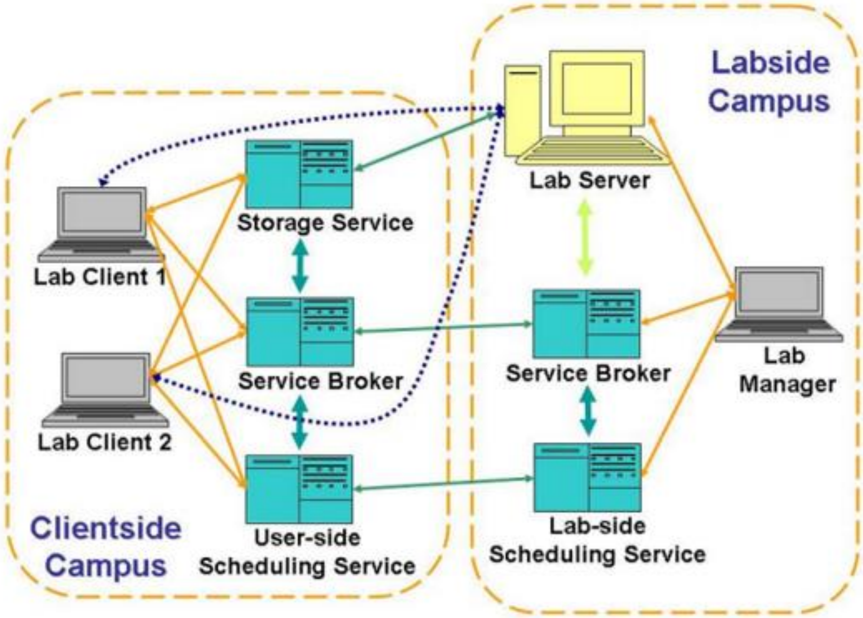


Figure 18: Model of the Interactive ISA

Chapter Three: METHODOLGY

3.1 Introduction

This section presents a chronology of processes undertaken to accomplish the research objectives.

It analyses the methods, tools and processes that were used in the design and implementation of the open source internet laboratory platform. It starts with a requirements specification of the system which then leads to design of the system information and development architectures. A description of the hardware and software modules development is then given; all of which constitute the prototype implementation.

3.2 Requirements Specification

This was developed to determine the pertinent features necessary for the development of an experiment on an open source internet laboratory platform. It comprises of a functional and non – functional requirements analysis and their respective deliverables.

3.2.1 Review of Literature

This was broken down into review of the College curricula and pertinent documentation such as scientific papers, journals, books and other relevant literature.

Review of Curricula

This was done to establish the relevant courses and particular course units for which the test experiment would apply. The courses and respective course units were established as shown in table 6.

Review of Documentation

The documentation review took a foundational understanding of full-wave and half-wave rectification concepts and features that could be investigated in a practical laboratory instance of experiments in those fields. It involved mainly reviewing scientific journals and papers that showcased published work that was done in the field of electronics and remote laboratories in existence worldwide.

In light of this, documentation, scientific papers and journals that had been published on remote educational labs in general were used.

Table 4 : Target courses of the rectifier lab within CEDAT

Course	Course Unit	Credit Units
BSc. Computer Engineering	CMP 1101 Electronics 1	4
BSc. Telecommunications Engineering	ELE 1102 Physical Electronics	4
BSc. Electrical Engineering	ELE 1104 Physical Electronics	4

3.2.2 Determining Functional Requirements

This involved determining the features of the Open source Internet Laboratory platform that were pertinent to the remote laboratory.

An analysis was made to derive use case diagrams. The deliverable to this work package was a comprehensive functional requirements specification document that describes the use cases as shown in Figure 19 below.

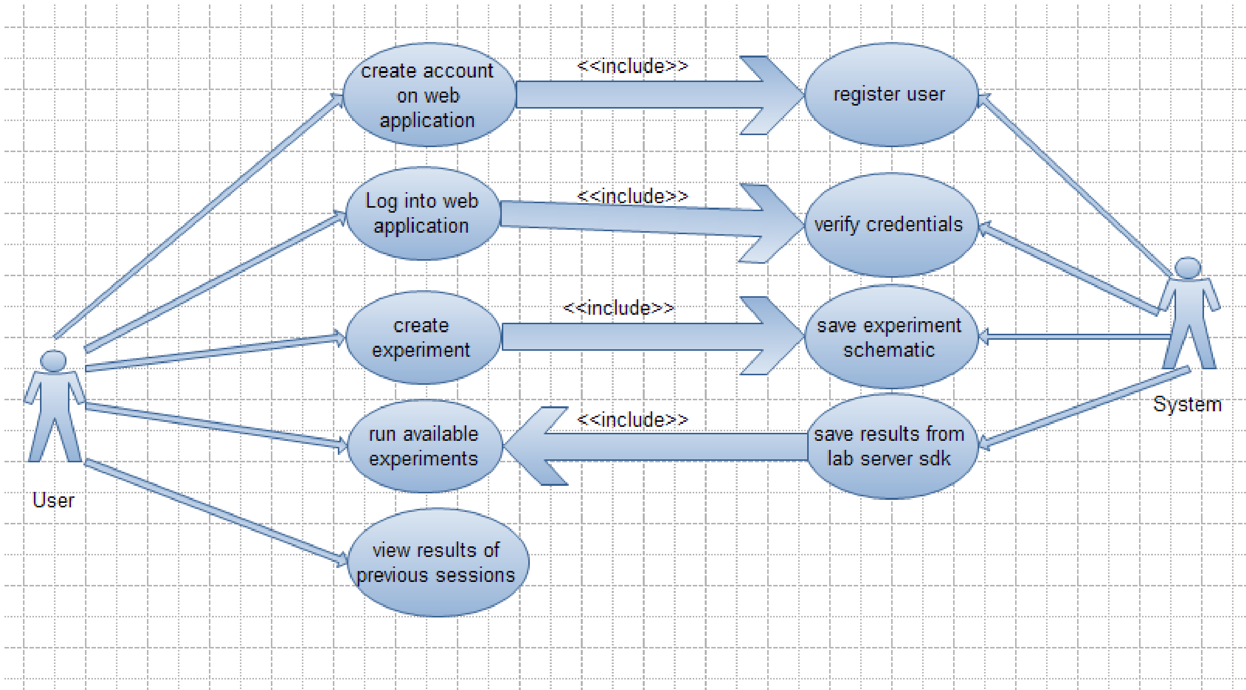


Figure 19: System Use Case diagram

3.2.3 Determining Non – Functional Requirements

Determining the non – functional architecture involved establishing the system quality attributes such as usability, reliability, availability, security, validation, authentication and other key performance factors that were considered.

