Marathwada Shikshan Prasarak Mandal's

Deogiri Institute of Engineering and Management Studies, Aurangabad

Project Report

On

I.R.O.N.A.

Submitted By

Vishal Panchal (46050)

Durgesh Rathod (46055)

Shreegeet Dixit (46069)

Dr. Babasaheb Ambedkar Marathwada University

Aurangabad (M.S.)

Department of Computer Science and Engineering

Deogiri Institute of Engineering and Management Studies, Aurangabad

(2016- 2017)

Project Report

On

I.R.O.N.A.

Submitted By

Vishal Panchal (46050)

Durgesh Rathod (46055)

Shreegeet Dixit (46069)

In partial fulfillment of

Bachelor of Engineering

(Computer Science & Engineering)

Guided By

Prof. P. M. Katkar

Department of Computer Science & Engineering

Deogiri Institute of Engineering and Management Studies,

Aurangabad

(2016- 2017)

CERTIFICATE

This is to certify that, the Project entitled "**I.R.O.N.A."** submitted by **Vishal Panchal, Durgesh Rathod, Shreegeet Dixit** is a bonafide work completed under my supervision and guidance in partial fulfillment for award of Bachelor of Engineering (Computer Science and Engineering) Degree of Dr. Babasaheb Ambedkar Marathwada University, Aurangabad.

Place: Aurangabad
Date: / /2017

Prof. P. M. Katkar
Project Guide
(Computer Science & Engineering)

Prof. S. B. Kalyankar
Head of Department
(Computer Science & Engineering)

Dr. Ulhas D. Shiurkar
Director,
Deogiri Institute of Engineering and Management Studies,
Aurangabad

DECLARATION

This is to certify that, the partial project report entitled, "**I.R.O.N.A.**" Submitted by **Vishal Panchal, Durgesh Rathod, Shreegeet Dixit** is a bonafide work completed under my supervision and guidance in partial fulfillment for award of Bachelor of Engineering in Computer Science and Engineering of Deogiri Institute of Engineering and Management Studies, Aurangabad under Dr. Babasaheb Ambedkar Marathwada University, Aurangabad.

Place: Aurangabad
Date: / /2017

External Examiner

Prof. P. M. Katkar
Project Guide

ABSTRACT

In educational organization as well as in giant IT organizations large number of Computer systems also called as workstation are used, they do have similar behavior, similar specifications despite of that while executing a common process on each system it is done manually. It requires a lot of efforts and human attendance. It makes man donkey in the digital era. Now-a-days the scenario has changed. In today's world, more focus is given on the availability of the internet and thus using various applications that can be handled remotely. This can be achieved by automating the simple task and running it parallel on every computer via a remote server. Giant IT organizations large number of Computer systems also called as workstation are used, they do have similar behavior, similar specifications despite of that while executing a common process on each system it is done manually. It requires a lot of efforts and human attendance. It makes man donkey in the digital era. Now-a-days the scenario has changed.

In today's world, more focus is given on the availability of the internet and thus using various applications that can be handled remotely. This can be achieved by automating the simple task and running it parallel on every computer via a remote server. Network automation is the use of IT controls to supervise and carry out everyday network management functions. This functions can range from basic network mapping and device discovery to network configuration management and to provisioning of virtual network resources. Public cloud services are already highly automated and sales service and many enterprise IT organization are moving to similar IT services model in their data centers.

This requires service level agreements (SLA) for the network, compute and storage resources that the organization business units required. Network automation plays key role in network vitualization technology, enabling automatic provisioning of virtual network tenants and network functions. Time is being affected by uneven conditions leading to increased time in manual installation of software.

CONTENTS

List of Figures

List of Tables

List of Screenshots

1. INTRODUCTION

1.1. Introduction

In educational organization as well as in giant IT organizations large number of Computer systems also called as workstation are used, they do have similar behavior, similar specifications despite of that while executing a common process on each system it is done manually. It requires a lot of efforts and human attendance. It makes man donkey in the digital era. Now-a-days the scenario has changed. In today's world, more focus is given on the availability of the internet and thus using various applications that can be handled remotely. This can be achieved by automating the simple task and running it parallel on every computer via a remote server. Network automation is the use of IT controls to supervise and carry out everyday network management functions.

This functions can range from basic network mapping and device discovery to network configuration management and to provisioning of virtual network resources. Public cloud services are already highly automated and sales service and many enterprise IT organization are moving to similar IT services model in their data centers. This requires service level agreements (SLA) for the network, compute and storage resources that the organization business units required. Network automation plays key role in network vitualization technology, enabling automatic provisioning of virtual network tenants and network functions.

1.2.Necessity

Time is being affected by uneven conditions leading to increased time in manual installation of software. This affects increased time. Moreover, it becomes even worst when the software are not installed on the first go. Also, increasing software burdens the processor to increase its efficiency. This is where modern automated software installation techniques and systems are needed to detect and prevent the time consuming process from being effected by different version of software. Last few years for installation of software different techniques have been used for example openOffice (windows), Ansible, network automation system, autoMate 10, volatile organic compounds profiling based etc.

1.2.1 Performance and SLAs

The first thing that network management does is performance monitoring. It's conceptually easy, but surprisingly challenging, primarily due to differences in vendors,

changes in standards (e.g., 32-bit vs 64-bit counters, different SNMP versions, and bugs in vendor implementations). Once those hurdles are past, the thousands of interfaces need to be sorted by a variety of criteria (e.g., percent utilization, error rates, broadcasts, etc). Alerting thresholds on performance data need to be defined and now you have a system that alerts you when utilization is high or errors suddenly appear on a link. Doing this task without automation is impossible in any network consisting of more than about 50 routers and switches.

SLAs are another area where automation is required. How else would you monitor the delay, jitter, and packet loss across a network (to pick three common SLA factors). An automated system is required for performing SLA tests, processing the results, and presenting the reports.

1.2.2. Scaling of processes

There are many processes in managing networks that should be performed regularly to have a smoothly running network with minimum downtime. But because these processes take a lot of time to implement manually, they are seldom performed. With network automation, these processes can be performed regularly, reducing risk of an unexpected network failure. Of course, the results of the processes should be sent to a network administrator, particularly regarding any alerts or exceptions. These processes include:

Compliance – Do your configurations adhere to your network policies? Check the security settings of your routers and switches or the network management settings (do all devices send syslog to the right place?)

Saving configurations – Save all configurations to non-volatile storage on the device and to a backup server.

Switch port utilization – Identify unused switch ports, allowing you to consolidate connections and perhaps re-deploy existing hardware.

Improve network resilience – Verify that the first hop redundancy protocol (HSRP/VRRP/GLBP) is configured and operating correctly. Consistent deployment – Are all devices running the OS that you have validated? Mixing multiple OS versions is a good way to encounter an unexpected bug.

1.2.3. Financial

Reduce operating costs by tracking the inventory of your network devices and paying maintenance only on those devices that are in your network. Know which devices you want

to upgrade next in a network refresh by tracking the age of all your devices and the OS loaded on them.

1.2.4. Topology

When troubleshooting, an accurate network topology drawing is valuable. Keeping network drawings up to date is a tedious and often neglected task and when a problem occurs, I typically see people sketching the network topology so they can proceed with the problem diagnosis. The NMS collects connectivity information, which can be displayed within the tool or exported to drawing tools (Microsoft has published the Visio XML format).

Topology information is also very valuable for network planning and preventing outages. It allows you to answer questions about uplink oversubscription ratios, verify redundant connections (or the lack thereof), and identify strange topologies that tend to appear in most networks (and that can cause strange behavior or failure modes).

