ASHESI UNIVERSITY COLLEGE

VIRTUAL LAB: A VIRTUAL MACHINE MANAGEMENT SYSTEM

by

MOHAMED KESSIR ADJAHO

Dissertation submitted to the Department of Computer Science,

Ashesi University College

In partial fulfilment of Science degree in Computer Science

APRIL 2014

Applied project

Declaration

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate’s Signature:………………………………………………………………

Candidate’s Name:………………………………………………

Date:…………………………..

I hereby declare that the preparation and presentation of the dissertation were supervised in accordance with the guidelines on supervision of dissertation laid down by Ashesi University College.

Supervisor’s Signature:………………………………………………………………..

Supervisor’s Name:…………………………………………….

Date:…………………………………..

Acknowledgements

I thank my supervisor, Mr. Aelaf Dafla for his guidance throughout this project.

My gratitude also goes to my parents for their support during the last four years.

Abstract

Virtual Lab is a virtual machine management system that seeks to streamline the process of creating, editing and deploying virtual machines onto a set of computers. Many universities already use virtualization to enable their computer science students to do practical work. In some of these universities, the deployment and management of virtual machines is done manually. Virtual Lab automates this process so that faculty can focus on more useful tasks.

Table of Contents

Chapter 1: INTRODUCTION

1.1 INTRODUCTION & BACKGROUND TO THE PROJECT

It is important for any university offering a Computer Science degree to allow its student to practice what they are taught. This practice usually takes the form of installing, configuring and using software. It cannot be done without disrupting the school's computer configuration and settings. Moreover, students may need to use software that is not available in their computer's operating system.

The proposed solution is the use of virtual machines. On virtual machines, students can install, configure software without restrictions or the fear of messing things up. They can create complex configurations that require more than one computer. If an operating system is corrupted or damaged, the virtual machine can be easily reinstalled or replaced. Virtual machines have worked well in that regard.

However, the process of managing virtual machines for the various courses is tedious. There is no easy way to deploy a virtual machine created for a course. Lecturers and faculty interns have to create a virtual machine, copy it on a flash drive, copy and configure it for every single computer. This process has to be repeated for every single change that is made to a virtual machine. Moreover, lecturers and faculty interns have to manually edit Virtualbox configuration files to make sure the virtual machine works. This may work with a few computers but it is not a scalable solution.

In addition to the management of the virtual machines being tedious, there is no way to manage access to the deployed virtual machines. At the moment, every existing virtual machine is available to anyone who can sign into the computer. Therefore students have access to virtual machines they do not need to know about.

The use of virtual machine effectively allows students to do more practical work and gain experience. However, the process of managing the virtual machines could be streamlined and made easier.

1.2 Objectives

The objective of this project is to develop a virtual machine management system that will make the life of the computer science faculty and students better. It would allow faculty to create virtual machines for different purposes and easily deploy them onto a number of computers. Changes to these virtual machines can also be easily propagated.

This management system will also allow faculty to specify which users have access to which virtual machine.

1.3 MOTIVATION

“I hear and I forget. I see and I remember. I do and I understand”. This quote from the Chinese philosopher Confucius reminds us of the importance of practical work. Practical work is even more important in a STEM-related field such as computer science [1]. A university cannot produce good computer scientist or engineers if it does not provide them with the tools to practice the theory they learn. The use of virtualization has proven to be helpful in allowing students to practice what they are taught. This project is interesting to me because it is helping to solve an immediate problem.

In addition to adding value to an existing solution, this applied project is an opportunity for me to work with new technologies such as .NET Windows Presentation Foundation and the VirtualBox developer API.

Chapter 2: DESIGN

2.1 ASSUMPTIONS

The design of this virtual machine management system is based on the following assumptions:

- The institution has a computer laboratory with more than one computer.
- These computers are networked and they are accessible over the network from one another. This makes deployment from one computer possible.
- The computers are running Microsoft Windows 7 or Windows 8 operating system. This is because Virtual Lab is written using C# on the .NET development platform.

2.2 FUNCTIONAL REQUIREMENT

- The system must allow faculty to create/import virtual machines
- The system must allow faculty to make changes to existing virtual machines
- The system must allow faculty to deploy virtual machines to a set of computers on the network
- The system must allow faculty to redeploy edited virtual machines to a set of computers
- The system must allow faculty to specify which group of students have access to a certain virtual machine
- The system must allow faculty to add computers to the Virtual Lab network
- The system must allow students to login and have access to a number of virtual machines

2.3 Nonfunctional Requirement

- The system should not disrupt the institution's network or computer configuration
- The system should not create a loophole in the institution's security

2.4 Use case diagram

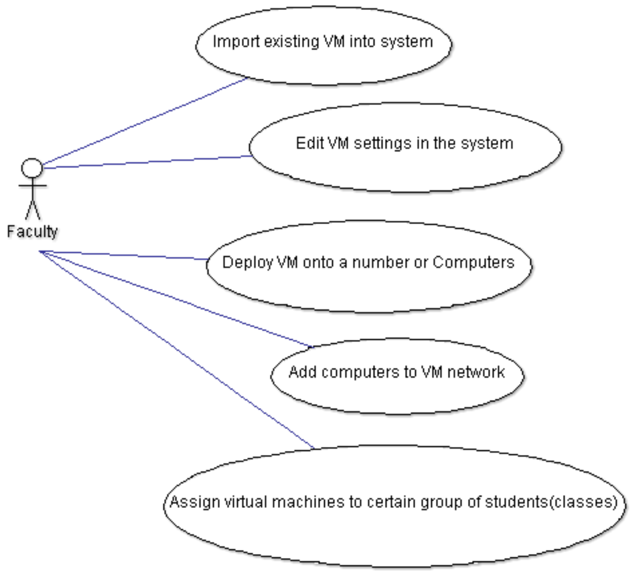

Import existing VM into system
Edit VM settings in the system
Faculty
Deploy VM onto a number or Computers
Add computers to VM network
Assign virtual machines to certain group of students(classes)
Login and have access to VMs
Lauch available VMs
Student

Figure 1: Use case diagram

2.5 USE CASE SCENARIOS

Mr. John Khan is a lecturer at Ashesi University. He figured that he could use virtual machines to allow his students to experiment with new software without disrupting the operating system on the school's computers.

Mr. Khan has been using the virtual machine solution for a while, it is very convenient. He usually creates a virtual machine for a particular class and copies it onto each of the school's computers. He then has to edit some xml files on each computer to make sure the virtual machine works correctly.

With this virtual machine management system however, all Mr. Khan has to do is to create a virtual machine, click a button to deploy it to all the computers in the Virtual Lab's network.

Fareed Johnson is a student at Ashesi University. As part of his Mobile Web coursework, he needs to do Windows phone development. However, Fareed uses an Apple Mac Book Pro so he cannot develop Windows phone applications on his computer. Conscious of this issue, his Mobile Web lecturer has created, configured and deployed a Windows 8 virtual machine onto the computers in the computer laboratory. Fareed can then log into one of those computers and launch that Windows 8 virtual machine. Because Fareed is taking only Mobile Web this semester, he only has access to the Virtual Machines pertaining to his Mobile Web class and does not have to browse through the plethora of available virtual machines.

2.6 System Architecture

Virtual Lab is comprised of a server application and a client application. The server application is only available to faculty. Faculty can use it to create, manage and deploy virtual machines. The server application will be running on a single computer.

The client application on the other hand is the system's interface with the majority of the users, the students. On the client application, students can login and launch available virtual machines. The client application will be running on all the computers that students can use.

Virtual Lab has a **repository architecture**. Besides the transfer of files between computers, there is no direct communication between the server application and the client application. They are rather connected to a central database (Figure 2).

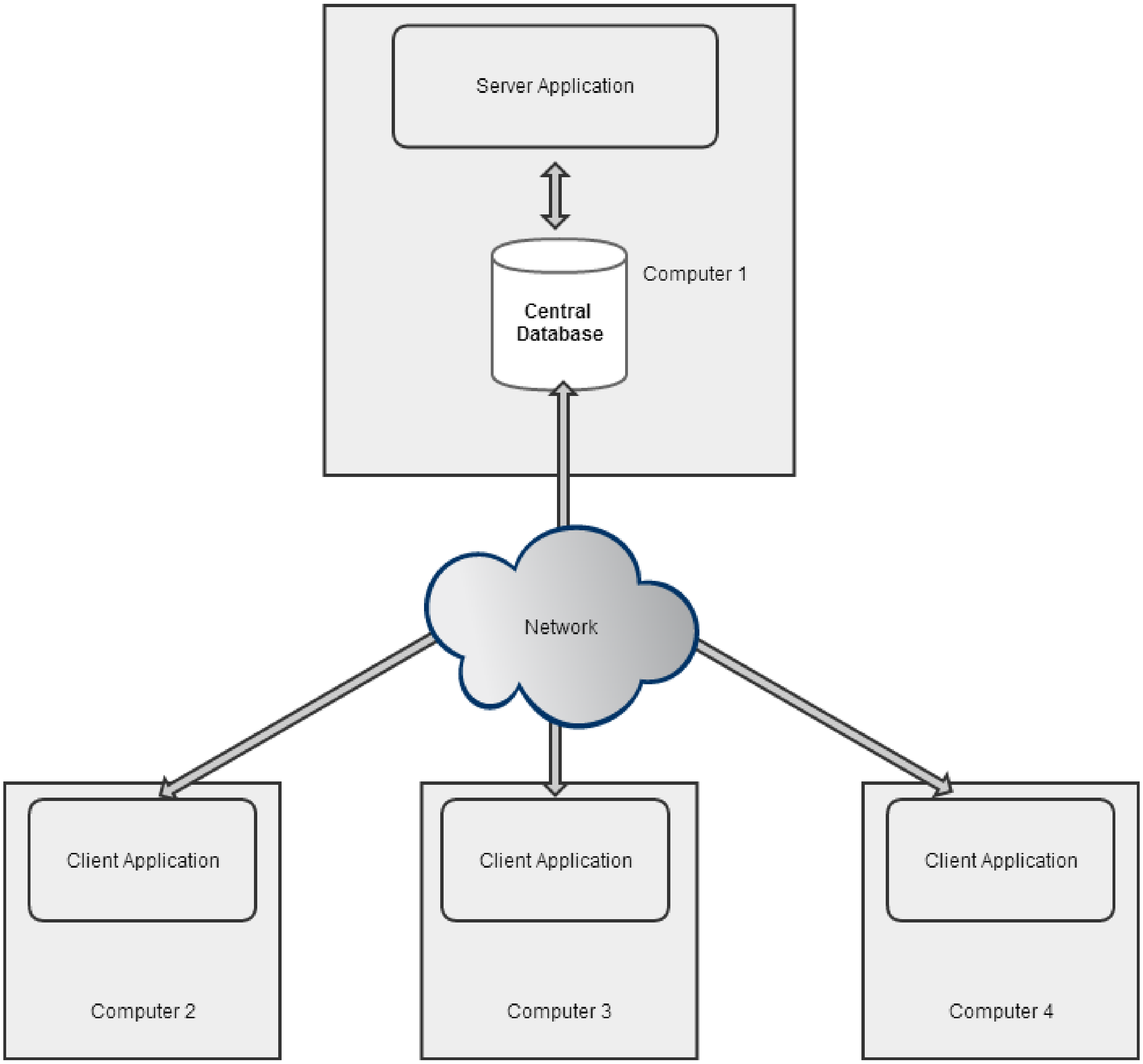


Figure 2: Architectural Diagram

2.7 DATABASE DESIGN

The system uses one central database to store all its information. The server application may be on the same computer as the database or not. The multiple clients will communicate with the database over the TCP/IP protocol.