System Quality Attributes

Usability

The platform was designed such that the user is only required to have basic computer knowledge, and a basic understanding of the concept he/she is trying to study i.e. the user should have some prior basic knowledge of electronic circuits.

The user is required to register on the web system in order to run or create experiments for other users.

Reliability and Availability

Any lab on this platform should be available full time unless there is a power cut or internet disconnection, so that any of the registered users can access it as long as they have an internet connection.

Security

The user can only attain access to his/her account after logging in and filling in the right credentials. No other person has access to user accounts or account session results.

Maintainability, Extensibility and Scalability

The open internet laboratory platform is highly extensible, supports a wide range of open source technologies, hence it is highly scalable.

3.3 System Design

This stage of project implementation involved designing the context and deployment architectures. These provided a high level view of system interaction and data flow within the system. The required software and hardware was determined at this stage.

3.3.1 System Information Architecture

The information architecture describes the relationships between the different components that make up the open internet laboratory platform.

It features a context diagram of the system and indicates interconnection of the different components. The system contextual diagram is illustrated in Figure 20.

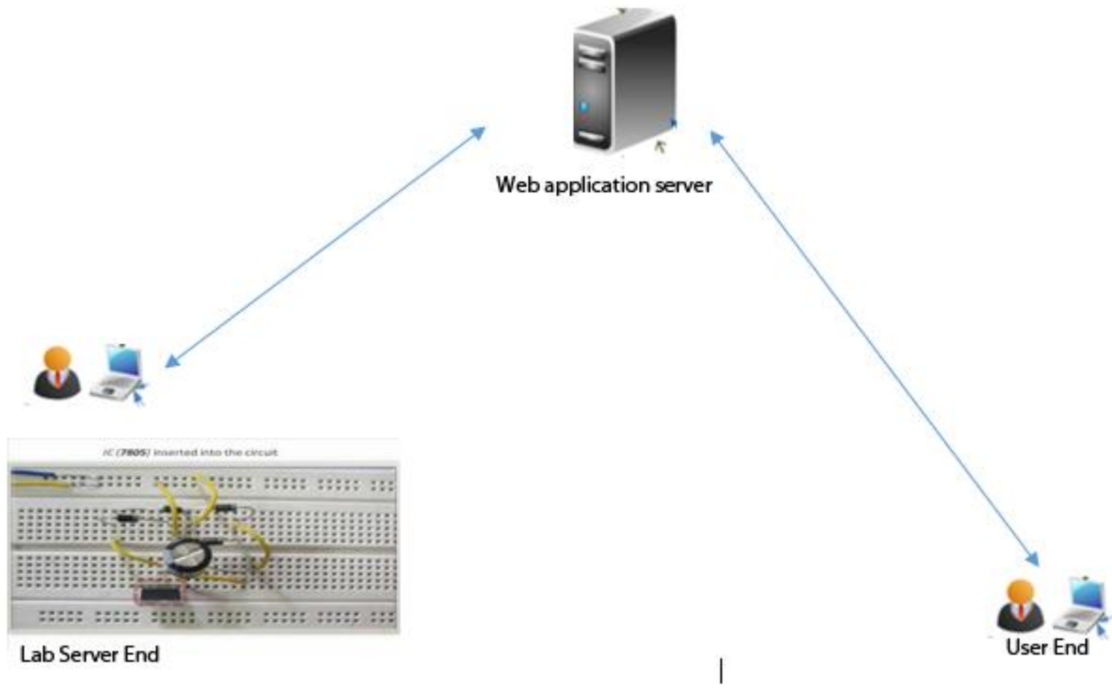


Figure 20: System Contextual Diagram

3.3.2 System Development Architecture

This architecture describes the system hardware modules, their specifications and functionalities. These modules include the laboratory server sdk module and the data acquisition module. In addition to the hardware modules, the website application interfaces that were to be used were developed and their conceptual interaction was described at this stage.

3.4 Circuit Wiring

To implement the test experiment on a breadboard, the various resistances and sources were connected as shown in the circuit diagram in Figure 21. The data acquisition module I used did not have the capability to sample negative values of the signal. In order for the module not to lose any data points, we had to introduce a DC component into the AC signal. In this case the signal would no longer oscillate about the time axis, but rather a certain positive value. This implied that the lowest data point of the AC input signal was a positive value. This error that we deliberately introduced was taken care of as results were represented in website application. The voltage divider containing 100kΩ resistors was to provide a 2.5V signal to be the ground of the circuit.

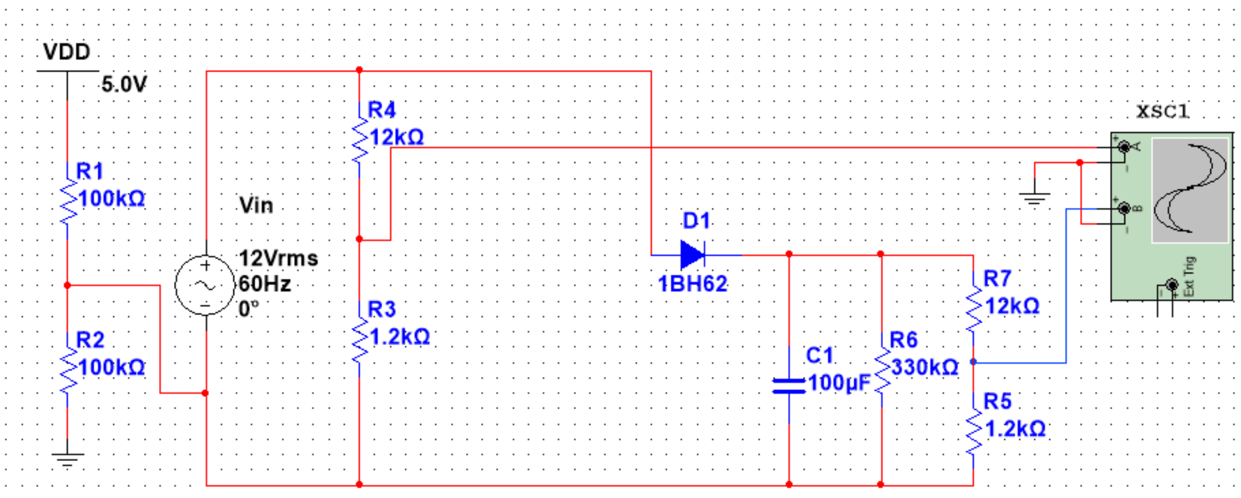


Figure 21: Laboratory circuit wiring

The shifted AC signal is then fed into the half-wave rectifier circuit.