1.2.5. Network Analysis

Network analysis is the process of taking all the collected data about a network and performing analysis on that data to identify current and potential problems. The simplest analysis is identifying interfaces running at high utilization. More complex analysis incorporates data from multiple devices, such as determining that a VRRP group only contains one router (the operational data from all routers shows that there is no peer router). The most complex analysis uses multiple sources of data, such as from both configuration files and operational data, exemplified by a duplex mismatch where an interface configuration shows a setting of ‘auto’, the interface’s state is ‘half’, and operational data shows late collisions.

Other network analysis incorporates data sources like events (syslog or SNMP traps). Most network management systems collect the data but then rely on the network engineer to perform the analysis. Because the network engineer is already busy, this limits what he or she can do, so it often defaults to looking at alerts generated by the interface utilization thresholds. Automating the analysis tasks allows easy identification of lots of problems that network engineers know that they know should be done but never have the time to perform.

1.2.6. Correlation of the above items

The next step in automation is to correlate several of the above items. A good example is to use the topology information to perform higher frequency interface performance polling on any interface where the neighboring device is another infrastructure device. Edge ports can be polled at a much lower frequency. Another example is using the topology information to

determine whether a subnet has been allocated multiple times. Similarly, it would be good to use topology to tell if two subnets that overlap, but have different masks are on the same segment due to a typo in the configuration or are they two different subnets in different parts of the network.

1.2.7. Human error

The biggest and most important reason for network automation is human error. It accounts for at least 40% of network failures (some estimates are as high as 80%). It has been proven that automation helps reduce those errors. Updating the configurations of hundreds of routers and switches is not something that should be done manually. Automated mechanisms to verify a proposed change and to implement a change control process where it is validated by other network engineers is important for reducing or eliminating silly mistakes.

That's the list. Networks are big. Networks are complex and are increasing in complexity. Automation is the only hope we have of managing the size and complexity while providing high availability.

1.3. Objectives

The main objective of this project is to help to increase process speed, reliability, and control and avoid human participation. In educational organization as well as in giant IT organizations large number of Computer systems also called as workstation are used, they do have similar behavior, similar specifications despite of that while executing a common process on each system it is done manually.

1.3.1 Full Client-Server Architecture

It is a multi-tiered client server business process automation platform for Windows. It logically separates high-level workflow design from the complexity of building detailed automation routines and provides automated execution and centralized control over processes involving one machine or the entire enterprise. With comprehensive job scheduling, event-based conditions, and robust application development tools, it enables organizations to streamline, manage and automate complex operations without writing a single line of code.

- **Multi-Machine Execution**

It multi-machine execution capability is made possible in part by BPA Agents that are deployed or installed on remote machines and their facility to interact with the server

component. Once installed or deployed on a machine, whether that machine is local or situated remotely, the server component can connect with the Agent via TCP/IP and notify it of what system or network events to monitor along with what tasks to execute as a result of such conditions or events occurring.

- **Centralized Management**

The Server Management Console features a modern and ultra-intuitive interface. Capabilities include: Management of workflows, repository objects, users and agents, reporting, real-time execution status and history, and settings and properties for the server including an object inheritance engine. With centralized administration and reporting, It provides unparalleled management of workflows and IT operations. All workflow information is centrally stored on the server. Workflow performance reporting is also centralized so administrators can easily monitor operations and quickly react to any exceptions.

- **Workflows**

Graphical workflows are the cornerstone of I.R.O.N.A. A workflow (or flow diagram) is a high-level graphical representation of an IT or business process, which might involve processing on multiple servers across the network. The main components of workflows include conditions, tasks, and flow control objects. Conditions are network or system events. The path a workflow takes changes based on conditions (such as the appearance of a file in a specified network share folder) that can initiate workflow execution or provide the basis for conditional event-based decision-making. Tasks, which can be simple or complex, involve the instructions for executing a series of processing steps. Tasks are created with award-winning drag-and-drop development tool, the Task Builder, which eliminates the need for writing and maintaining code. Flow control objects allow for the inclusion of complex logic within the workflow and conditional branching.

- **Intuitive Task Construction and Debugging**

The Task Builder is easy-to-use, intuitive interface for developing automation applications. It is used to visually construct and examine the steps that an task should carry out when it is run. To ensure that newly created tasks are working properly before they are put into production, the Task Builder also includes a variety of testing and debugging features that outputs real time information about a running task, enables the inspection of variables, datasets and other dynamic data, and many other debugging features which aids a developer in monitoring and examining tasks as they are being constructed.

- **Conditions**

Conditions can be used to initiate workflows (i.e., start workflow A when file X appears),

wait for certain events to transpire before an active workflow is permitted to proceed past a certain point (i.e., after workflow A is initiated, wait for the Notepad Window to appear before continuing), or to make a decision and conditional branch within the workflow (i.e., after workflow A is initiated, evaluate whether the Notepad Window is open – if yes, execute Task 1; if no, execute Task 2).

- **FTP / SFTP**

File Transfer protocol (FTP) is one of the earliest yet still most common method of transferring files across the Internet mainly due to its simplicity and cross-platform capability. The FTP module in acts as an FTP client, however, its advantage over other equivalent clients is that it has the ability to automate unattended file transfer operations. This can be beneficial to organizations that conduct daily transfers of many files or files that are significantly large in size. In such cases, FTP tasks can be created and scheduled to execute in the evening, when network traffic is at a minimum, allowing for more rapid and reliable transfers. Internal FTP engine is compatible with all FTP connection methods, particularly security-based connections. This includes both explicit and implicit FTPS (FTP Secure or FTP-SSL) that adds support for the Transport Layer Security (TLS) and the Secure Sockets Layer (SSL) cryptographic protocols. Certificate Authenticity is supported for those servers that require clients to authenticate themselves using a certificate. SFTP (or Secure FTP), which uses Secure Shell (SSH) protocol is also supported. Additionally, it supports Tumbleweed Secure Transport, the industry's most secure Managed File Transfer solution.

1.4 Scope:

- Automation Simplified service creation and maintenance processes via computer-aided provisioning and the introduction of service templates
- Data and process integration Fully integrated process for automating management in all network infrastructure tiers and a generic interface to operations support systems
- Reporting Comprehensive service-assurance function for monitoring end-to-end performance and the availability of optical services.
- Full life cycle support Spanning the full lifecycle of optical service management – provisioning, change, performance and availability management.

1.5 Uses

Automatic configuration and provisioning:

Automation capabilities such as provisioning are embedded into the network systems' architecture. Many are considered standard features today, but often started their lives as automation features. A basic example is DHCP since you don't allocate static IP addresses for client devices. It's so fundamental that we don't think of it as automation, but has become so useful we can't think of deploying client devices without it. A more modern example for infrastructure networking systems is Brocade's Zero Touch Provisioning for switch image and configuration downloads since it makes configuring switches a lot easier.

2. LITERATURE SURVEY

2.1 Literature Survey

2.1 Ansible

Ansible is a free-software platform for configuring and managing computers which combines multi-node software deployment, ad hoc task execution, and configuration management. It manages nodes over SSH or over PowerShell. Modules work over JSON and standard output and can be written in any programming language. The system uses YAML to express reusable descriptions of systems.

Michael DeHaan, the author of the provisioning server application Cobbler and co-author of the FUNC framework for remote administration, developed the platform. It is included as part of the Fedora distribution of Linux, owned by Red Hat Inc., and is also available for Red Hat Enterprise Linux, CentOS, and Scientific Linux via Extra Packages for Enterprise Linux (EPEL) as well as for other operating systems. Ansible, Inc. (originally AnsibleWorks, Inc.) was the company set up to commercially support and sponsor Ansible. It was acquired by Red Hat in October 2015. The name "Ansible" references a fictional instantaneous hyperspace communication system (as featured in Orson Scott Card's Ender's Game (1985), and originally invented by Ursula K. Le Guin for her novel Rocannon's World (1966)).