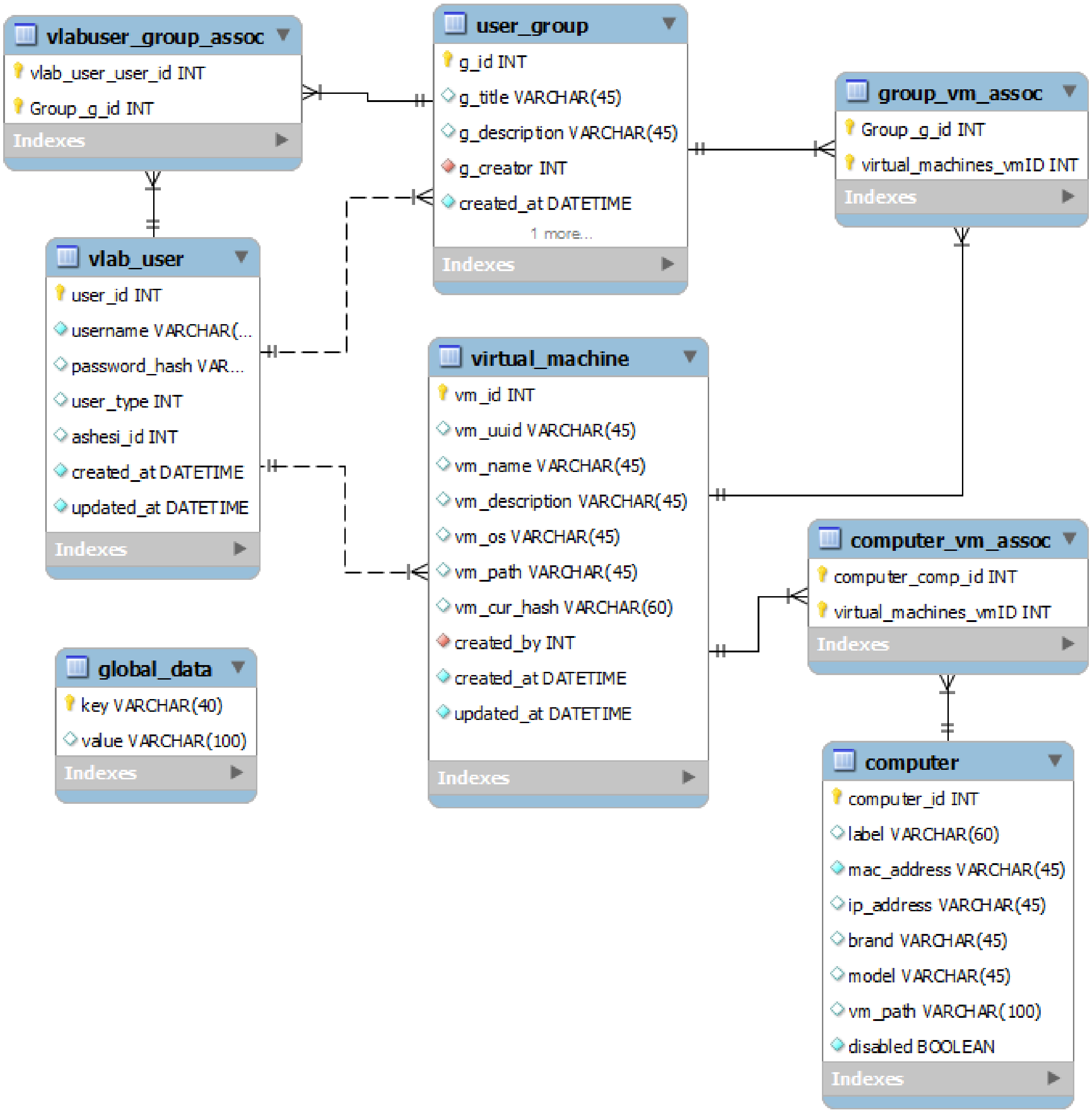


Figure 3: Database design

virtual_machine: This table represents a virtual machine that is created by a faculty and available for students to use. A virtual machine is identified by its unique UUID.

computer: The computer table represents physical computers that are in a university's computer laboratory. These are the computers onto which virtual machine can be deployed. The MAC address of the physical computer is used as a unique identifier.

vlab_user: This table represents all the users of the Virtual Lab system. The user_type attribute determines the type of user: A super user is identified by the integer 0, Faculties are identified by the integer 1 and Students are identified by the integer 2.

user_group: This table represents a group of users. This could be the set of users taking a particular course. A faculty can attach a virtual machine to a user group so that students in that group have access to the virtual machine.

global_data: This table will hold all global data needed by the system.

vlabuser_group_assoc: Association table between the Vlab_User table and the User_Group table.

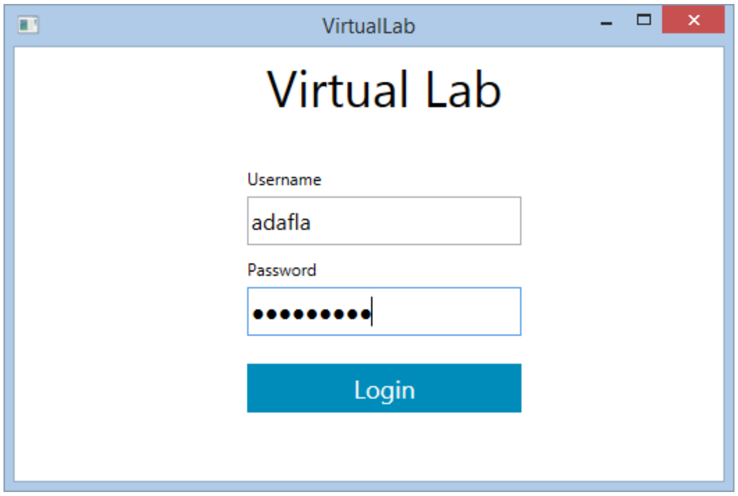
group_vm_assoc: Association table between the User_Group table and the Virtual_Machine table.

computer_vm_assoc: Association table between the Computer table and the Virtual_Machine table.

2.8 User Interface

2.8.1 Server Application

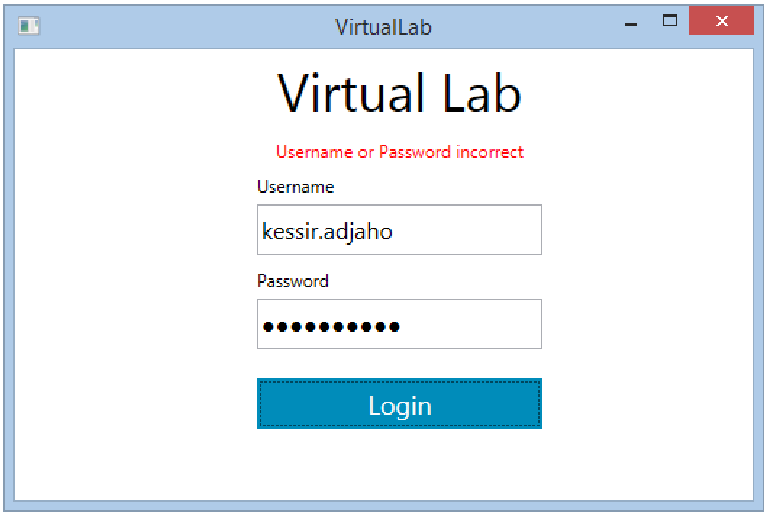
2.8.1.1 Login Screen



Username
adafla

Password
●●●●●●●●●

Figure 4: Login Screen



Username or Password incorrect

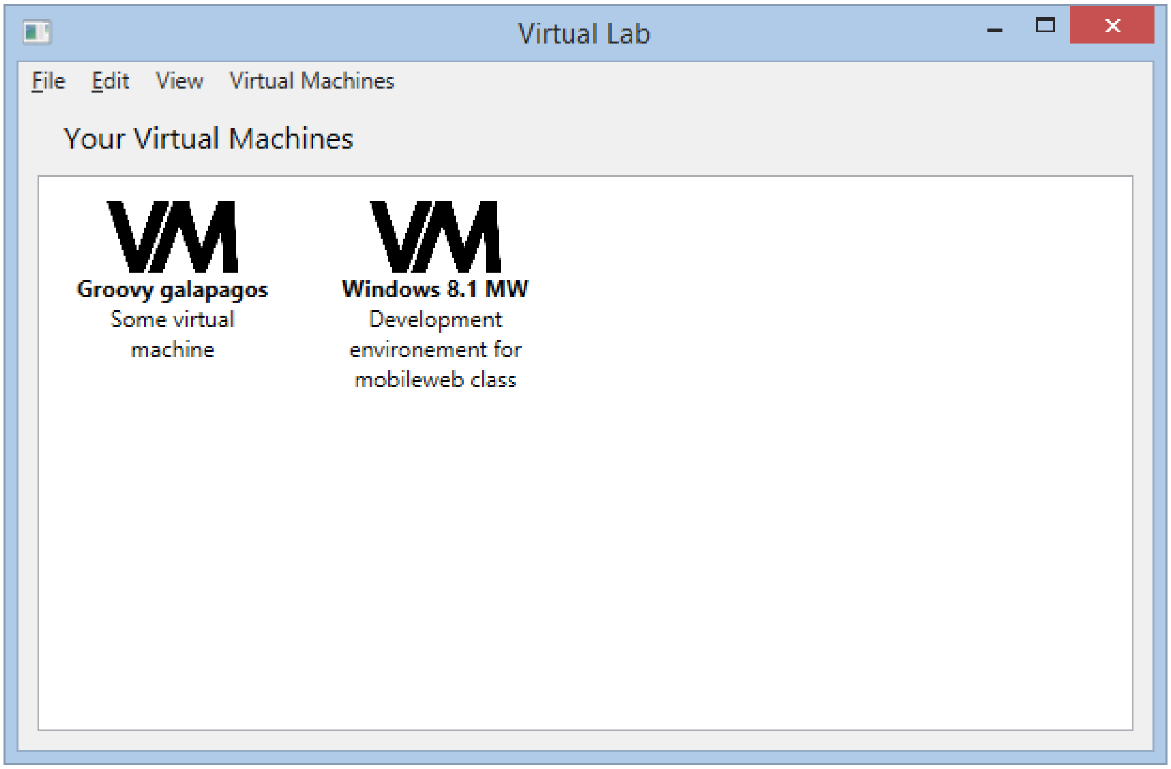
Username
kessir.adjaho

Password
●●●●●●●●●●

Figure 5: Login Screen, Auth. Failed

Users log in by providing their username and password. An error message is displayed if either the username or the password is incorrect.