3.5 Prototype Implementation

This section gives a description of the process of implementing the prototype systems namely, the software configuration and hardware configuration. Development of the prototype was categorized into two modules namely; Openlabs Website application development, and Data acquisition.

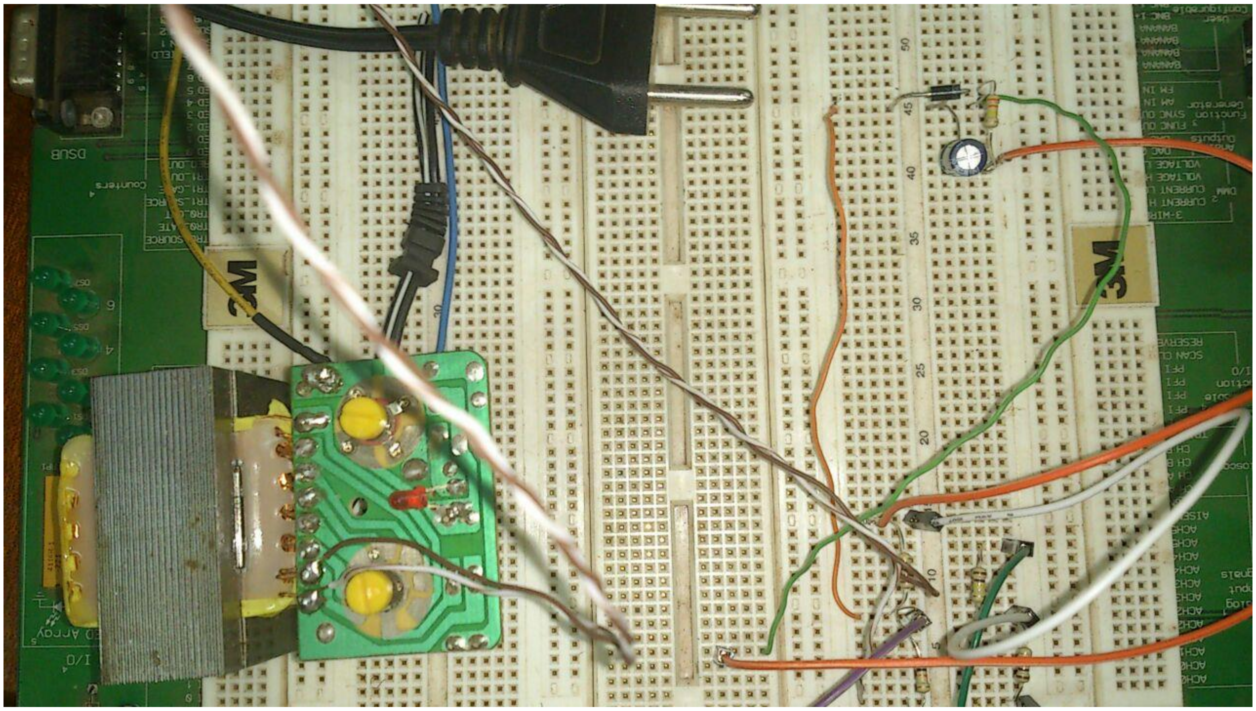


Figure 22: Circuit wiring diagram on a bread board

3.5.1 Openlabs Website Application Development

This was the biggest module because it is what supports the core framework of the openlabs including its nature as an open source platform for online laboratories.

PHP5 was used as the core scripting server side language. PHP is not only open source but is also feature rich and has all the functionality that a proprietary or paid scripting language would offer, it has the ability to run on all major operating systems like Linux, UNIX, Mac OS and windows, and finally it has multiple layers of security to prevent threats and malicious attacks. These are just a few of the many advantages PHP has over other scripting languages in the business.

In addition, other website engineering technologies like HTML, CSS, and JS we used to implement the interfaces and additional functionality of the website application.

The website application was supported by a database based on MySQL for storing user credentials, user sessions, and experiment configurations.

The website application had the following modules;

- User registration and authentication. This module generally focused on user management as well as security.