2.1.1 Architecture

As with most configuration management software, Ansible distinguishes two types of servers: controlling machines and nodes. First, there is a single controlling machine which is where orchestration begins. Nodes are managed by a controlling machine over SSH. The controlling machine describes the location of nodes through its inventory.

To orchestrate nodes, Ansible deploys modules to nodes over SSH. Modules are temporarily stored in the nodes and communicate with the controlling machine through a JSON protocol over the standard output. When Ansible is not managing nodes, it does not consume resources because no daemons or programs are executing for Ansible in the background. In contrast with popular configuration management software — such as Chef, Puppet, and CFEngine — Ansible uses an agentless architecture. With an agent-based architecture, nodes must have a locally installed daemon that communicates with a controlling machine. With an agentless architecture, nodes are not required to install and run background daemons to connect with a

controlling machine. This type of architecture reduces the overhead on the network by preventing the nodes from polling the controlling machine.

2.1.2 Design goals

The design goals of Ansible include:

1. Minimal in nature. Management systems should not impose additional dependencies on the environment.
2. Consistent.
3. Secure. Ansible does not deploy agents to nodes. Only OpenSSH is required, which is thoroughly tested.
4. Highly reliable. When carefully written, an Ansible playbook can be idempotent, in order to prevent unexpected side-effects on the managed systems. It has to be noted, anyway, that it is perfectly possible to have a poorly written playbook that is not idempotent.
5. Low learning curve. Playbooks use an easy and descriptive language based on YAML and Jinja templates.

2.1.3 Modules

Modules are considered to be the units of work in Ansible. Each module is mostly standalone and can be written in a standard scripting language (such as Python, Perl, Ruby, bash, etc.). One of the guiding properties of modules is idempotency, which means that even if an operation is repeated multiple times (e.g., upon recovery from an outage), it will always place the system into the same state.

2.1.4 Platform support

Control machines must have Python 2.6 or 2.7. Operating systems supported on control machines include most Linux and UNIX distributions, such as Red Hat, DEBIAN, CentOS, OS X, BSD, and Ubuntu, among others. Managed nodes must have Python 2.4 or later. For managed nodes with Python 2.5 or earlier, the python-simple JSON package is also required. Ansible can manage Windows nodes starting from version 1.7.

2.1.5 Cloud integration

Ansible can deploy to virtualization environments and public and private cloud environments, including Amazon Web Services, CloudStack, DigitalOcean, Eucalyptus Cloud, Google Cloud Platform, KVM, Microsoft Azure, OpenStack, Rackspace, SoftLayer, VMware, and XenServer.

2.1.6 Big data integration

Ansible can deploy big data, storage, and analytics environments, including Hadoop, Riak, and Aerospike. In these environments, Ansible helps manage each node's resource consumption, consuming little CPU time and memory in the process. Furthermore, Ansible provides monitoring capabilities that measure such properties as available CPU resources, which can help in the supervision of these nodes.

2.2 Nmap

Nmap (Network Mapper) is a security scanner originally written by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich)[2] used to discover hosts and services on a computer network, thus creating a "map" of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host and then analyzes the responses.The software provides a number of features for probing computer networks, including host discovery and service and operating system detection. These features are extensible by scripts that provide more advanced service detection,[3] vulnerability detection,[3] and other features. Nmap is also capable of adapting to network conditions including latency and congestion during a scan. Nmap is under development and refinement by its user community.Nmap was originally a Linux-only utility,[4] but it was ported to Windows, Solaris, HP-UX, BSD variants (including OS X), AmigaOS, and IRIX.[5] Linux is the most popular platform, followed closely by Windows.[6]

2.2.1 Features

Nmap features include:

Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to TCP and/or ICMP requests or have a particular port open.

Port scanning – Enumerating the open ports on target hosts.

Version detection – Interrogating network services on remote devices to determine application name and version number.[7]

OS detection – Determining the operating system and hardware characteristics of network devices.

Scriptable interaction with the target – using Nmap Scripting Engine (NSE) and Lua programming language. Nmap can provide further information on targets, including reverse DNS names, device types, and MAC addresses.[8]

2.2.2 Typical uses of Nmap:

Auditing the security of a device or firewall by identifying the network connections which can be made to, or through it.[3]Identifying open ports on a target host in preparation for auditing.[2]Network inventory, network mapping, maintenance and asset management. Auditing the security of a network by identifying new servers.[4] Generating traffic to hosts on a network, response analysis and response time measurement.[5] Find and exploit vulnerabilities in a network.[8]

2.2.3 Graphical interfaces

NmapFE, originally written by Zach Smith, was Nmap's official GUI for Nmap versions 2.2 to 4.22.[14] For Nmap 4.50 (originally in the 4.22SOC development series) NmapFE was replaced with Zenmap, a new official graphical user interface based on UMIT, developed by Adriano Monteiro Marques.

Various web-based interfaces allow controlling Nmap or analyzing Nmap results from a web browser. These include LOCALSCAN,[15] nmap-web,[16] and Nmap-CGI.[17] Microsoft Windows specific GUIs exist, including NMapWin,[18] which has not been updated since June 2003 (v1.4.0), and NMapW[19] by Syhunt.

Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and official binary packages are available for Linux, Windows, and Mac OS X. In addition to the classic command-line Nmap executable, the Nmap suite includes an advanced GUI and results viewer (Zenmap), a flexible data transfer, redirection, and

debugging tool (Ncat), a utility for comparing scan results (Ndiff), and a packet generation and response analysis tool (Nping).Nmap was named “Security Product of the Year” by Linux Journal, Info World, LinuxQuestions.Org, and Codetalker Digest. It was even featured in twelve movies, including The Matrix Reloaded, Die Hard 4, Girl With the Dragon Tattoo, and The Bourne Ultimatum.

Nmap is...

- **Flexible:** Supports dozens of advanced techniques for mapping out networks filled with IP filters, firewalls, routers, and other obstacles. This includes many port scanning mechanisms (both TCP & UDP), OS detection, version detection, ping sweeps, and more. See the documentation page.
- **Powerful:** Nmap has been used to scan huge networks of literally hundreds of thousands of machines.
- **Portable:** Most operating systems are supported, including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga, and more. Easy: While Nmap offers a rich set of advanced features for power users, you can start out as simply as "nmap -v -A targethost". Both traditional command line and graphical (GUI) versions are available to suit your preference. Binaries are available for those who do not wish to compile Nmap from source.
- **Free:** The primary goals of the Nmap Project is to help make the Internet a little more secure and to provide administrators/auditors/hackers with an advanced tool for exploring their networks.

Nmap is available for free download, and also comes with full source code that you may modify and redistribute under the terms of the license.

- **Well Documented:** Significant effort has been put into comprehensive and up-to-date man pages, whitepapers, tutorials, and even a whole book! Find them in multiple languages here.
- **Supported:** While Nmap comes with no warranty, it is well supported by a vibrant community of developers and users. Most of this interaction occurs on the Nmap mailing lists. Most bug reports and questions should be sent to the nmap-dev list, but only after you read the guidelines. We recommend that all users subscribe to the low-traffic nmap-hackers announcement list. You can also find Nmap on Facebook and Twitter. For real-time chat, join the #nmap channel on Freenode or EFNet.
- **Acclaimed:** Nmap has won numerous awards, including "Information Security Product of the Year" by Linux Journal, Info World and Codetalker Digest. It has been featured in

hundreds of magazine articles, several movies, dozens of books, and one comic book series. Visit the press page for further details.

- **Popular:** Thousands of people download Nmap every day, and it is included with many operating systems (Red Hat Linux, DEBIAN Linux, Gentoo, FreeBSD, Open BSD etc). It is among the top ten (out of 30,000) programs at the Freshmeat.Net repository.