2.8.1.2 Faculty's virtual machines



Your Virtual Machines

Groovy galapagos
Some virtual machine

Windows 8.1 MW
Development environement for mobileweb class

Figure 6: Faculty's Virtual Machines

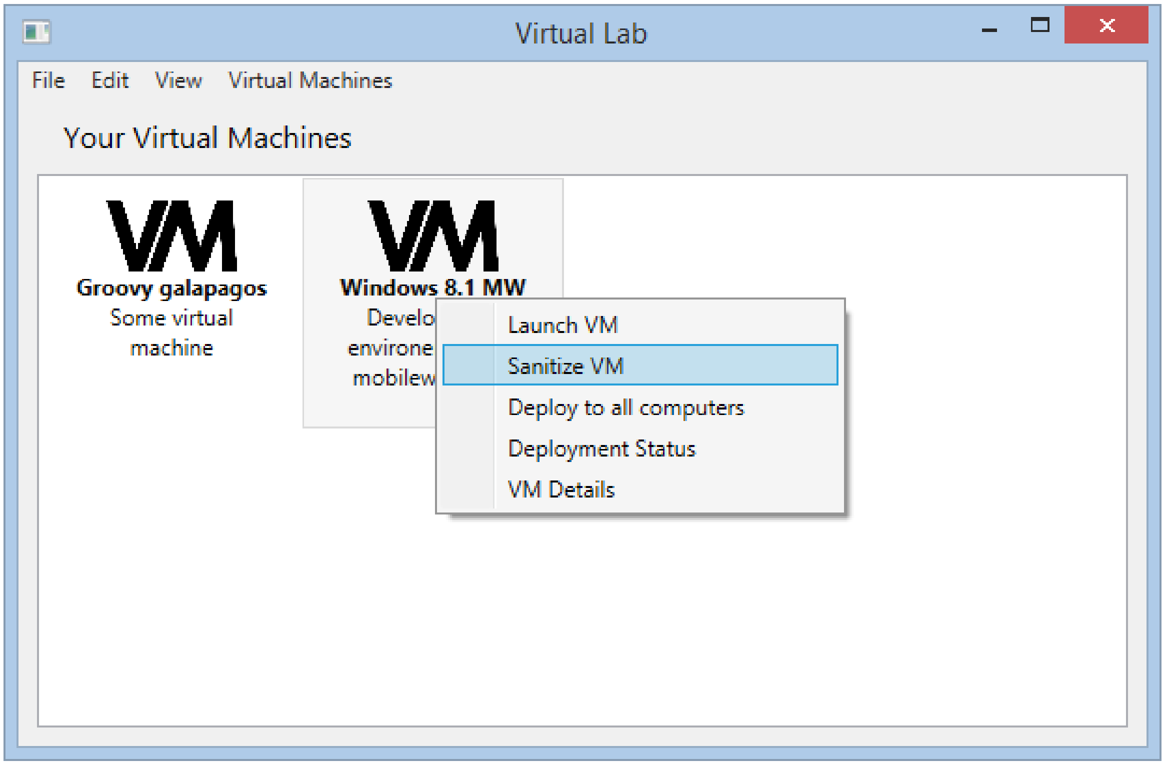


Figure 7: Faculty's Virtual Machines, Context Menu

After successfully logging in, the faculty is presented with a list of the virtual machines he has created. He can launch, modify and delete or deploy these virtual machines.

Double clicking on a virtual machine will launch it. To see the list of operations that can be performed on a specific virtual machine, the faculty can right-click on it (Figure 7).

2.8.1.3 Available computers in Virtual Lab

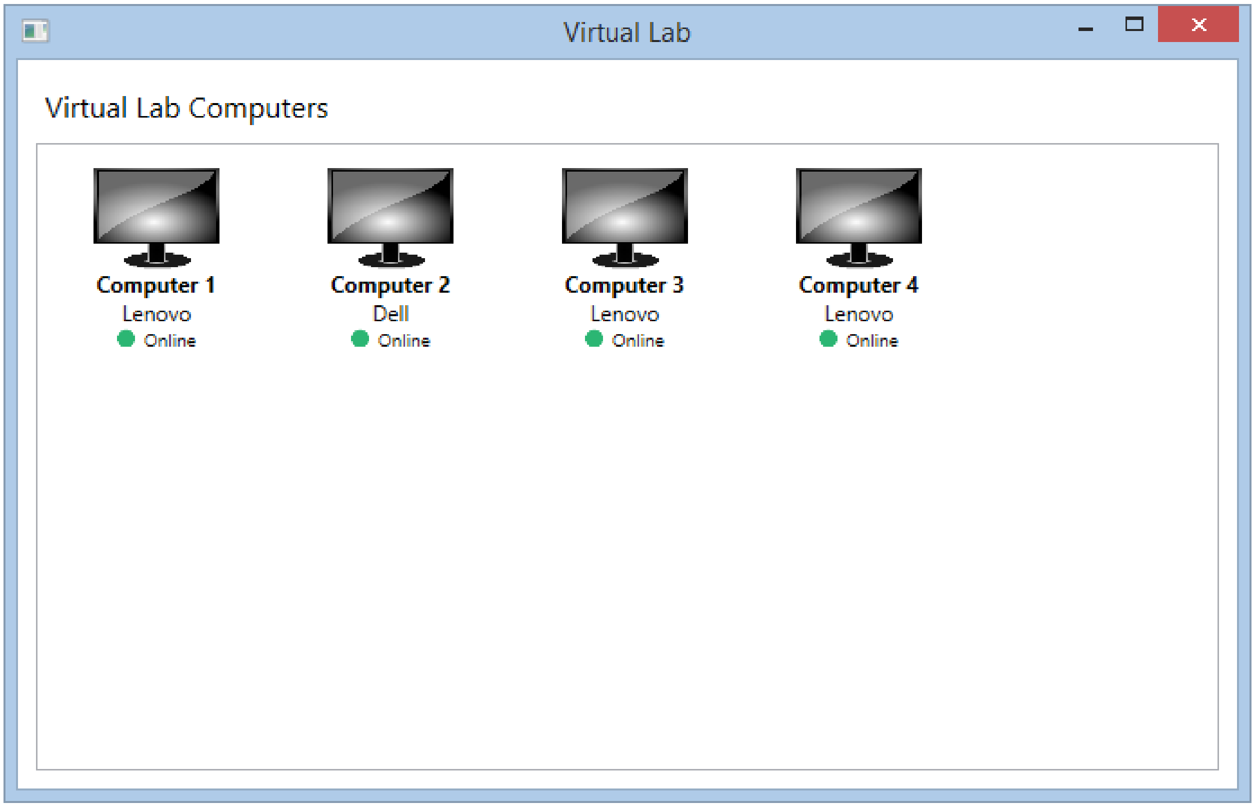


Figure 8: Available computers

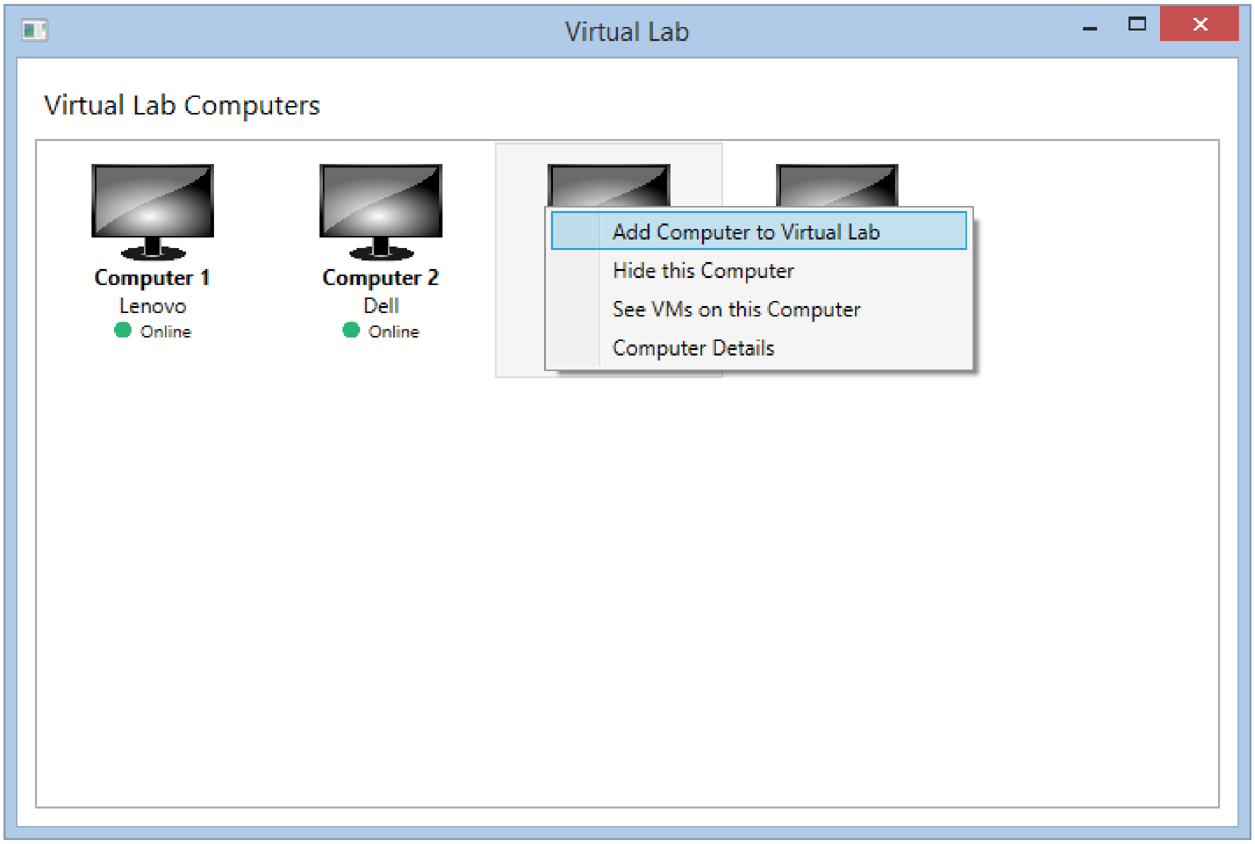


Figure 9: Available computers, context menu

On this screen, faculty can see which computers are online or offline at a particular moment. This screen accessed by clicking on ‘View’ in the menu bar, then clicking ‘All computers’.