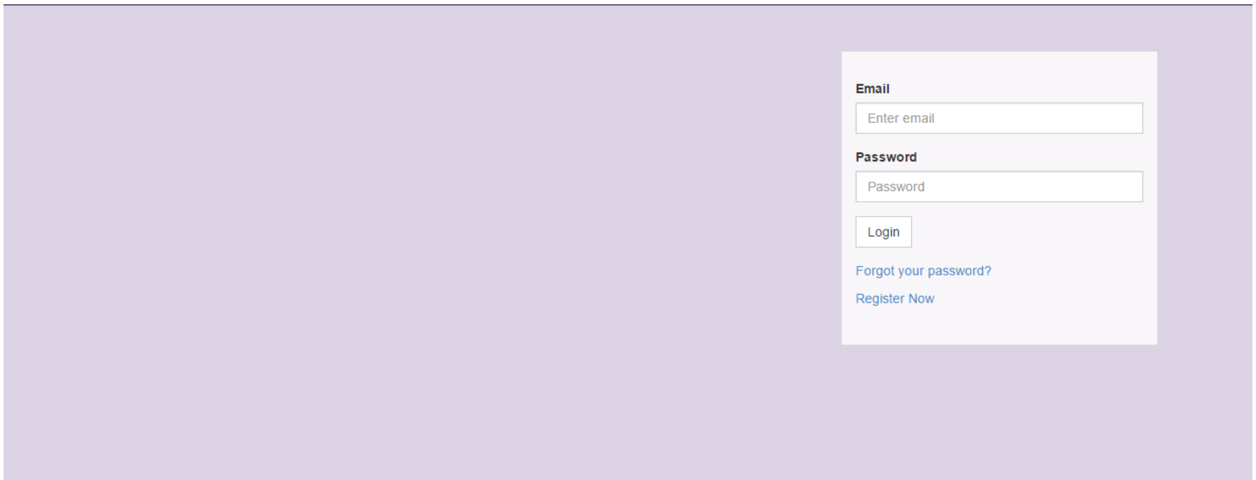


Figure 23: Login screen

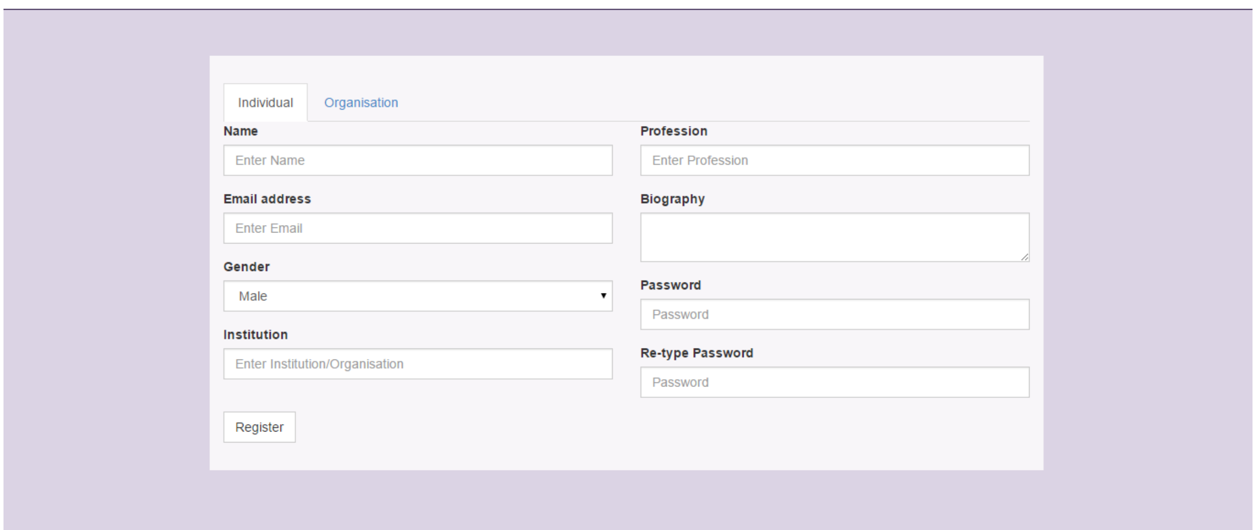


Figure 24: Registration Screen

- Experiments module: This module involved creation of an interface where a logged in user can add an experiment. When a user adds an experiment, they need to provide information such as the experiment background, procedure and schematic. For this particular module, a representation of the basic electronic components had to be designed such that the creator(s) design the circuit schematic by drag and drop of the components they are interested in.

