



INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

**Project Report on
LAB MONITOR**

**SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF**

**CS699:Software Lab (Computer Science and
Engineering)**

BY

Pranav Chaudhary

Roll No:193059004

Harshwardhan Thorwat

Roll No:193059006

Shailendra Kirtikar

Roll No:193059007

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**



INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Project Entitled

LAB MONITOR

Submitted by

Pranav Chaudhary

Roll No:193059004

Harshwardhan Thorwat

Roll No:193059006

Shailendra Kirtikar

Roll No:193059007

is a bonafide work carried out by students and it is submitted towards the partial fulfillment of the requirement of CS699:Software Lab (Computer Science and Engineering).

Prof. Kavi Arya
Department of Computer Science, IIT Bombay

Abstract

In every Institute and corporation there is a need of system administrators who will maintain the labs and computers. IIT Bombay also have system administrators maintaining various labs in the institute. Work of the administrators is very important in order to conduct practicals and exams.

One of the major challenges that system administrators face is they have to check individual systems manually for status of the system. This work is repetitive and highly time consuming. The system administrators waste lot of time for this. They will have to check for individual system for presence of hardware devices, whether the computer is connected to network, list of softwares installed in the system, etc. System administrators have to put lot of time for this work. It will be better if we can make this task simple.

In order to do the lab maintenance effectively and without wasting much time, we can make this work remotely. We can use android application for this work. This will make the work of system administrator easy and effective.

Contents

1	Problem Statement	6
2	Goal and Objective	6
3	Implementation	7
3.1	Code	7
3.2	Design	7
3.2.1	Server	7
3.2.2	Client	7
3.3	Experimental Results	8
3.4	Use Cases	10
4	User Documentation	12
4.1	Server	12
4.2	client	13
5	Future Usage	14
6	Conclusion	15

List of Figures

1	Design of System	7
2	Screenshot: Software Lab 1 status.	8
3	Screenshot: Software Lab 2 status.	8
4	Screenshot: Basement Lab status.	8
5	Screenshot: List of Labs.	8
6	Screenshot: Shut Down function.	9
7	Screenshot: Wake on lan function.	9
8	Screenshot: Keyboard function.	9
9	Screenshot: Exam Mode option.	9
10	Use Cases for client.	10
11	Use Cases for functionalities provided.	11

1 Problem Statement

The work of system administrators is very tedious as they have to check individual system for checking whether the system is connected to network, presence of hardware devices, list of installed softwares, etc. We will create an client server architecture which will be used by system administrators remotely using their android device to do most of their work simple.

2 Goal and Objective

- Remotely monitor which systems are connected to network and which are not.
- Currently the lab machines are just locked in idle time leading to huge wastage of electricity.
- Remotely shutdown and power on the systems.
- Finding missing hardware components like keyboard, mouse, etc.
- Finding list of installed software in a system.
- Disabling memory devices on usb port on enabling exam mode.
- Find out CPU utilization of a system.

3 Implementation

3.1 Code

The code is developed in Python and java with the help of libraries mainly Flask server and android studio.

3.2 Design

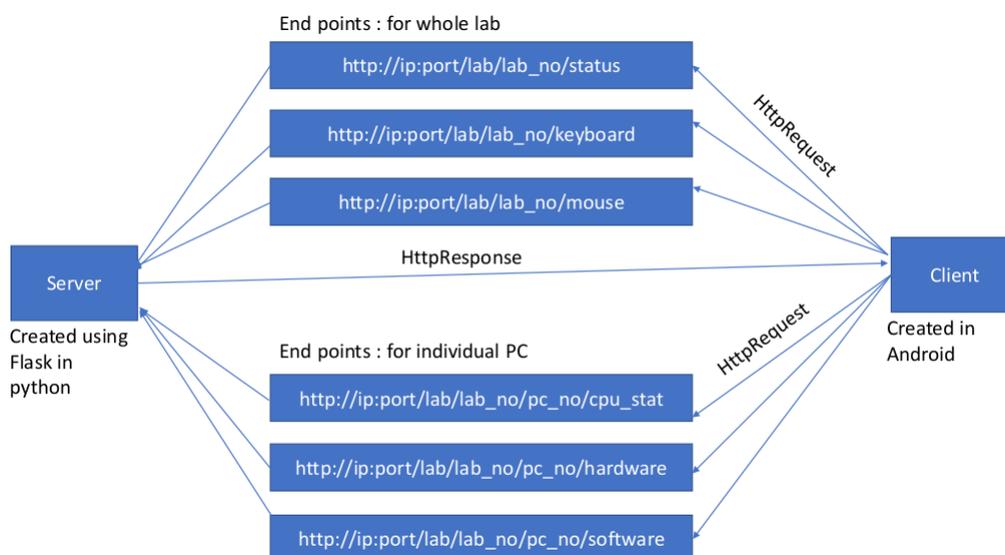


Figure 1: Design of System

3.2.1 Server

Server is implemented using Python and Flask server. The server will be in one of the lab system so that it can access all the systems present in the lab. Server will accept the requests made by remote client then it will perform all the necessary actions and then it will reply the client.