By right-clicking on any of the computers, a context menu appears. From this context menu, a faculty can add a new computer to Virtual Lab’s network, hide the computer, see the list of VMs available locally on that computer, or view the computer’s details.

2.8.1.4 Faculty's user groups

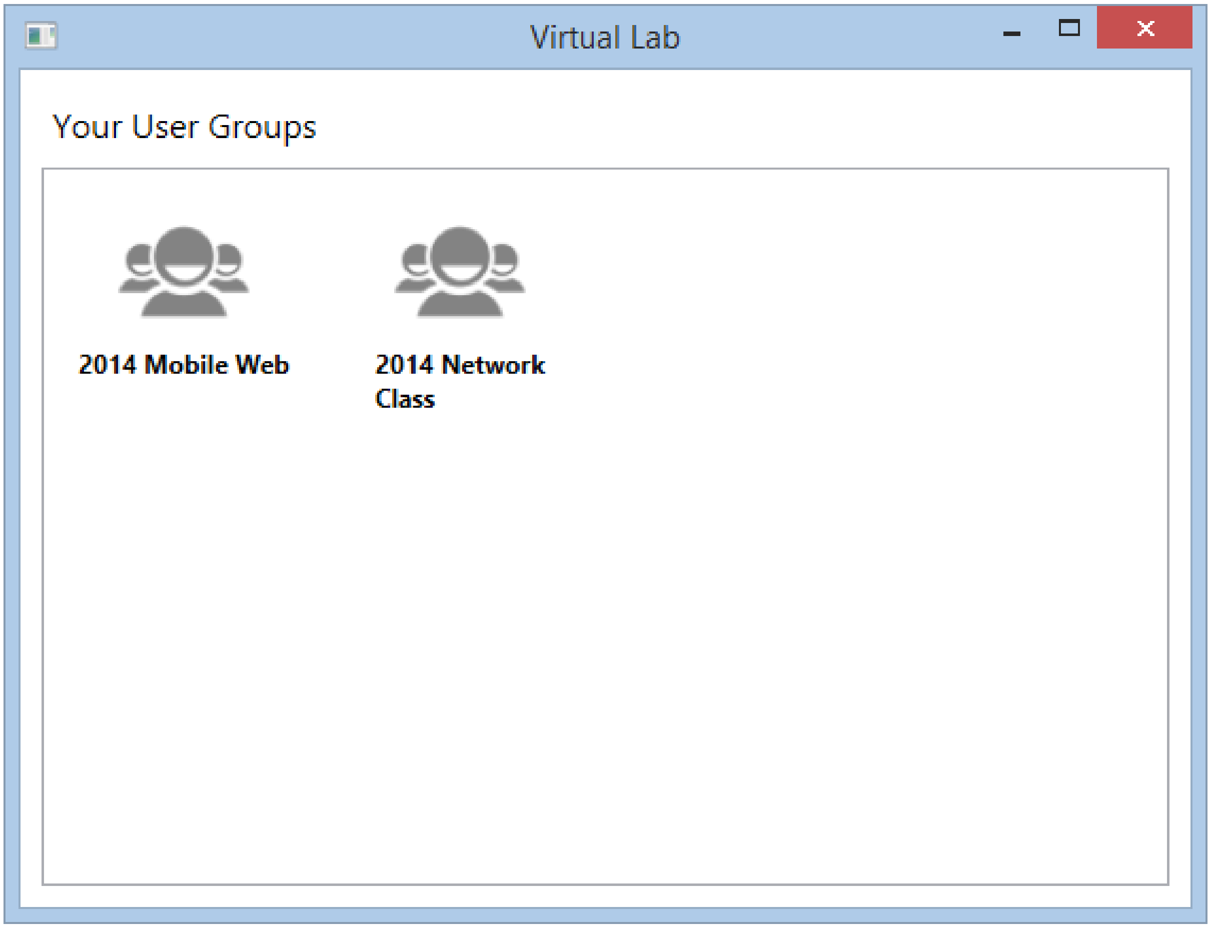


Figure 10: User Groups

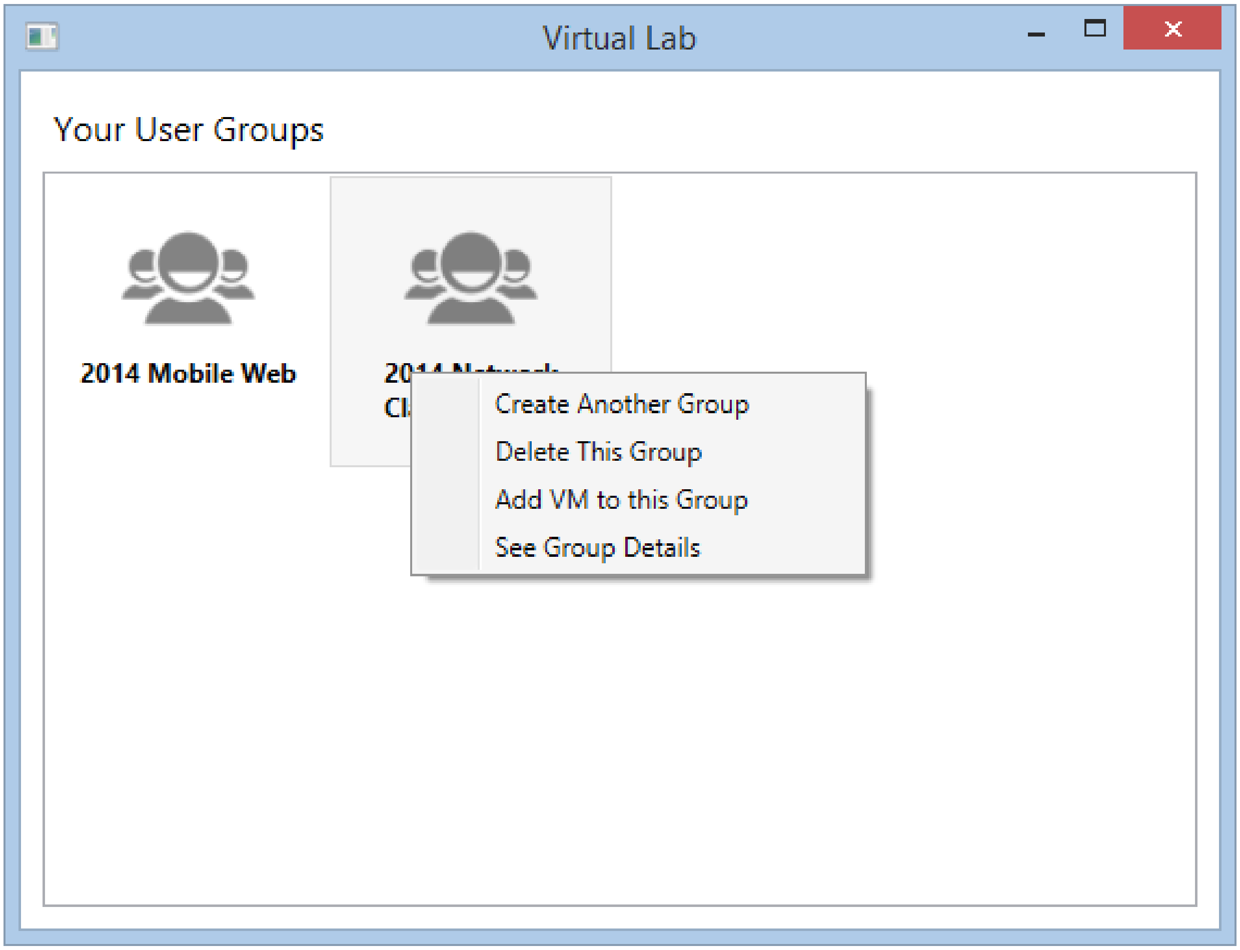


Figure 11: User Groups, Context menu

On this screen, faculty can see all the user groups (student groups) that he created. This screen accessed by clicking on 'View' in the menu bar, then clicking 'My User Groups'.

By right-clicking on any of the groups, a context menu appears. From this context menu, a faculty can create another group, delete the selected group, make a virtual machine available to the selected group or view details about the group.