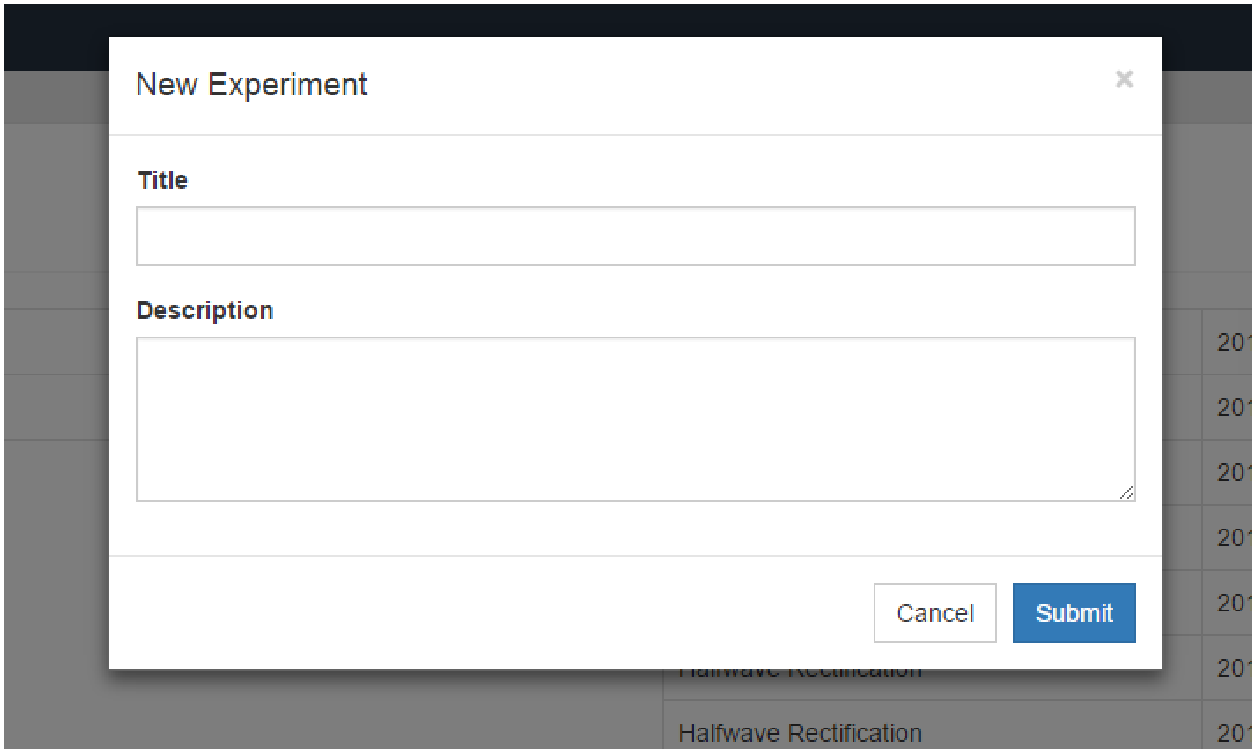


Figure 25: Screen for experiment creation

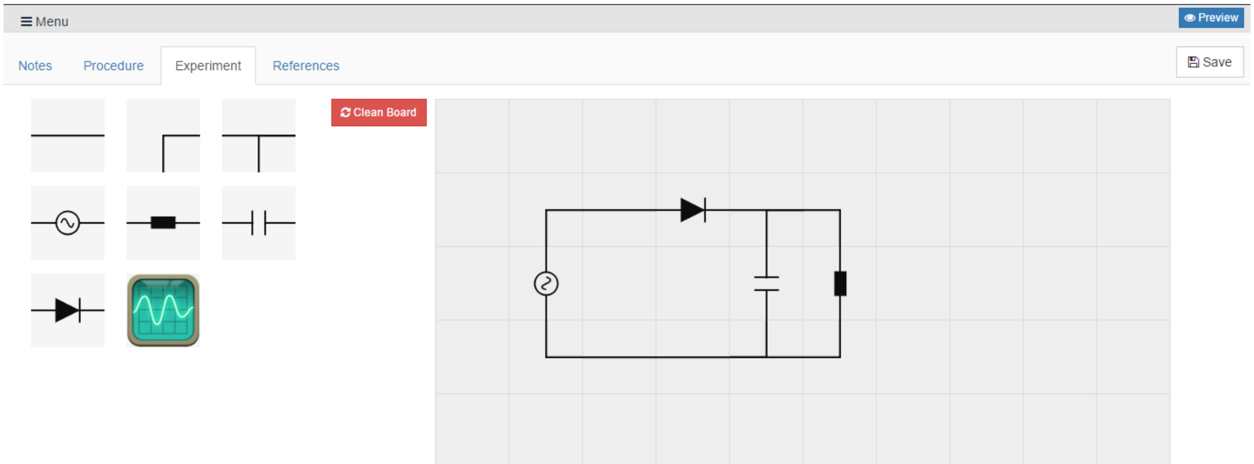


Figure 26: Screen for creation of the experiment schematic

In addition to designing the circuit schematic, the experiment creator(s) need(s) to save the circuit configurations for each of the components on the circuit schematic. These include values of resistors, capacitors, power sources, and many other values that may need to be set.

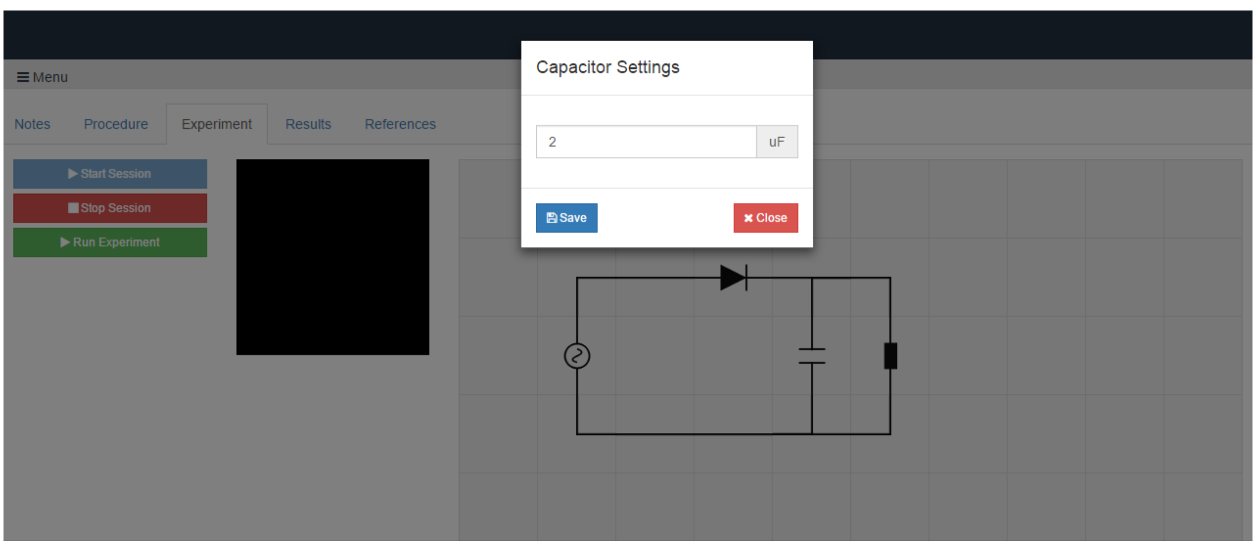


Figure 27: Screen for setting the capacitor configurations

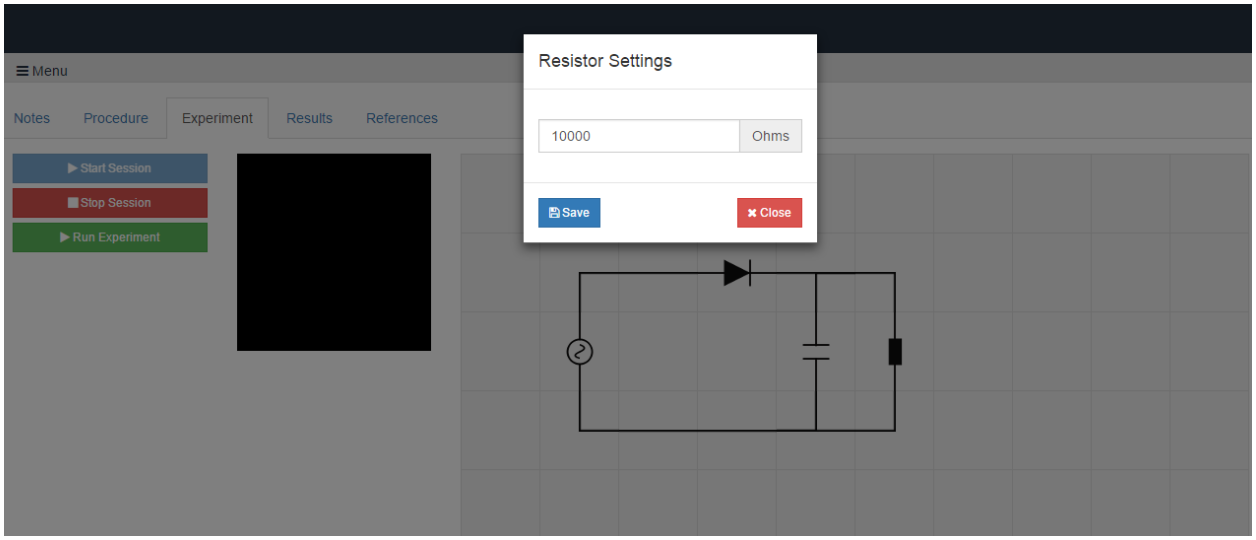


Figure 28: Screen for setting the resistor configurations

Users that have successfully logged into the system can access and run any of experiments that are available (including those that have been created by their respective accounts). When user accesses an available experiment, information about the background, procedure and creator(s) of the experiment is availed alongside the schematic of the experiment circuit.