2.3 Application

Universal Silent Switch Finder

Universal Silent Switch Finder is a compact utility designed to help you find the silent switch in the applications that you want to install. If you want to deploy software without the user intervention you can use this tool in order to run the installer silently.

- **Suitable for LAN environments**

Network administrators often use the silent installation in order to deploy applications, patches or updates on multiple computers from the network. It is a useful option that saves time and can be performed remotely, without the administrator being present in front of the workstation.

- **Not applicable to all programs**

However, not all the programs have the option of being deployed with a silent switch. Or even if the option exists, it is not described in the documentation offered by the developer. In the latter case, the administrator can use Universal Silent Switch Finder to determine if the application has a silent switch and how it can be accessed.

- **Automatically detects installer type**

The interface of the program is rather simple and allows you to simply select the installer in order to display the available commands. It includes the PEiD tool that can identify the type of the installer and the packer that was used to create it.

- **Quickly gets options ready**

The analysis takes just a few seconds before the program can display the file extension and the file type in the main window. If the file has been recognized, you can also view the recommended usage that includes the command line for launching the installation in silent mode.

- **Good but far from being a pro**

Unfortunately, the app is not actively maintained and failed to identify most of the installers that used the EXE extension. In our tests it had better results with the MSI files that were successfully identified.

The file analysis and switch identification required insignificant resources to run and had no impact on the computer performance. When the silent switch is identified, the provided command line correctly launches the installation in command line mode.

3. SYSTEM DEVELOPMENT

3.1 System Architecture

3.1.1 Specification Document/UML Diagrams of all modules

Use case diagram

To model a system the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running /operating. So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Purpose

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when initial task is complete use case diagrams are modelled to present outside view.

- So in brief, the purposes of use case diagrams can be as follows:
- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

Use Case:

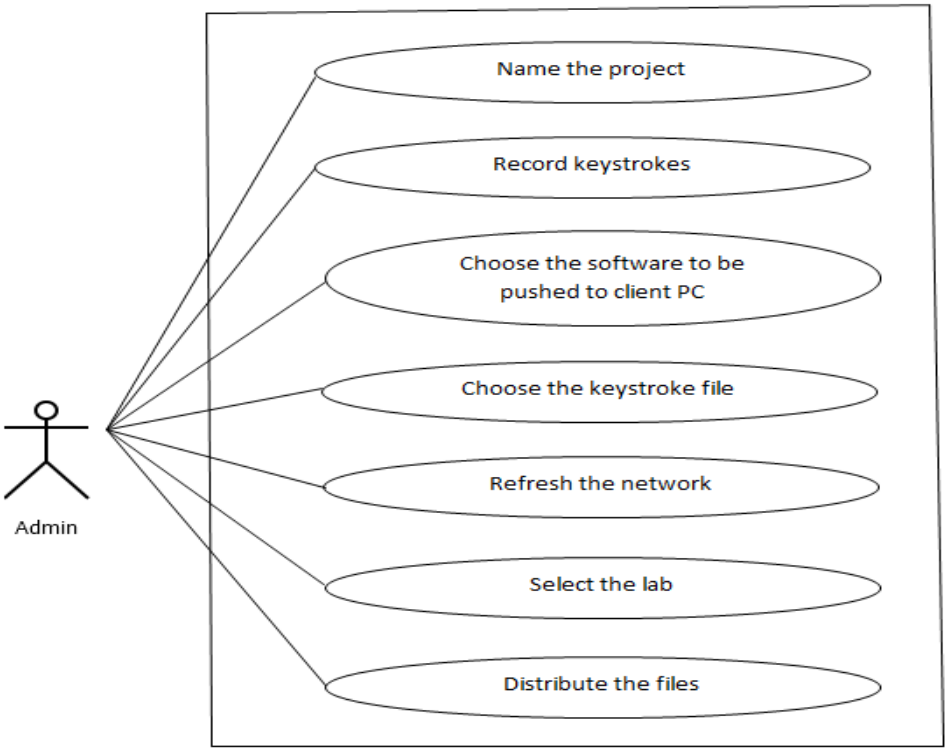


Figure 3.1: Use Case

The above use case diagram show the various task perform by the user.

Class Diagram:

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages. The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

Purpose:

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.

The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community.

- So the purpose of the class diagram can be summarized as:
- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

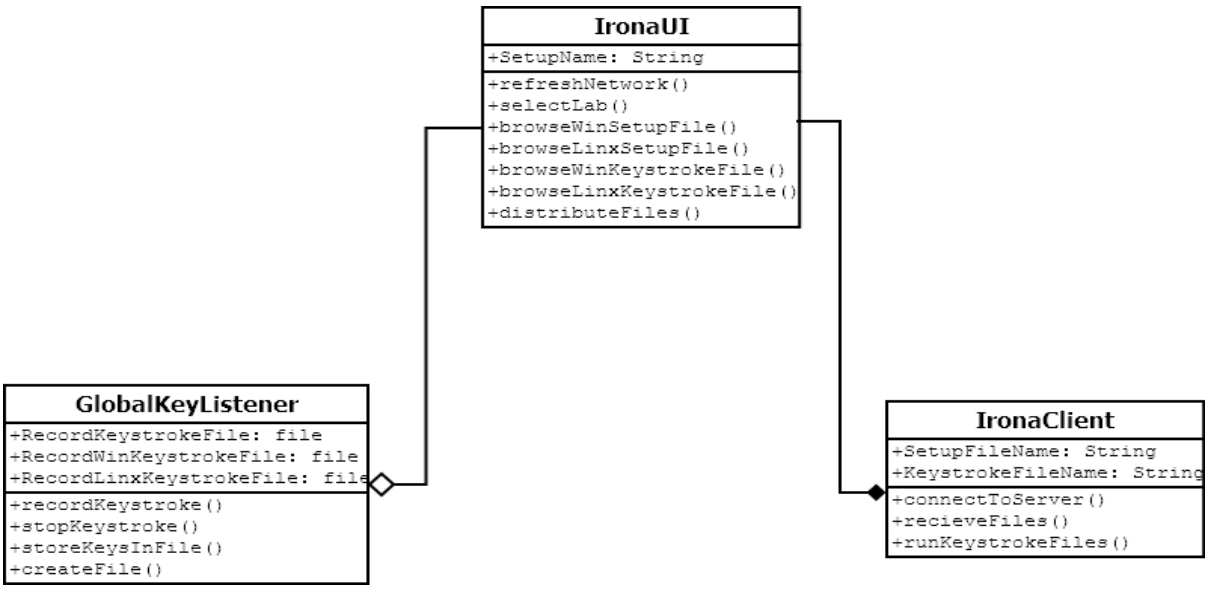


Figure 3.2: Class Diagram

The above class diagram show the classes including in that proposed system. It is also helpful for identify the various function of the particular class. Relation between two classes are also mentioned in that diagram.