2.8.1.5 Super user landing screen

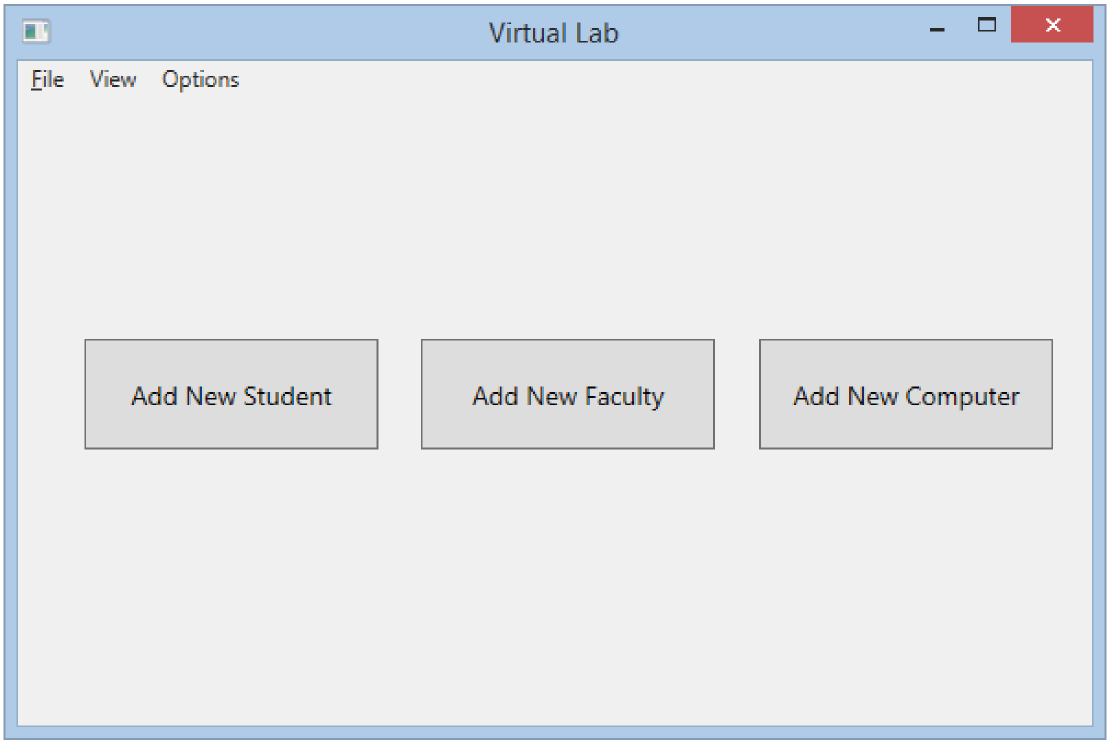


Figure 12: Super user landing screen

There is a super user that has administrative access to the whole system. He can add new students and faculty to the system. Figure 12 is the super user’s landing screen. The actions such as creating new user and adding new computers to the system are the most important and therefore are made more accessible.

2.8.1.6 New User

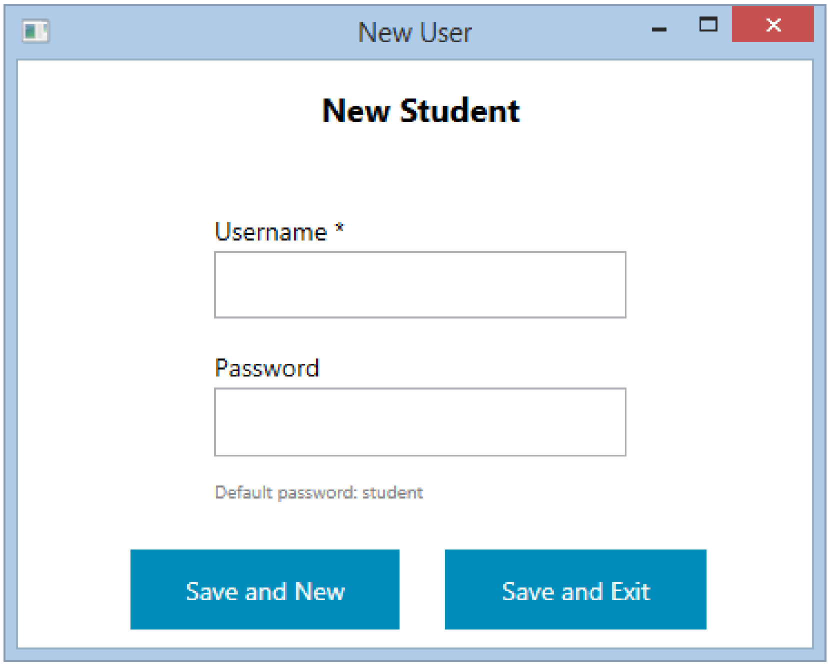


Figure 13: New User

The super user can create new users (Faculty and Student). A similar screen is used for creating a student and a faculty. If the super user does not provide a password, a default password is used. The user will then have to modify that password when he logs in the first time.

2.8.1.7 New computer

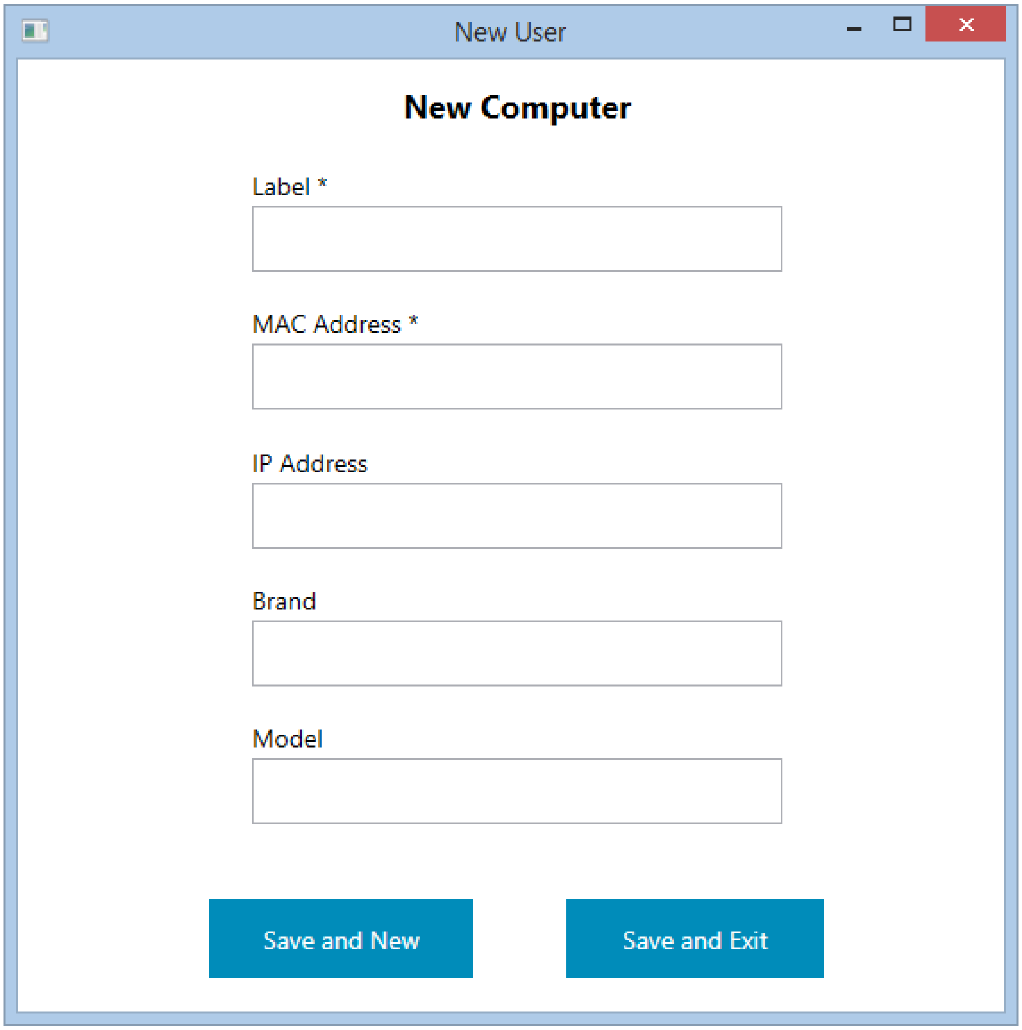
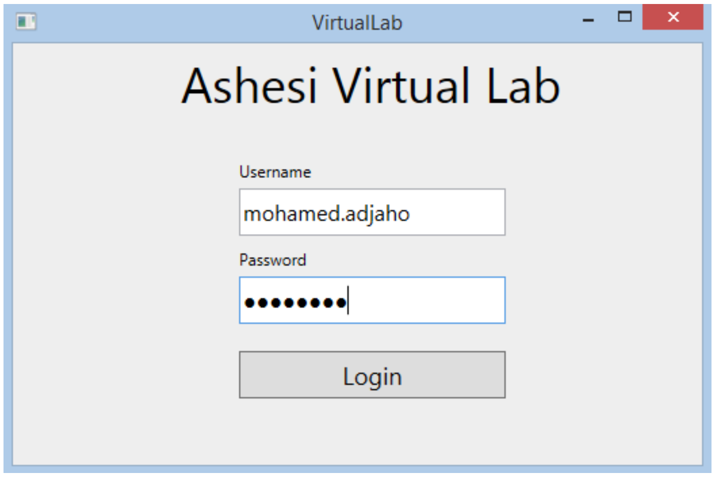


Figure 14: Adding New Computer

The super user is the only user allowed to add new computers to the Virtual Lab network. The computers in the Virtual Lab network are supposed to have a label by which they can be identified. A MAC address is also used as a unique identifier for the computers. The label and the MAC address are the only information required to add a new computer to the system.

2.8.2 Client Application

2.8.2.1 Login Screen

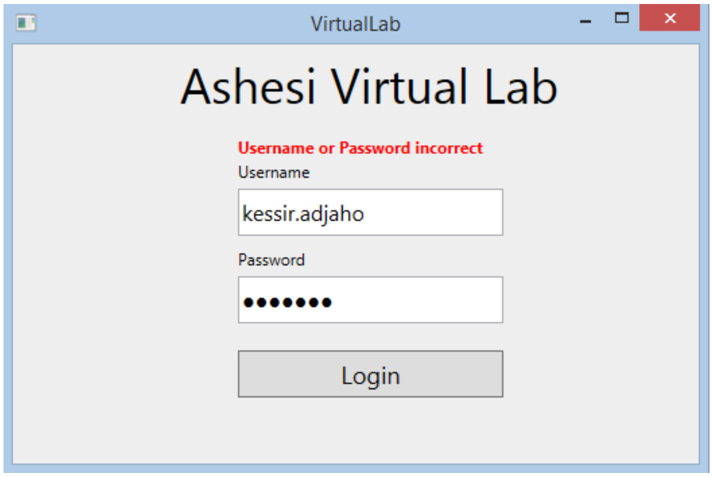


Ashesi Virtual Lab

Username: mohamed.adjaho

Password: ••••••••

Figure 15: Login screen



Ashesi Virtual Lab

Username or Password incorrect

Username: kessir.adjaho

Password: •••••••

Figure 16: Login Screen, Auth. failed

Users log in providing their username and password. An error message is displayed if either the username or the password is incorrect.