3.2.2 Client

Client is a android application implemented using Java and android studio. Client will take commands from the user and then it will send it to server and wait for server reply. Then it will give the output to the user.

3.3 Experimental Results

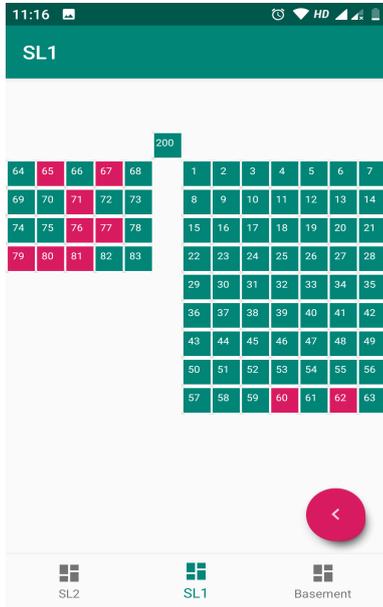


Figure 2: Screenshot: Software Lab 1 status.

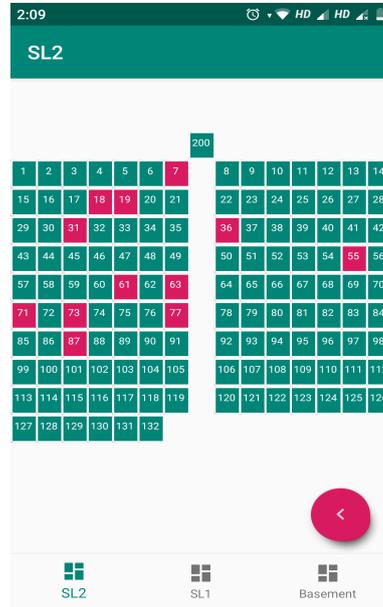


Figure 3: Screenshot: Software Lab 2 status.

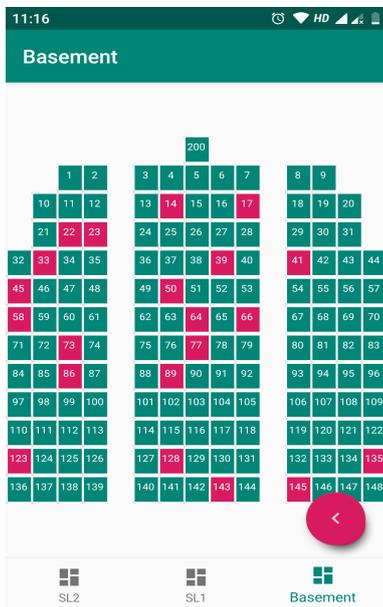


Figure 4: Screenshot: Basement Lab status.

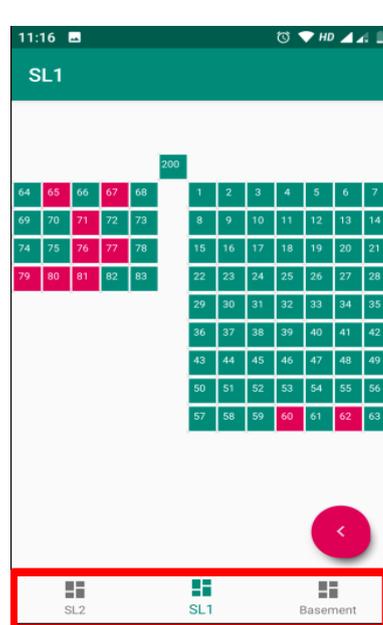


Figure 5: Screenshot: List of Labs.

The above set of images shows the results of the android client. This shows lab status of various labs.