3.2. Coding

```
package my.irona;

import com.sun.xml.internal.ws.api.ha.StickyFeature;
import java.io.BufferedInputStream;
import java.io.File;
import java.io.IOException;
import java.net.MulticastSocket;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFileChooser;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import java.io.FileWriter;
import java.io.FileInputStream;
import java.io.BufferedOutputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.net.Socket;
import java.nio.file.Path;
import java.nio.file.Paths;
public class IronaUI extends javax.swing.JFrame {

    /**
     * Creates new form IronaUI
     */
    public IronaUI() throws IOException {
     if(System.getProperty("os.name").toLowerCase().contains("win".toLowerCase()))
       {
           keypath= "c:\\users\\"+System.getProperty("user.name")+"\\keystrokes.txt";
       }
```

```
        else if(System.getProperty("os.name").toLowerCase().contains("linux".toLowerCase()))
        {
           keypath="/tmp/keystrokes.txt";
        }
           System.out.println(keypath);
  initComponents();
  ImageIcon image=new ImageIcon("C:\\Users\\Shreegeet\\Desktop\\Server final
proj\\irona.jpg");
       jLabel6.setIcon(image);
         fw=new FileWriter(keypath);
         fw.write("");
         fw.close();
        winFile = new String();
        linuxFile = new String();
        winKeyStrokes = new String();
        linuxKeyStrokes = new String();
        recording=0;
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {
        jPanel1 = new javax.swing.JPanel();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jTextField5 = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel6 = new javax.swing.JLabel();
        jPanel3 = new javax.swing.JPanel();
        jTextField3 = new javax.swing.JTextField();
```

```
jButton3 = new javax.swing.JButton();
jTextField4 = new javax.swing.JTextField();
jButton4 = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jButton7 = new javax.swing.JButton();
jPanel4 = new javax.swing.JPanel();
jButton5 = new javax.swing.JButton();
jComboBox1 = new javax.swing.JComboBox<>();
jButton6 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("I.R.O.N.A.");
jTextField1.setEnabled(false);
jTextField2.setEnabled(false);
jButton1.setText("Browse");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("Browse");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jLabel1.setText("Windows file");
jLabel2.setText("Linux file");
jLabel5.setText("Software Name");
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
```

```
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
                    .addComponent(jTextField1, javax.swing.GroupLayout.DEFAULT_SIZE, 197,
Short.MAX_VALUE)
                    .addComponent(jTextField2))
                .addGap(61, 61, 61)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jButton1)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 64,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jButton2)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE, 135,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(180, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                    .addComponent(jButton1)
                    .addComponent(jLabel1)
                    .addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel5)))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton2)
                .addComponent(jLabel2))
            .addGap(0, 13, Short.MAX_VALUE))
        );

        jPanel2.setBackground(new java.awt.Color(0, 0, 0));

jPanel2.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelB
order.RAISED));

        jLabel6.setText("icon");
```

```
        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE, 355,
Short.MAX_VALUE)
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
        );
        jTextField3.setEnabled(false);
        jButton3.setText("Browse");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        jTextField4.setEnabled(false);
        jButton4.setText("Browse");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton4ActionPerformed(evt);
            }
        });
        jLabel3.setText("Windows Keystroke");
        jLabel4.setText("Linux Keystroke");
```

```
        jButton7.setText("Start Recording Keystrokes");
        jButton7.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton7ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                    .addGroup(jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG, false)
                            .addComponent(jTextField3, javax.swing.GroupLayout.DEFAULT_SIZE,
174, Short.MAX_VALUE)
                            .addComponent(jTextField4))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
                            .addGroup(jPanel3Layout.createSequentialGroup()
                                .addComponent(jButton3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(jLabel3))
                            .addGroup(jPanel3Layout.createSequentialGroup()
                                .addComponent(jButton4)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel4))))
                .addComponent(jButton7))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        jPanel3Layout.setVerticalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addComponent(jButton7)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 13,
Short.MAX_VALUE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton3)
                .addComponent(jLabel3))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton4)
                .addComponent(jLabel4))
            .addGap(19, 19, 19))
        );
        jButton5.setText("Refresh Network");
        jButton5.addActionListener(new java.awt.event.ActionListener() {
```

```
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton5ActionPerformed(evt);
            }
        });

        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "lab1", "lab2", "lab3", "lab4", "lab5", " " }));

        jButton6.setText("Distribute Files");
        jButton6.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton6ActionPerformed(evt);
            }
        });
        javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jButton5)
                .addGap(35, 35, 35)
                .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(32, 32, 32)
                .addComponent(jButton6)
                .addContainerGap(127, Short.MAX_VALUE))
        );
        jPanel4Layout.setVerticalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel4Layout.createSequentialGroup()
                .addContainerGap(40, Short.MAX_VALUE)
```

```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(jButton5)
                .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(36, 36, 36))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        );
```

```
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(0, 37, Short.MAX_VALUE))
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton6ActionPerformed
        // TODO add your handling code here:
            System.out.print("Distributing the files to all selected ip::
"+jTextField1.getText().toString());
            //Socket programming should be done here
try{
```

```
   Socket sock = new Socket("172.16.9.21", 12345); // <<--------Socket code to send a file
File file1 = new File(jTextField1.getText().toString());
length = file1.length();
System.out.println(length);
Path p = Paths.get(jTextField1.getText());
String fname= p.getFileName().toString();
System.out.println("FILE NAME IS "+fname);
            FileInputStream fis = new FileInputStream(file1);
       bis = new BufferedInputStream(fis);
       dis = new DataInputStream(bis);
       bos = new BufferedOutputStream(sock.getOutputStream());
       dos = new DataOutputStream(bos);
       dos.writeInt((int) length); // sends the length as number bytes is file size
       int count = 0;
       byte[] buffer = new byte[(int) length];
       while ((count = bis.read(buffer)) > 0)
       {
          bos.write(buffer, 0, count);
       }
bos.flush();
       bis.close();
       sock.close();
   bos.close();
   sock.close();
}
   catch(Exception e)
{
   System.out.println(e );
}

   }//GEN-LAST:event_jButton6ActionPerformed
   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton1ActionPerformed
       JFileChooser fileChooser = new JFileChooser();
```

```
    int returnValue = fileChooser.showOpenDialog(null);
    if (returnValue == JFileChooser.APPROVE_OPTION)
    {
       File selectedFile = fileChooser.getSelectedFile();
       System.out.println(selectedFile.getName());
           try {
              jTextField1.setText(selectedFile.getCanonicalPath());
              winFile=selectedFile.getCanonicalPath();
              winFile.replace("\\", "\\\\");


           } catch (IOException ex) {
              Logger.getLogger(IronaUI.class.getName()).log(Level.SEVERE, null, ex);
           }

    }

    }//GEN-LAST:event_jButton1ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton2ActionPerformed


        JFileChooser fileChooser = new JFileChooser();
    int returnValue = fileChooser.showOpenDialog(null);
    if (returnValue == JFileChooser.APPROVE_OPTION)
    {
       File selectedFile = fileChooser.getSelectedFile();
       System.out.println(selectedFile.getName());
           try {
              jTextField2.setText(selectedFile.getCanonicalPath());
              linuxFile=selectedFile.getCanonicalPath();
              linuxFile.replace("\\", "\\\\");
              System.out.println(linuxFile);
```

```
        } catch (IOException ex) {
            Logger.getLogger(IronaUI.class.getName()).log(Level.SEVERE, null, ex);
        }

    }
    }//GEN-LAST:event_jButton2ActionPerformed

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

        JFileChooser fileChooser = new JFileChooser();
    int returnValue = fileChooser.showOpenDialog(null);
    if (returnValue == JFileChooser.APPROVE_OPTION)
    {
       File selectedFile = fileChooser.getSelectedFile();
       System.out.println(selectedFile.getName());
           try {
              jTextField3.setText(selectedFile.getCanonicalPath());
              winKeyStrokes=selectedFile.getCanonicalPath();
              winKeyStrokes.replace("\\", "\\\\");
          } catch (IOException ex) {
              Logger.getLogger(IronaUI.class.getName()).log(Level.SEVERE, null, ex);
           }

    }
        // } else {

         //}
    }//GEN-LAST:event_jButton3ActionPerformed

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton4ActionPerformed


        JFileChooser fileChooser = new JFileChooser();
```

```
   int returnValue = fileChooser.showOpenDialog(null);
   if (returnValue == JFileChooser.APPROVE_OPTION)
   {
      File selectedFile = fileChooser.getSelectedFile();
      System.out.println(selectedFile.getName());
         try {
            jTextField4.setText(selectedFile.getCanonicalPath());
            linuxKeyStrokes=selectedFile.getCanonicalPath();
            linuxKeyStrokes.replace("\\", "\\\\");



         } catch (IOException ex) {
            Logger.getLogger(IronaUI.class.getName()).log(Level.SEVERE, null, ex);
         }

   }
   }//GEN-LAST:event_jButton4ActionPerformed

   private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton5ActionPerformed

      System.out.println(System.getProperty("os.name"));

      if(System.getProperty("os.name").toLowerCase().contains("win".toLowerCase()))
      {
         System.out.println("executing the ping sweep in cmd prompt");
         JOptionPane.showMessageDialog(null, "Network is successfully refreshed", "Network
Refresh Done" , JOptionPane.INFORMATION_MESSAGE);
      }
      else if(System.getProperty("os.name").toLowerCase().contains("linux".toLowerCase()))
      {
         System.out.println("executing the ping sweep in linux terminal");

         try {
```