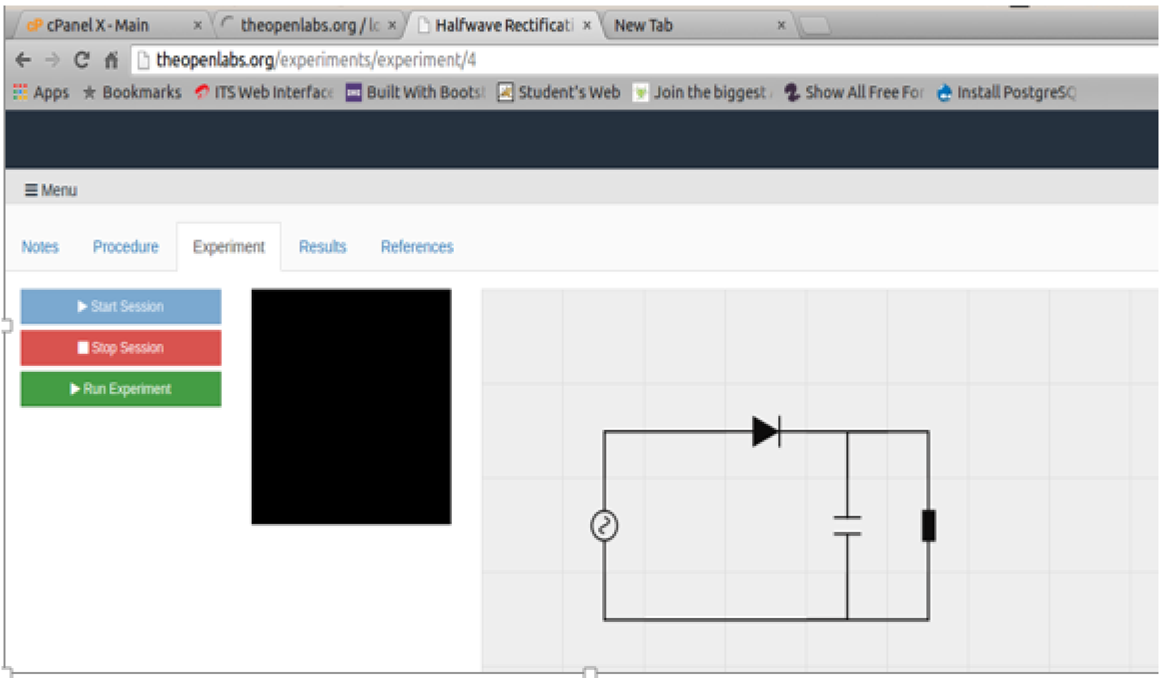


Figure 29: Screen for the schematic user interface

When user runs and experiment the results of that session are stored and can be accessed for future reference

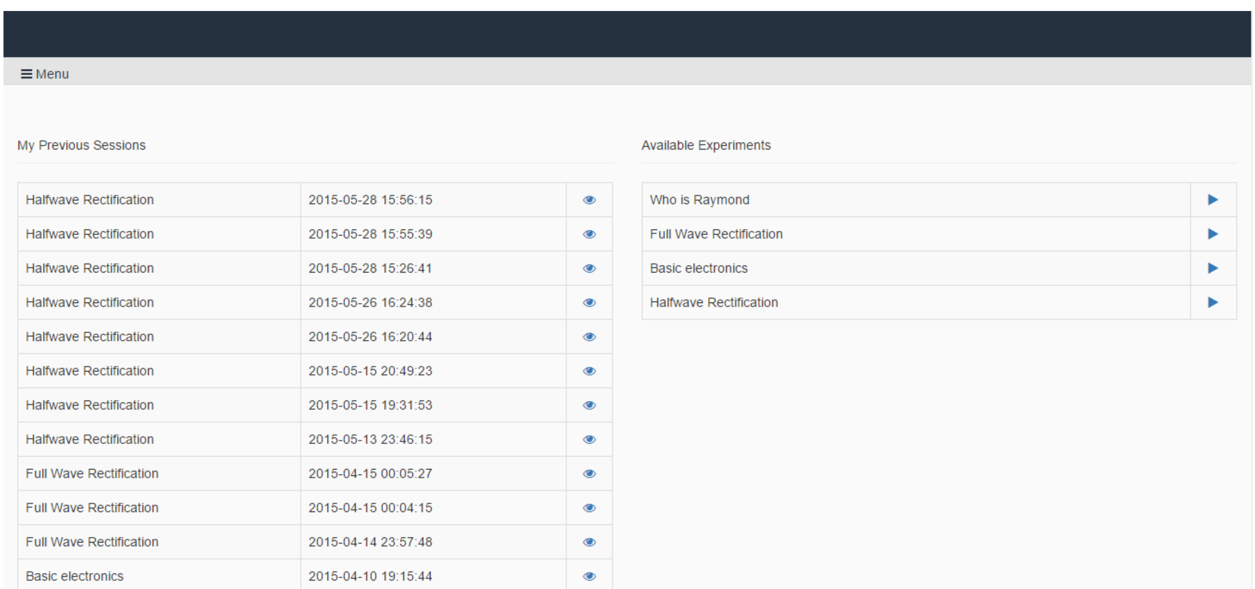


Figure 30: Screen for landing page displaying previous user sessions

- Application Programming Interface (API): This created the connection between the laboratory server and website application server. This enabled exchange of data between those two servers at any one moment when a user tried accessing an experiment.
  It consisted of PHP functions that could create, read, update and delete records from the webserver database.

3.5.2 Data Acquisition Module

The data acquisition module consisted mainly of Arduino which is an open-source electronics prototyping platform [5] as well as a python SDK. Precisely, an Arduino board is a microcontroller board that was programmed to sample values of the rectifier input and output signals for a given period of time when the experiment is being accessed by a user on the website server end, and print them to the serial port of the laboratory server. Data to be acquired was mainly voltage values.

The python Software Development Kit was designed to notify the Arduino on the laboratory server of an active session for that particular experiment, hence giving instructions to start sampling. This happened for a certain period of time, and once all the results were printed to the serial monitor, the python program was designed to collect them and send them to the openlabs website application to be availed to the user who initiated the session.