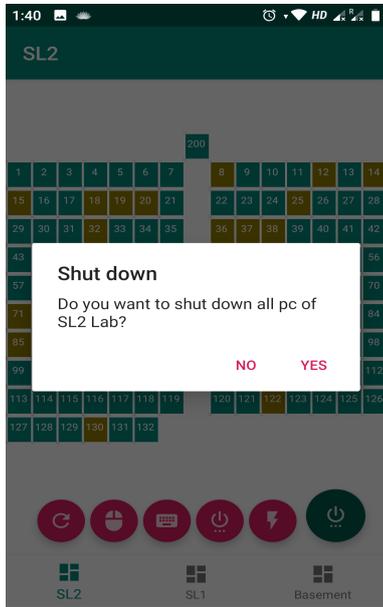


Figure 6: Screenshot: Shut Down function.

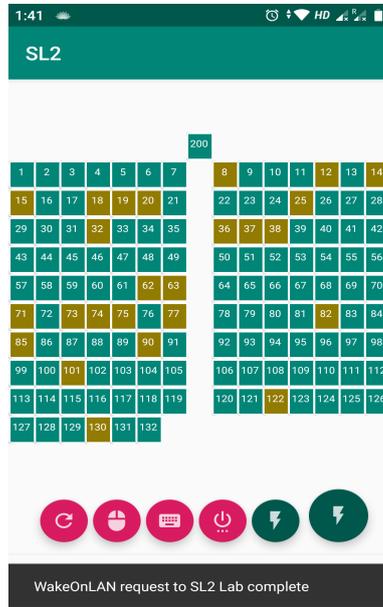


Figure 7: Screenshot: Wake on lan function.

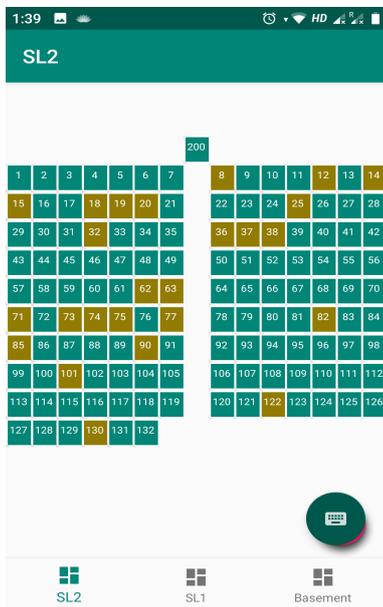


Figure 8: Screenshot: Keyboard function.

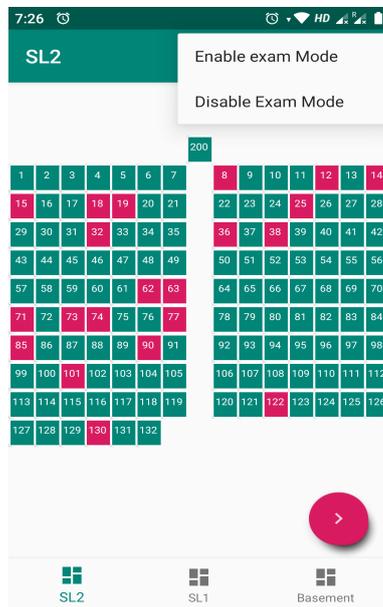


Figure 9: Screenshot: Exam Mode option.

3.4 Use Cases

The following image shows the use case description of the android client. This diagram gives detailed description of monitoring individual system in particular lab.

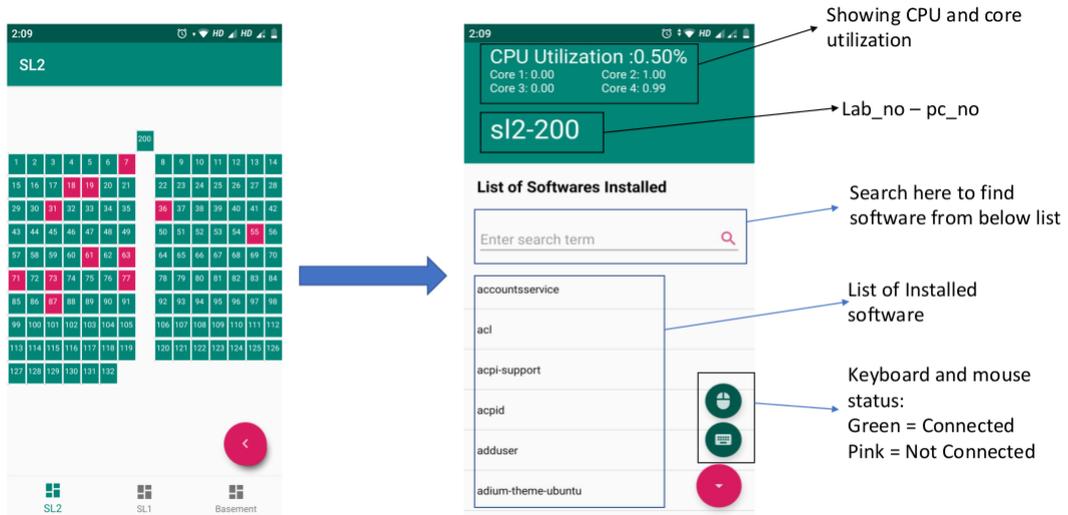


Figure 10: Use Cases for client.