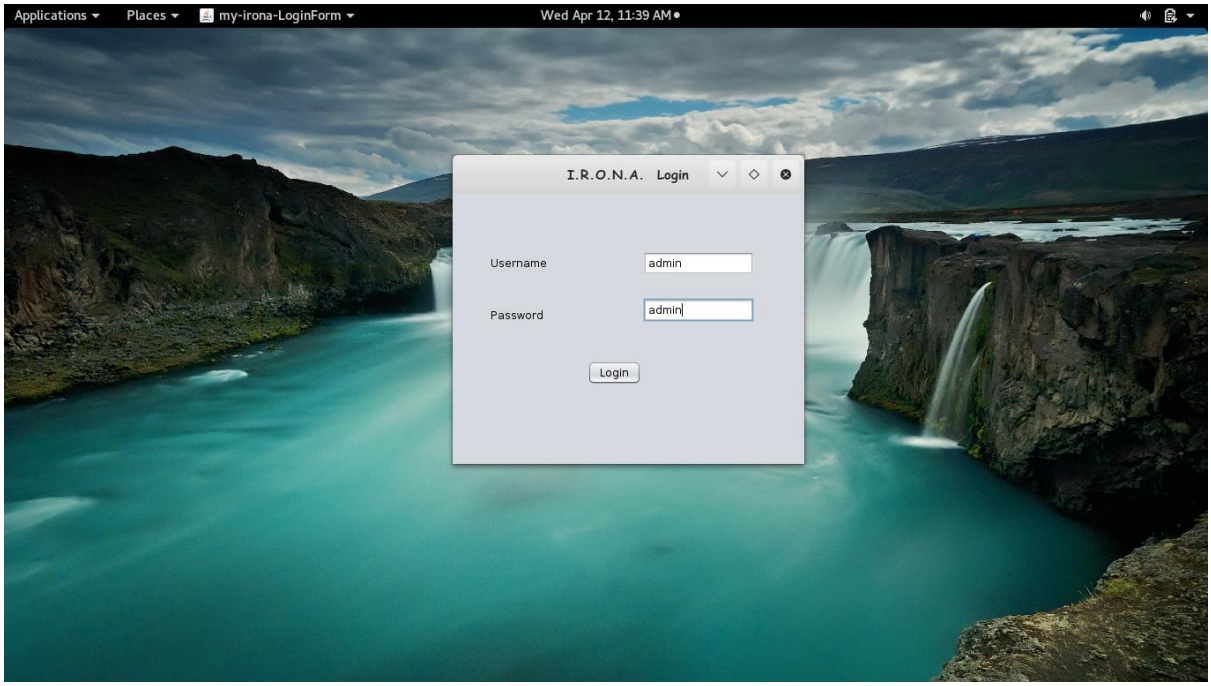
```
            //in ter : nmap 192.168.1.0/24 -n -sP | grep report | awk '{print $5}' > /tmp/iplist.txt
            Process p = Runtime.getRuntime().exec("nmap 192.168.1.0/24 -n -sP | grep report | awk '{print $5}' > /tmp/iplist.txt");
        } catch (IOException ex) {
            Logger.getLogger(IronaUI.class.getName()).log(Level.SEVERE, null, ex);
        }
        JOptionPane.showMessageDialog(null, "Network is successfully refreshed", "Network Refresh Done" , JOptionPane.INFORMATION_MESSAGE);
    }
    else
    {
        System.out.println("Currently this os is not supported.");
            JOptionPane.showMessageDialog(null, "This Operating system is currently not supported", "InfoBox: " + "Unable to refresh the Network", JOptionPane.INFORMATION_MESSAGE);
    }
  }//GEN-LAST:event_jButton5ActionPerformed

  private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton7ActionPerformed
    // TODO add your handling code here:
    jButton7.setFocusable(false);


  }//GEN-LAST:event_jButton7ActionPerformed

  /**
   * @param args the command line arguments
   */
   public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
```

```
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(IronaUI.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(IronaUI.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(IronaUI.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(IronaUI.class.getName()).log(java.util.logging.Level.SE
VERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                new IronaUI().setVisible(true);

            } catch (IOException ex) {
                Logger.getLogger(IronaUI.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    });
```

```
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JComboBox<String> jComboBox1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
// End of variables declaration//GEN-END:variables
private String winFile;
private String linuxFile;
private String winKeyStrokes;
private String linuxKeyStrokes;
public static int recording=0;
private MulticastSocket s;
static String keypath=null;//="/home/student/" + "keystrokes.txt";
```

```
    FileWriter fw;
    long length;
    FileInputStream fis;
    BufferedInputStream bis;
    BufferedOutputStream bos;
    DataInputStream dis;
    DataOutputStream dos;
}
```

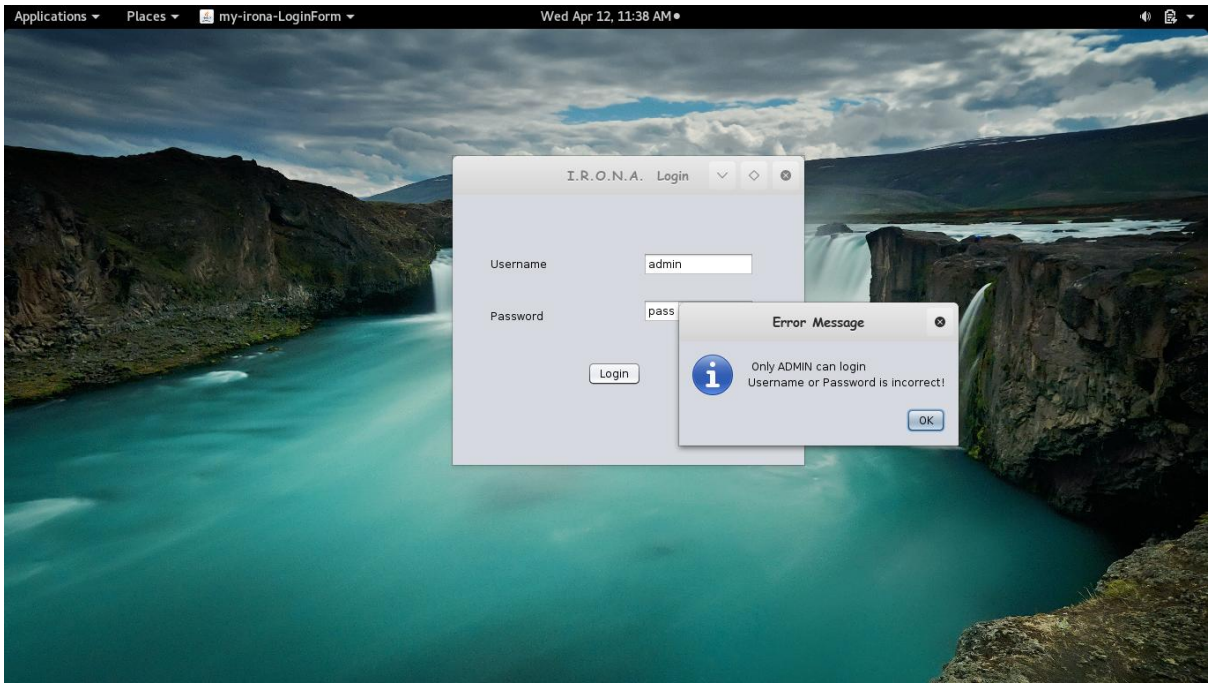
3.3 Screenshots



Screenshot 3.3.1 Login Panel

This is the first UI that will be displayed when the software runs. Only admin is authorized to login with a specific username and password.