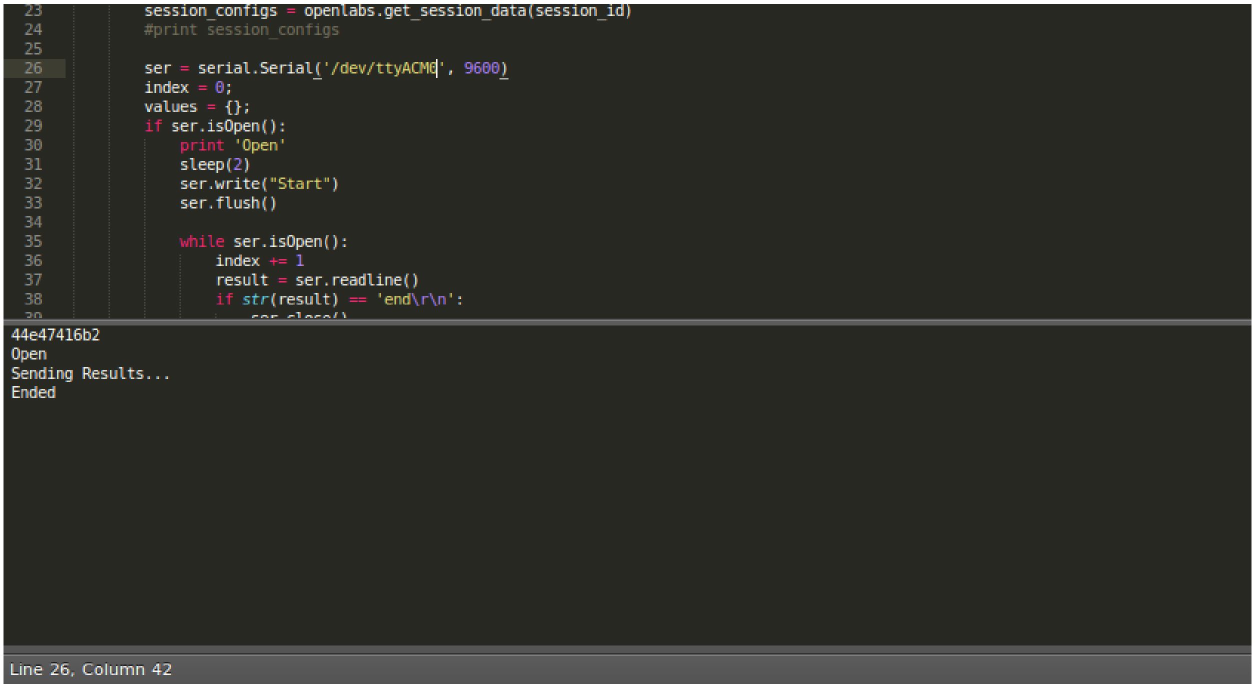


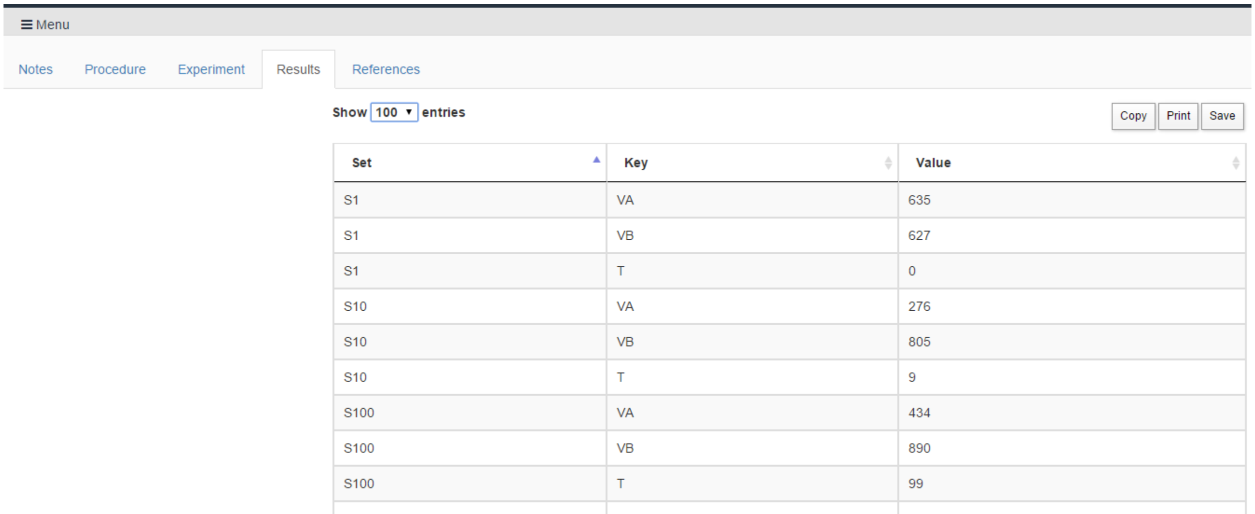
Figure 31: Python Program running on the laboratory server

It should be noted that for an experiment to be accessible to other users on this platform, the laboratory server needed to be connected to reliable internet, and secondly, the laboratory server must have the python SDK running to notify the Arduino when an experiment session starts.

It should also be noted that any other microcontroller other than the Arduino can be used to sample the two signals.

3.5.3 Displaying Results

Results were displayed in the form of a table having sets of values for input signal, output signal, and time.



Set	Key	Value
S1	VA	635
S1	VB	627
S1	T	0
S10	VA	276
S10	VB	805
S10	T	9
S100	VA	434
S100	VB	890
S100	T	99

Figure 32: Screen for results

These results are specific to a user account as well as a session. They can be downloaded in form of a CSV file for further analysis.

Chapter Four: CONCLUSION, CHALLENGES AND RECOMMENDATIONS

4.1 Summary of Results

The research yielded a fully developed website application to support the open source internet laboratory framework, and a lab server program written in python to act as an interface between the actual experiment hardware and its representation on the website application. The test laboratory experiment that was deployed is a half-wave rectification experiment. It utilizes power from the mains or sockets to generate an input signal. This is as opposed to using the functional generator instrument of the NI ELVIS board.

The test iLab deployed on this framework will support diode characteristics experimental requirements in various courses at CEDAT such as Electronics, and Physical Electronics that are offered to the Bachelor of Science in Electrical, Telecommunication and Computer Engineering programs at the College.

4.2 Research Contribution

The major objective of this research was to design, construct and implement an open source internet laboratory platform; this was successfully accomplished. The developed Lab will be mainly utilized by students offering courses with concepts in Diode characteristics and wave rectification at the College of Engineering, Design, Art and Technology. The lab will therefore help to support curricula and avail practical opportunities in the basic electronics course units in the different programs at the college.