The following use case diagram shows the functionalities provided by the Lab Monitor. These are listed at bottom of the page. All the functionalities can be performed simultaneously on the one lab at a time and not multiple labs at same time.

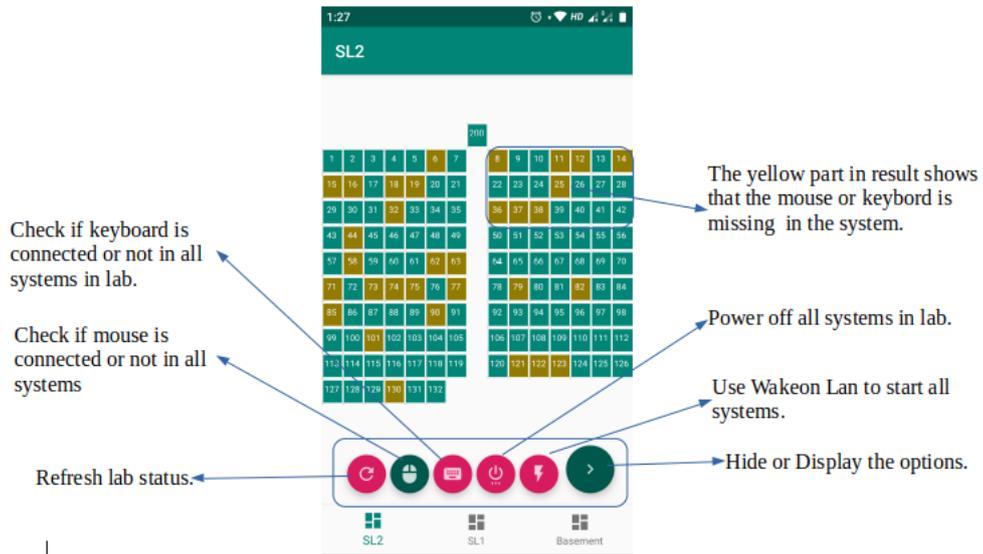


Figure 11: Use Cases for functionalities provided.

4 User Documentation

The source code for this project is available on GitHub
Clone this to your system using following command.

```
git clone https://github.com/prantostic/CS699.git
```

4.1 Server

1. For running the server following are the requirements. Install them first

- (a) python

```
sudo apt-get update  
sudo apt-get install python3.6
```

- (b) pip3

```
sudo apt install python3-pip
```

- (c) paramiko==2.6.0

```
pip3 install paramiko
```

- (d) pynacl==1.3.0

```
pip3 install pynacl
```

- (e) flask==1.1.1

```
pip3 install flask
```

2. To run server go to the directory containing the project CS699. Open terminal in that directory. Go to the folder containing server code.

```
cd CS699/source/server
```

3. Main program for the project is app.py. Run this program for starting the server.

```
python3 app.py
```

4. After the server is up it will ask for the system administrator credentials. Enter username and password. Then the server will be on and it will start listening to client.

4.2 client

1. Android application(.apk) file is provided in git directory. You can directly install it in your android device.
2. If you want to compile code in your system follow the steps below.
 - (a) Following are the requirements for compiling client.
 - i. Java
 - ii. Android Studio
 - (b) You have to import the client code in Android Studio
 - i. Open android studio.
 - ii. Go to File→New→Import Project
 - iii. Choose the folder CS699/source/client/LabStatus
 - iv. Click Next→Finish
 - (c) Android Studio will build the project for you.
 - (d) You can now run the client project in Android Studio.

5 Future Usage

The Lab Monitor project is very useful in future as the further batches of system administrators will make use of it and it will be continued. Lab Monitor will help system administrators in reducing the physical efforts.

Everything in our project is working in very good way but we will look forward to add more functionalities.

6 Conclusion

We have built the client server model using which the system administrator will be able to perform their task effectively and fast. This app can provide details like is system connected to network, list of softwares installed, list of connected hardware devices, etc. User will also be able to shut down the systems as well as start the system remotely.