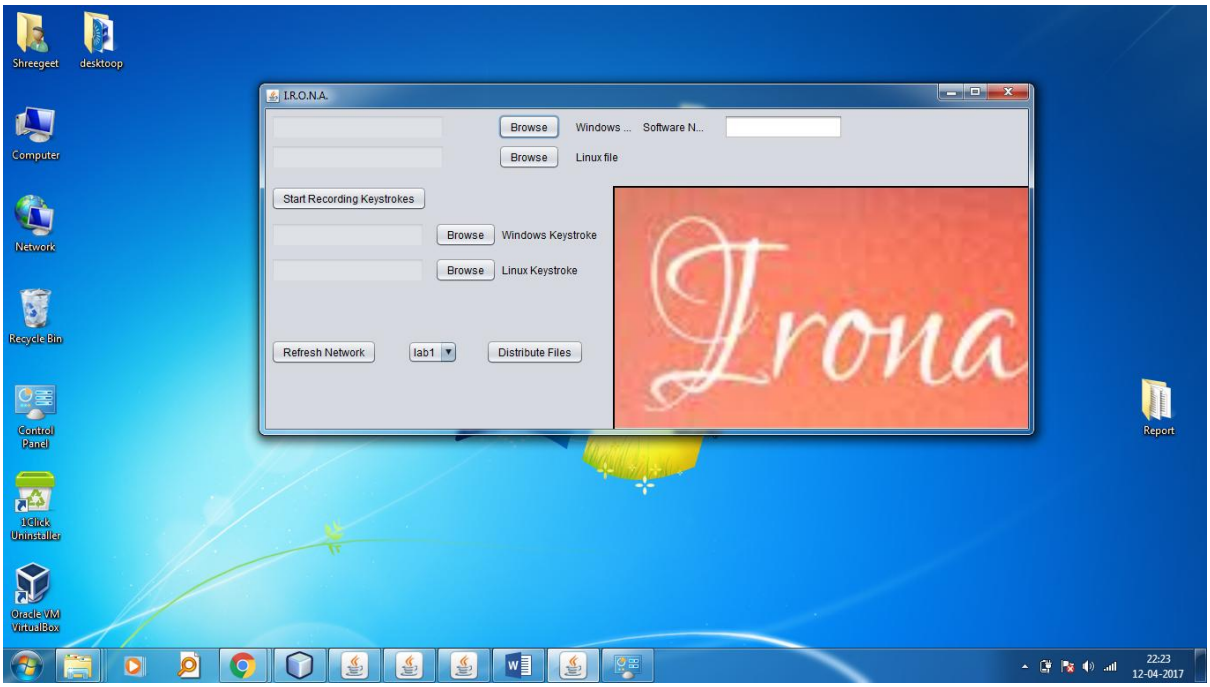
The username and password fields are validated to accept any kind of string which the admin chooses to keep as its credentials. The admin enters the username and password to login to the software for further performing the task



Screenshot 3.3.2 Error Message

An error message is displayed using a pop up alert box in case the user enters a wrong username or password.

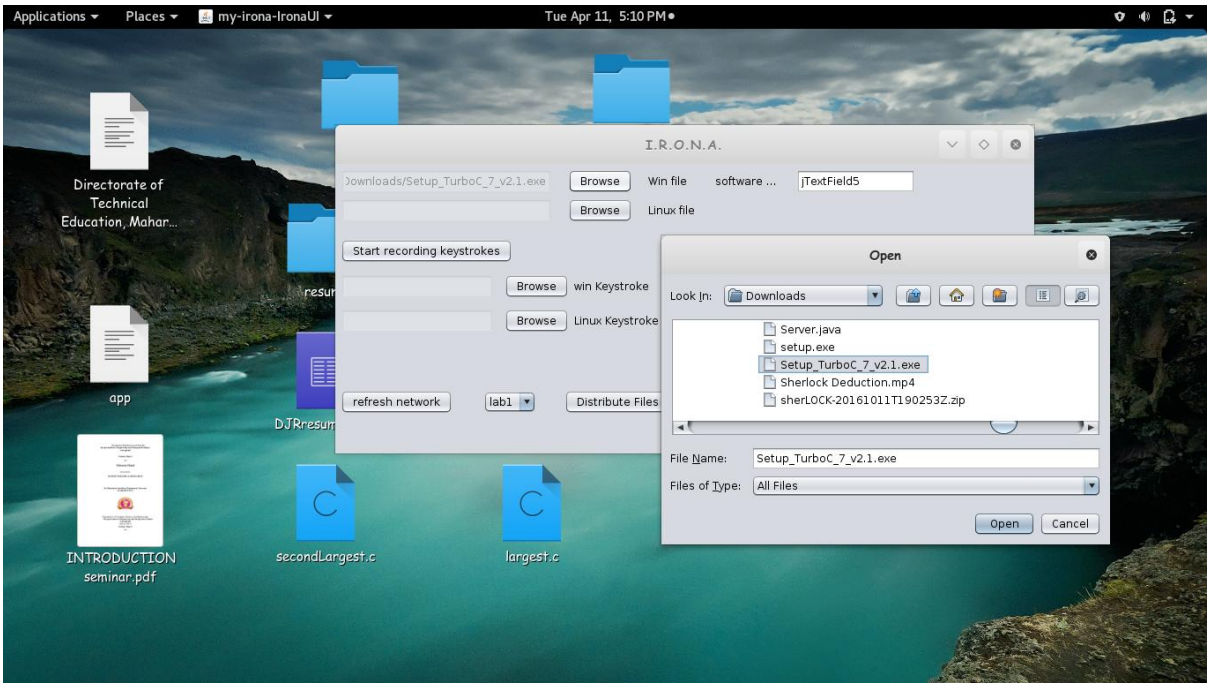
If the user enters a wrong username or password the login panel shows the error message and requests the user to enter a valid username and password.



Screenshot 3.3.3 Execution panel

This is a panel where the user is provided with the components to perform some specific task using the UI.