The development of this open source platform, will make it affordable for students and lecturers to set up online laboratories that are not hardware or software specific. The platform still addresses the issue of lack of laboratory equipment which proves the fact that online experiments are cost effective just as the conventional ISA.

The open source nature of this framework also makes it possible for stakeholders in the engineering and technology fields to contribute to electronics projects, since all online experiments are accessible to all accounts registered on the openlabs application, hence promoting innovation.

The developed test experiment will reinforce the engineering knowledge of the students because it tests the knowledge accumulated from class and puts it to the test while carrying out the experiment.

The researchers have gained skills pertinent to their personal and professional development namely; website application development, project documentation, electronics, and bootstrapping among others. These provide the necessary tools as an Engineer to face various challenges.

4.3 Challenges and Limitations

4.3.1 Challenges

There are quite a few challenges faced in the research. The major challenges were the power cuts and unavailability of internet access that continually slowed down the progress of the research considering all the three core components of this architecture require to be fully connected to each other remotely.

4.3.2 Limitations

The sampling device used (the Arduino microcontroller) does not recognized negative values. This led to loss of data considering some of the data points where missed. The sampling device used also had limitations in the sampling rate, the data obtained was not sufficient enough to construct a perfect waveform of the results.

4.4 Recommendations

Despite the fact that this research proved the concept of an open source internet laboratory platform, issues of scheduling handling multiple users simultaneously where not addressed. Future research could be done to ensure the system stay up and running when a particular experiment on the platform is being accessed by more than one user at a time.

Future research could also cater for group access to the facility, where certain experiments are only accessible by a given group of accounts registered on this system. This will be very important for protection of company confidential information and secrets.

References

[1] O. Akinwale, A. . T. Asiimwe , M. Cosmas, T. T. Sandy-Stevens, M. I. Paul, B. Julius and K. Andrew, "Online Laboratories: Enhancing the Quality of Higher Education in Africa," 2011.

[2] A. S. Sedra and K. C. Smith, "Microelectronics Circuits," New York, Oxford University Press, 2004.

[3] "half wave rectirier," [Online]. Available: http://www.circuitstoday.com/half-wave-rectifiers. [Accessed 29 May 2015].

[4] J. A. d. A. R. L. H. B. C. K. D. C. F. J. H. H. I. J. I. P. D. L. T. M. N. J. N. M. S. V. JudsonHarward, "The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories," in *Proceedings of the IEEE* , 2008.

[5] 2014. [Online]. Available: http://arduino.cc/en/Main/arduinoBoardLeonardo.

[6] P. L. P. I. Evan Lindsay, "Workshop - Remote Laboratories: Approaches for the future.," *Frontiers in Education conference - Global engineering: Knowledge without borders - borders without passports*, pp. W1C-1 - W1C-2 , October 10 - 13, 2007.

[7] S. M. E. L. D. L. David Lowe, "Evolving Remote Laboratory Architectures to Leverage Emerging Internet Technologies.," *IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES*, pp. 289 - 294, October - December 2009.

[8] C. M. A. A. T. P. I. M. Sandy S. Tickodri-Togboa, *iLabs: Revolutionizing Teaching and Learning at Makerere University.*

[9] "Background History," [Online]. Available: http://cedat.mak.ac.ug/ilabs/about-us/. [Accessed 20 May 2015].

APPENDIX

Appendix I: Non-Functional System requirements

System quality attributes

These are the benchmarks that describe the system's intended behavior within the environment for which it was built. They are categorized under mainly two types; runtime and non-runtime attributes.

Runtime Qualities

1. **Performance**

   The execution of required functions in the laboratory experiment should be no more than 30 seconds given a stable and reliable internet connection. Lab Server System execution instructions for input/output operations should be given high processing priority.

2. **Functionality**

   The system should provide the required functions. These include; registration, and actual experimentation which involves experiment creation, access, configuration of parameters, and display of results.

3. **Availability**

   The system should always be functional and working; the length of time between failures and the length of time needed to resume operation and a failure should be very minimal, typically 40-90 seconds. Infrastructural problems, system loads, network faults, application failures should be quickly and easily resolved to increase the system availability.

4. **Security**

   The system should be able to resist unauthorized attempts at usage, disclosure or loss of information, or modification of information while still providing service to legitimate users. This is ensured by issuing usernames and passwords to registered and accepted users only.

5. **Interoperability**

   This is the ability of two or more systems to cooperate at runtime. Specific to this system, the lab server should exchange information with the openlabs website application server. The openlabs website application server will handle all services that involve user authentication, the Lab Server will have the experiment executed on instruction from the client on the website application and present results depending upon configuration.

6. **Reliability**

   The system should be able to perform the intended functions under stated conditions in the expected time period for the experimentation. It should stay up running unless there is internet disconnection, cut off in power supply, inconsistent outputs that require instrumentation or may be the unlikely, system crashing.

7. **Usability**

   The system should provide generally good user-experience with the interfaces. The knowledge from the availed experiment documentation should be simple to comprehend in order to achieve the desired goal. The experiment steps should be clear and easy to follow for the user.

Non-runtime attributes

1. **Affordability**

   The services offered by this system should be afforded by the user for example, the user should be able to access the internet easily and possess a computer to carry out these experiments.

2. **Customizability**

   The system software should be highly tailored by the developers to avoid effortless changes in the code used. This enables the system to perform the required functions and deliver the intended goals to the end user.

3. **Extensibility/Upgradability**

   The system should be designed with considerations for further modifications for making a laboratory richer in experimentation with the available documentation and infrastructural developments. The system should also be designed with flexibility to accommodate other instruments such as raspberrypis.

4. **Maintainability**

   The system shall be code based; any faults should be identified by the output or message displays on the computer and should be rectified through the back bone programs. The website application should also be easy to maintain.

5. **Portability**

   Since the system is open source, it should be able to run on all the common operating system platforms like Windows, Linux, MacOSX to access the experiments.

6. **Recoverability**

   The user should be able to recover their account credentials in case of loss of username and/or password. The user's experiment data should also be recoverable during the session period.

7. **Standards Compliance**

   The system should use internationally recognized standards, prime of which are ISO/IEC 15288 standard: "System Engineering- System Life Cycle Processes" and ISO 12207 standard: "Software Engineering- Software Life Cycle Processes standards.