2.8.2.2 Available VMs screen

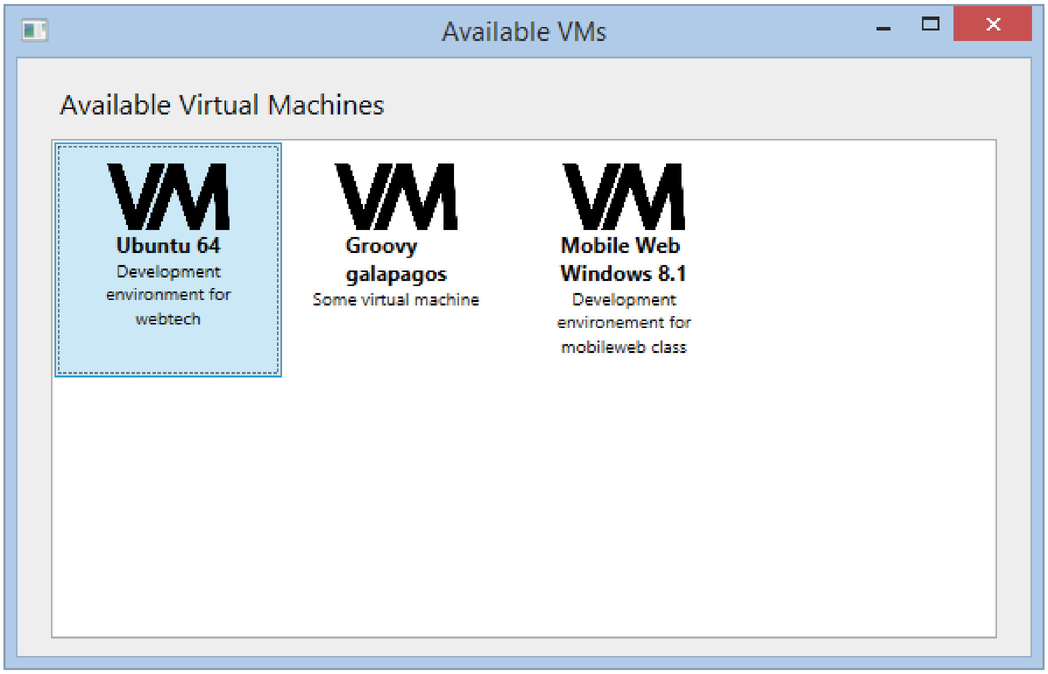

Available VMs
Available Virtual Machines
VM
Ubuntu 64
Development environment for webtech
VM
Groovy galapagos
Some virtual machine
VM
Mobile Web Windows 8.1
Development environement for mobileweb class

Figure 17: Available VMs

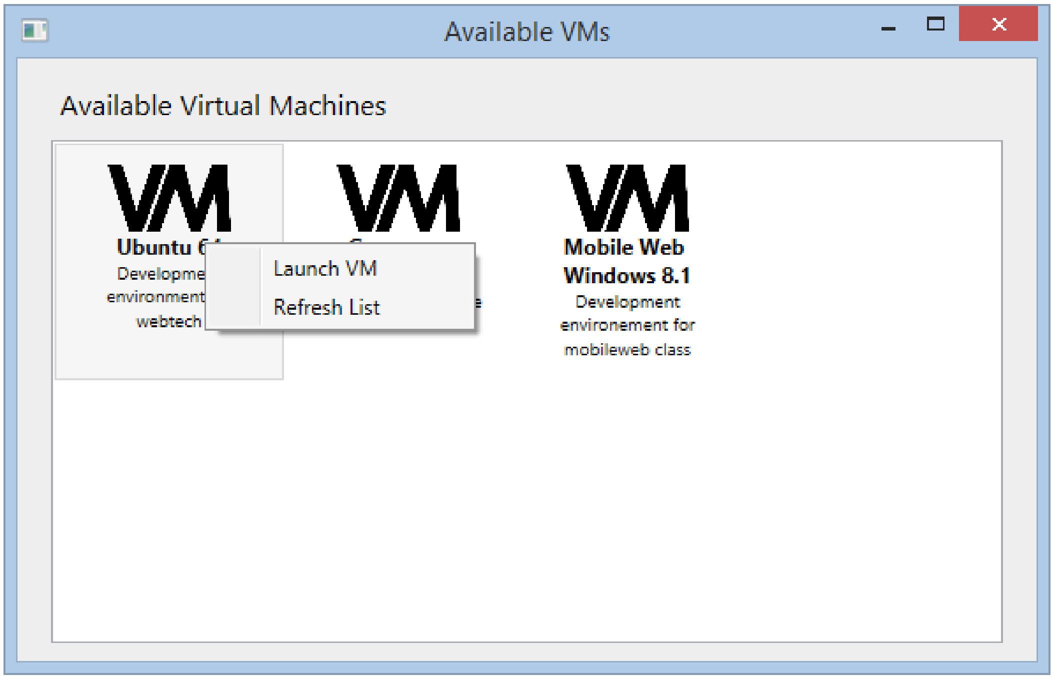


Figure 18: Available VMs context menu

Once logged in, the user (student) is presented with a list of available virtual machines. To launch a virtual machine, the user can double-click on it. The clickable area for a virtual machine is large enough to make double-clicking easy (Figure 17).

Alternatively, the user can right-click on the virtual machine to open a context menu and click ‘Launch VM’ (Figure 18). The context menu also allows them to refresh the list of virtual machines displayed by clicking on ‘Refresh List’. The ability to refresh the list is provided in case any change is made to the system after the user logged in.

2.9 DESIGN LIMITATIONS

It is important to note some of the limitations of the current design. If the central database is inaccessible for any reason, nobody will be able to use the system, even if the client computer has all the virtual files stored locally. This is because user authentication is performed against the central database. This situation is however preferable to the complexity of having a copy of the database on each computer and having to synchronize them.

Chapter 3: IMPLEMENTATION

3.1 TECHNOLOGY AND TOOLS USED

3.1.1 Virtual Box

The backbone of the Virtual Lab system is the free version of Oracle's Virtualbox running on Microsoft Windows 7 or Windows 8 Operating System. VirtualBox is a Type 2 Hypervisor, a virtualization software that allows virtual machines to be run on top of a host operating system [2]. VirtualBox provides an API [3] that allows developers to extend its functionalities. VirtualBox was chosen over other virtualization software because it "is the only professional [virtualization] solution that is freely available as Open Source Software" [4].

3.1.2 .NET Platform

The client and the server application will both run on a Microsoft Windows operating system hence the choice of the .NET development platform. Both the server and the client application were developed using Windows Presentation Form (WPF) and the C# programming language.

3.1.3 MySQL

MySQL is one of the most popular open source relational database management system [5]. It is the database software used for Virtual Lab's central database. MySQL provides a driver for the .NET development platform. MySQL was chosen over Microsoft SQL server because it is free and open source.

3.1.4 FTP Protocol

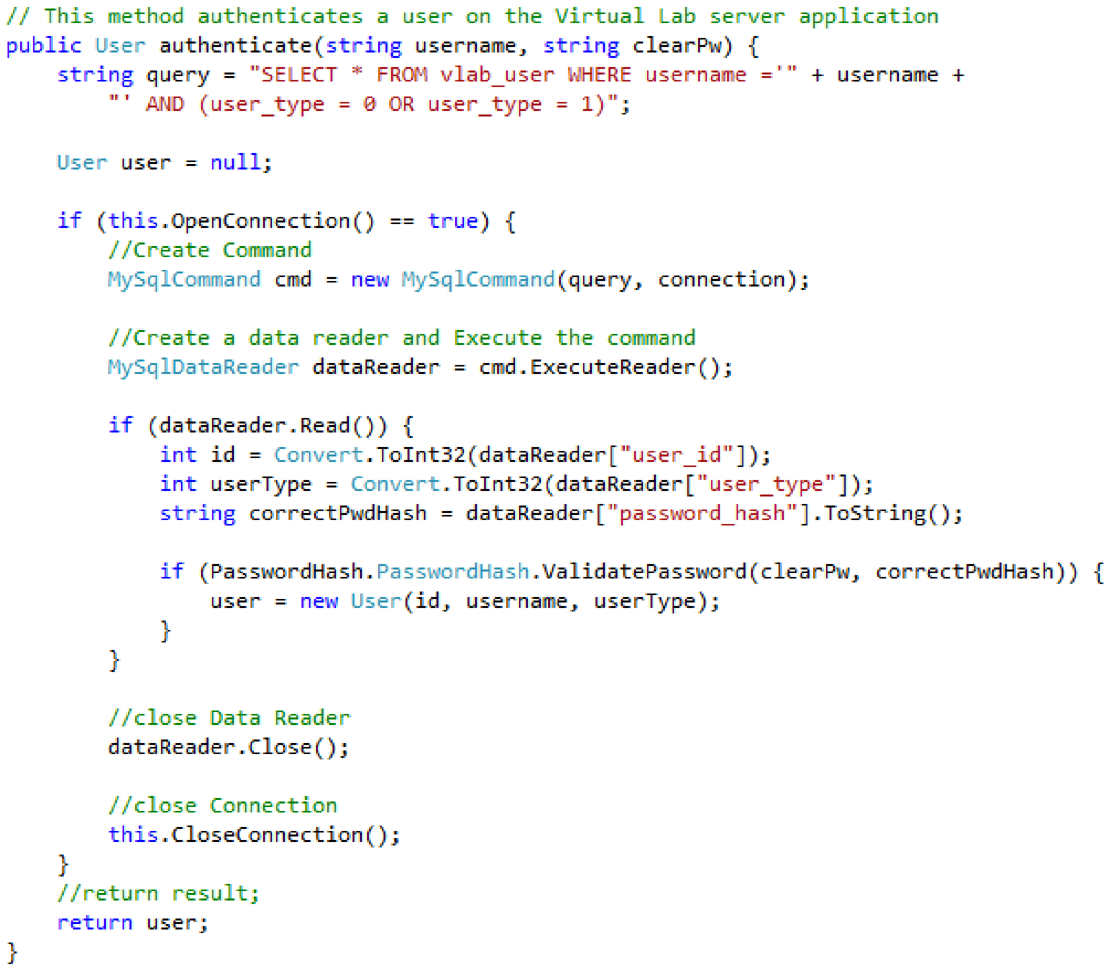
The File Transfer Protocol is a networking protocol that ensures reliable data transfer over a TCP/IP connection. This is the protocol used to copy the virtual machine files from the server computer to all the client computers.

3.2 Implementation

In order to use the VirtualBox developer API, one must add VBoxC.dll as a reference to the WPF project. This provides us with C# Classes and methods to interact with VirtualBox.

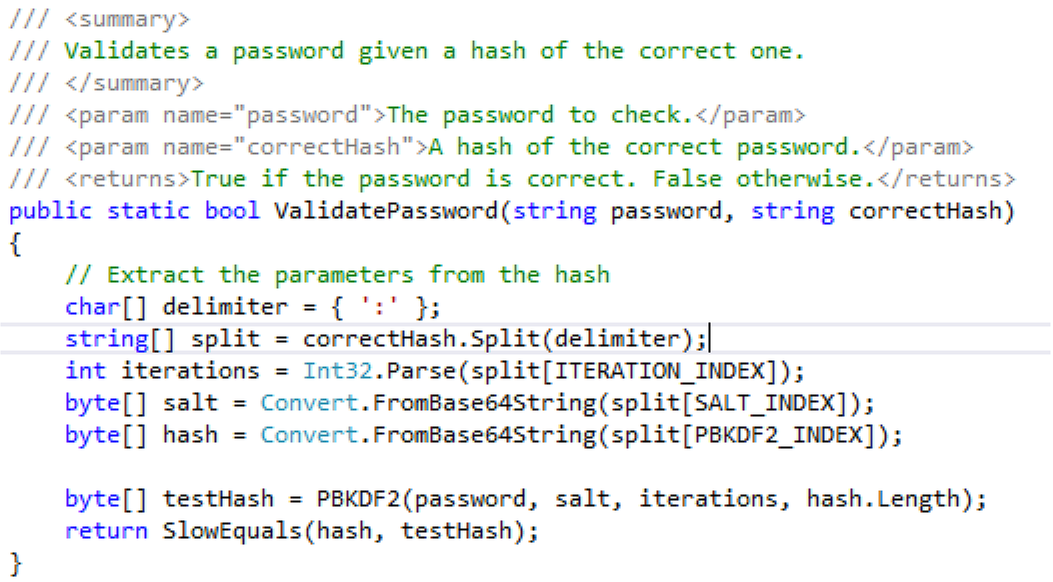
3.2.1 Authentication

To authenticate a user, we try to select from the vlab_user table where the credentials are equal to the one provided by the user. Following good practices for storing user credentials, the password is hashed and salted during the user registration process. This makes the authentication process more complex than expected (Figure 19).



```
// This method authenticates a user on the Virtual Lab server application
public User authenticate(string username, string clearPw) {
    string query = "SELECT * FROM vlab_user WHERE username ='" + username +
        "' AND (user_type = 0 OR user_type = 1)";

    User user = null;

    if (this.OpenConnection() == true) {
        //Create Command
        MySqlCommand cmd = new MySqlCommand(query, connection);

        //Create a data reader and Execute the command
        MySqlDataReader dataReader = cmd.ExecuteReader();

        if (dataReader.Read()) {
            int id = Convert.ToInt32(dataReader["user_id"]);
            int userType = Convert.ToInt32(dataReader["user_type"]);
            string correctPwdHash = dataReader["password_hash"].ToString();

            if (PasswordHash.PasswordHash.ValidatePassword(clearPw, correctPwdHash)) {
                user = new User(id, username, userType);
            }
        }

        //close Data Reader
        dataReader.Close();

        //close Connection
        this.CloseConnection();
    }
    //return result;
    return user;
}
```

Figure 19: Code: Authentication



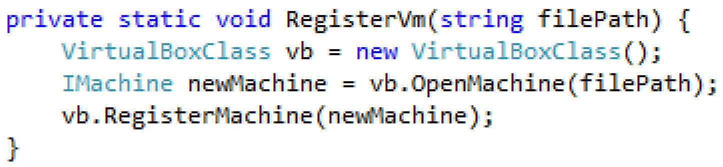
```
/// <summary>
/// Validates a password given a hash of the correct one.
/// </summary>
/// <param name="password">The password to check.</param>
/// <param name="correctHash">A hash of the correct password.</param>
/// <returns>True if the password is correct. False otherwise.</returns>
public static bool ValidatePassword(string password, string correctHash)
{
    // Extract the parameters from the hash
    char[] delimiter = { ':' };
    string[] split = correctHash.Split(delimiter);
    int iterations = Int32.Parse(split[ITERATION_INDEX]);
    byte[] salt = Convert.FromBase64String(split[SALT_INDEX]);
    byte[] hash = Convert.FromBase64String(split[PBKDF2_INDEX]);

    byte[] testHash = PBKDF2(password, salt, iterations, hash.Length);
    return SlowEquals(hash, testHash);
}
```

Figure 20: Code: Password Validation [6]

3.2.2 Importing Existing Virtual machine

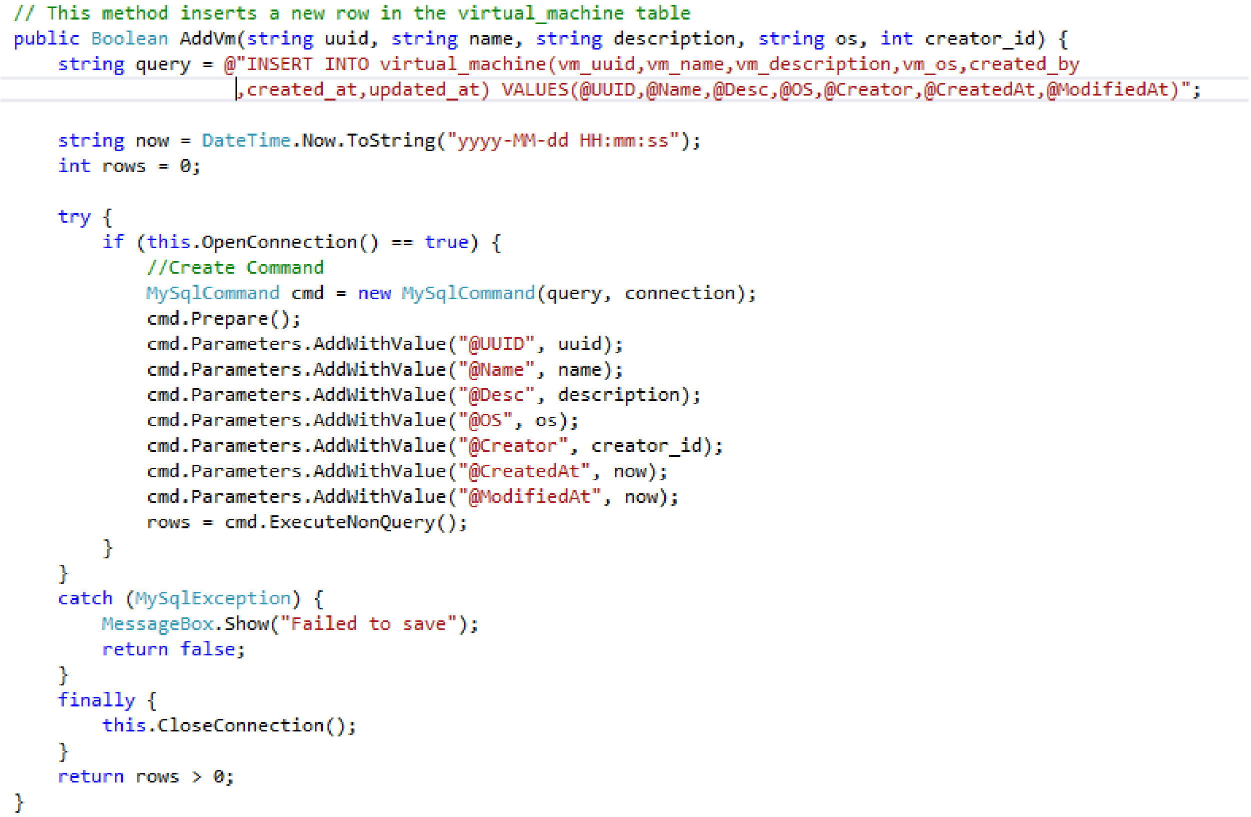
Importing a virtual machine into Virtual Lab is the process of registering a virtual machine with the Virtual Lab system. This process consists of two steps. The first step is to register the virtual machine with the Virtual Lab Server's instance of VirtualBox. This is done by providing the path to the virtual machine's .vbox xml file (Figure 21). This is necessary because the virtual machine cannot be launched if it is not registered with the local instance of VirtualBox.

Once the virtual machine is successfully registered, we can proceed to the second step which is about saving metadata about the virtual machine into the virtual_machine table in the central database (Figure 22).



```
private static void RegisterVm(string filePath) {
    VirtualBoxClass vb = new VirtualBoxClass();
    IMachine newMachine = vb.OpenMachine(filePath);
    vb.RegisterMachine(newMachine);
}
```

Figure 21: C# Code: register a VM with VirtualBox



```
// This method inserts a new row in the virtual_machine table
public Boolean AddVm(string uuid, string name, string description, string os, int creator_id) {
    string query = @"INSERT INTO virtual_machine(vm_uuid,vm_name,vm_description,vm_os,created_by
                   ,created_at,updated_at) VALUES(@UUID,@Name,@Desc,@OS,@Creator,@CreatedAt,@ModifiedAt)";

    string now = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss");
    int rows = 0;

    try {
        if (this.OpenConnection() == true) {
            //Create Command
            MySqlCommand cmd = new MySqlCommand(query, connection);
            cmd.Prepare();
            cmd.Parameters.AddWithValue("@UUID", uuid);
            cmd.Parameters.AddWithValue("@Name", name);
            cmd.Parameters.AddWithValue("@Desc", description);
            cmd.Parameters.AddWithValue("@OS", os);
            cmd.Parameters.AddWithValue("@Creator", creator_id);
            cmd.Parameters.AddWithValue("@CreatedAt", now);
            cmd.Parameters.AddWithValue("@ModifiedAt", now);
            rows = cmd.ExecuteNonQuery();
        }
    }
    catch (MySqlException) {
        MessageBox.Show("Failed to save");
        return false;
    }
    finally {
        this.CloseConnection();
    }
    return rows > 0;
}
```

Figure 22: Code: Insert into virtual_machine table

3.2.3 Deployment

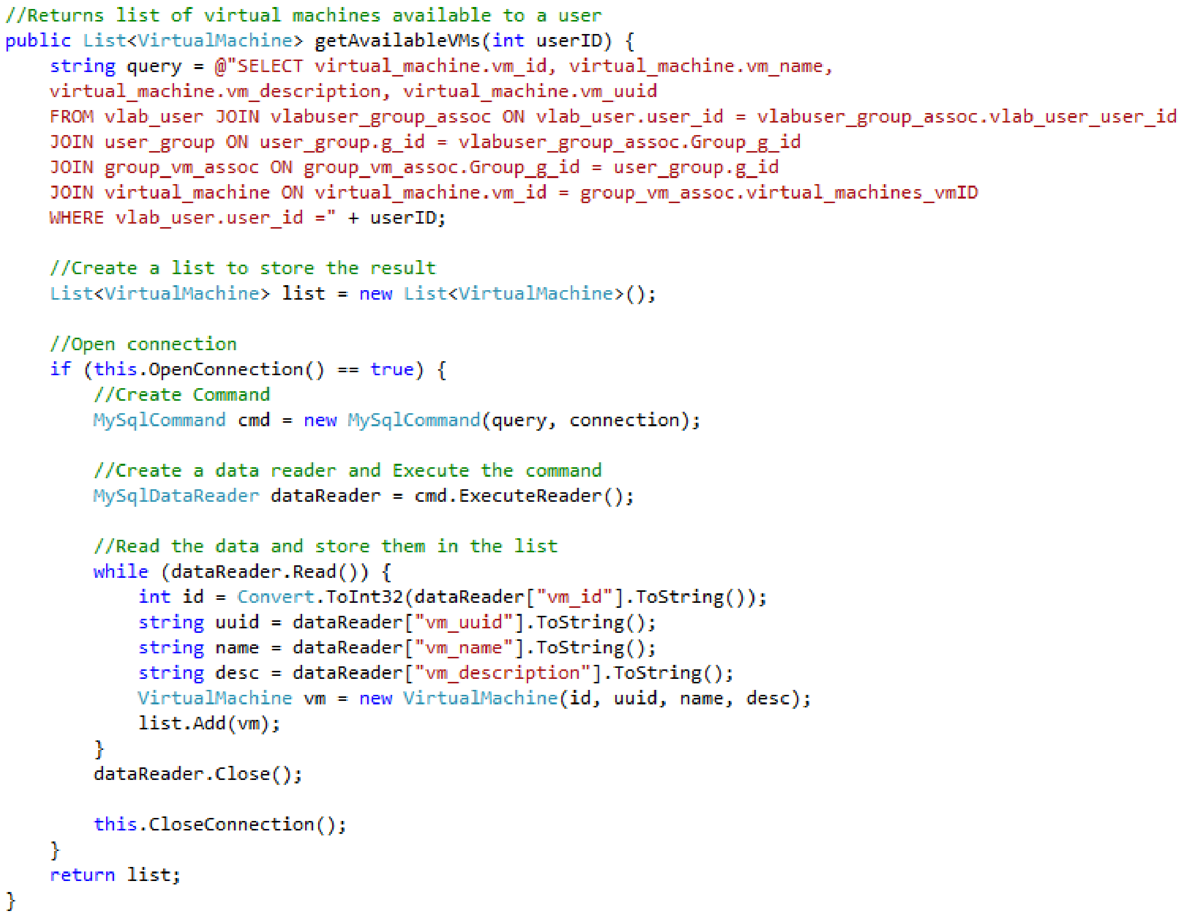
Deploying a virtual machine is simply the process of copying all files related to that virtual machine onto another computer. Virtual Lab automates this process that was previously done manually.

To copy the files from the server to the client computers, we rely on the FTP protocol. The client application runs an FTP server. The Virtual Lab server then connects to the client's FTP server and transfers the files.

Once the files are copied, the virtual Lab client registers the virtual machine with the local instance of VirtualBox and inserts its metadata in the central database.

3.2.4 Displaying VMs relevant to a student

Displaying the virtual machines relevant to a student is not as straight forward as it seems. This information is spread across 5 tables: vlab_user, virtual_machine, user_group, vlabuser_group_assoc and group_vm_assoc. Thus a SQL 'join' operation between these tables has to be performed to extract meaningful information.



```
//Returns list of virtual machines available to a user
public List<VirtualMachine> getAvailableVMs(int userID) {
    string query = @"SELECT virtual_machine.vm_id, virtual_machine.vm_name,
    virtual_machine.vm_description, virtual_machine.vm_uuid
    FROM vlab_user JOIN vlabuser_group_assoc ON vlab_user.user_id = vlabuser_group_assoc.vlab_user_user_id
    JOIN user_group ON user_group.g_id = vlabuser_group_assoc.Group_g_id
    JOIN group_vm_assoc ON group_vm_assoc.Group_g_id = user_group.g_id
    JOIN virtual_machine ON virtual_machine.vm_id = group_vm_assoc.virtual_machines_vmID
    WHERE vlab_user.user_id =" + userID;

    //Create a list to store the result
    List<VirtualMachine> list = new List<VirtualMachine>();

    //Open connection
    if (this.OpenConnection() == true) {
        //Create Command
        MySqlCommand cmd = new MySqlCommand(query, connection);

        //Create a data reader and Execute the command
        MySqlDataReader dataReader = cmd.ExecuteReader();

        //Read the data and store them in the list
        while (dataReader.Read()) {
            int id = Convert.ToInt32(dataReader["vm_id"].ToString());
            string uuid = dataReader["vm_uuid"].ToString();
            string name = dataReader["vm_name"].ToString();
            string desc = dataReader["vm_description"].ToString();
            VirtualMachine vm = new VirtualMachine(id, uuid, name, desc);
            list.Add(vm);
        }
        dataReader.Close();

        this.CloseConnection();
    }
    return list;
}
```

Figure 23: Code: selecting VMs available to a student

3.3 Challenges

One of the challenges faced early in this project was an exception that prevented the program from launching a virtual machine. It took weeks to find a solution. The solution was to downgrade from .NET framework 4.5 to .NET framework 4. There seem to be an incompatibility between the VirtualBox developer API and .NET framework 4.5.

Another issue faced was the difficulty to find an implementation of an FTP server that can inter-operate with a C# WPF application. The main reason for a C# implementation of an FTP server was the ability to bind to certain events such as 'on transfer completed'. The solution to this was to extend an FTP server found in a tutorial online [7].

Chapter 4: TEST AND RESULT

4.1 UNIT TESTING

The development of this application was done on a HP Envy dv6 Core i7 with 8GB of ram. Unit testing was done throughout the whole development process. It allowed me to find problems early and fix them. No specific unit testing framework was used.

4.2 COMPONENT TESTING

Virtual Lab consists of two main components which are the server application and the client application. The client application however, is loosely coupled with an FTP server which will be considered a third component. Each of these components was tested independently.

For the component testing, I used two Lenovo Desktop computers in the computer laboratory at Ashesi University. Those computers belonged to the same subnet. This ensured that they could communicate over the network.

Server Application: The server application was installed on one computer which was labeled 'Vlab server'. After setting up the database, the following features were tested: login, creating users (student, faculty), importing existing virtual machines. This test revealed a design issue: the database credentials were hardcoded into the source code. To allow the database credentials to be easily changed, they are now stored in a file from which the server application can read.

Client Application: The client application was installed on the second computer which was labeled ‘Client 1’. The label is important to identify the computer later.

To test the client application, I tried to connect to the remote database on the Vlab server computer.

This test revealed an issue with the two computers’ IP addresses: they were not static so they changed a few times and made testing tedious. To make testing easier, I set static IP addresses on the two computer. I also realized that I had to disable the firewall while testing.

FTP server: The FTP server was installed alongside the client application. To test the FTP server, I installed FileZilla, an FTP client, on the Virtual Lab server computer. I then used FileZilla to connect and upload virtual machine files to the FTP server.

Chapter 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 LIMITATIONS AND FURTHER WORKS

Like any software project, Virtual Lab does not pretend to be complete. This applied project merely represent its first version. Here are a few features that could make Virtual Lab more useful to an academic environment:

At the moment, a specific physical computer serves as the server. To create, edit or deploy a virtual machine, faculty have to walk to the location of that specific computer. This also means that not more than one faculty can work on Virtual Lab at the same time. This is not convenient nor scalable. A possible solution to this would be to make the current server application a web application. This means that it would be available from any computer on the institution's intranet or even over the internet. This would facilitate the work of the faculty who will not have to walk to a physical computer.

In this version of Virtual Lab, the virtual machine files are copied from the server to each of the client computers one after the other. Thus it takes a long time to copy the files to all client computers. It would be interesting to explore peer-to-peer technologies in order to allow the client computers to share files among themselves. This way, a client computer will not have to wait for the server to send the files, it can receive it from a peer that already has it. This would make deployment faster.

Finally, virtual Lab could be integrated with the institution’s S-drive so that students can save their work directly over the networks. This means that students will have access to their data on any computer that is connected to the institution’s intranet.

5.2 CONCLUSION

The use of virtual machines is an important instrument in allowing students to do computer science practical work. Although virtual machines have limitations, their benefits greatly outweigh these limitations. Virtual Lab is an attempt at streamlining the process of managing and deploying virtual machines. Both faculty, students and ultimately society will benefit from the implementation of Virtual Lab.

REFERENCES

[1] S. Uludag, E. Guler, M. Karakus and S. W. Turner, "An affordable virtual laboratory infrastructure to complement a variety of computing classes," *Journal of Computing Sciences in Colleges,* pp. 158-166 , 2012 .

[2] Oracle, "Chapter 1: First steps," [Online]. Available: https://www.virtualbox.org/manual/ch01.html. [Accessed 5 4 2014].

[3] Oracle, "VirtualBox Main API Documentation," 25 Feb 2014. [Online]. Available: https://www.virtualbox.org/sdkref/index.html. [Accessed 5 April 2014].

[4] Oracle, "Welcome to VirtualBox.org!," [Online]. Available: https://www.virtualbox.org/. [Accessed 6 March 2014].

[5] Oracle, "What is MySQL?," [Online]. Available: http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html. [Accessed 6 April 2014].

[6] T. Hornby, "Password Hashing With PBKDF2," 2013. [Online]. Available: https://crackstation.net/hashing-security.htm#aspsourcecode. [Accessed 5 April 2014].

[7] R. Bassham, "Creating an FTP Server in C# - with IPv6 Support," 7 October 2013. [Online]. Available: http://www.codeproject.com/Articles/380769/Creating-an-FTP-Server-in-

Csharp-with-IPv-Support. [Accessed 6 April 2014].