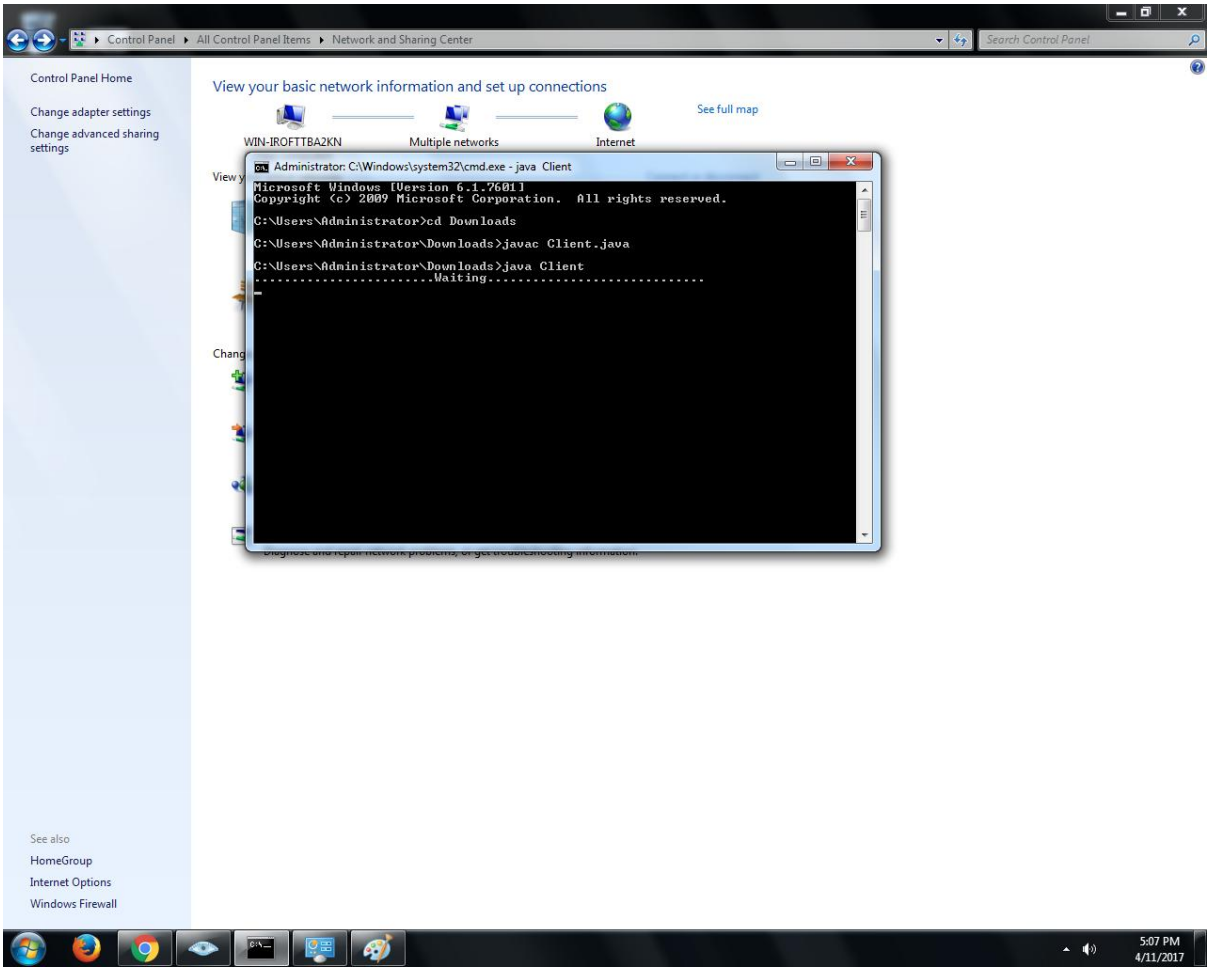
If the user enters a correct username and password then this panel appears to the user. Using this panel the admin can perform different tasks such as browsing the executable file, sending the file to the client.



Screenshot 3.3.4 File Browser

A File browser is given to the user from which the user can choose the particular file which is needed to be sent to the client.

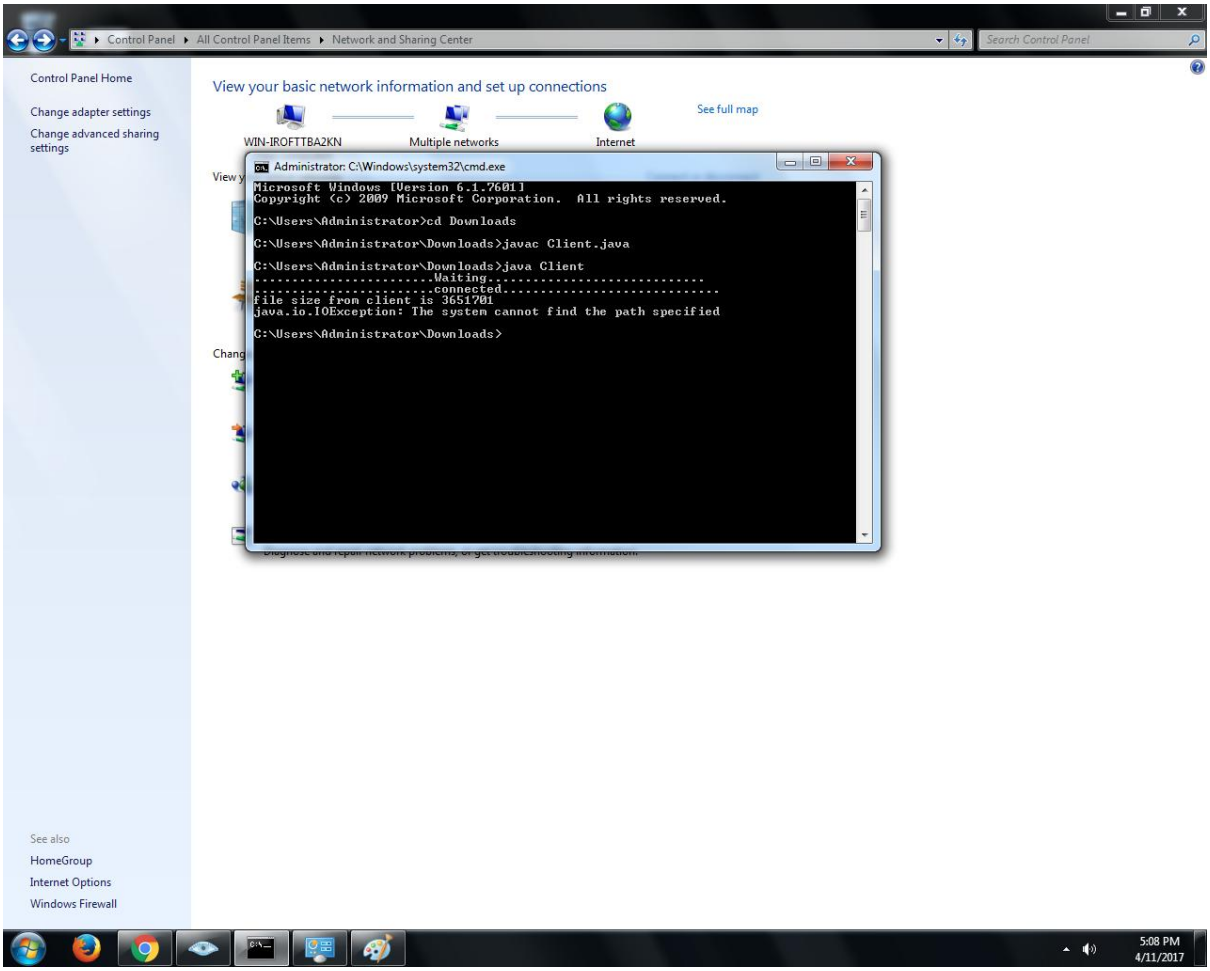
We are providing the admin with the file browser which eases the effort of selecting the file from the file system. The file path is displayed in the text field when the admin selects a file from the file system using the file browser.



Screenshot 3.3.5 Client is waiting to receive the file

Here the client is waiting to receive any file which is sent by the server or we can say the admin of the software.

For the client side we have developed a ever running software which runs in background and is ready to receive whatever file is being sent by the server. At all time it keeps checking if the server is trying to send any file to the client.



Screenshot 3.3.6 Client receives the file

The client receives the file and automatically installs the file to the local machine. If the server send the file to the client then the client displays the connection with the server and the file size that is being received.

Once the file is fully received then it automatically installs the software on the local machine. The setup file that is received is saved in the local machine as transfer.exe. Each time a new software is received then the previous setup is overwritten.

4. PERFORMANCE ANALYSIS

4.1 Testing

Testing is asset of activities that can be planned in advanced and conducted systematically. For this reason a template for software testing a set into which we can specific test case design techniques and testing methods should be defined for the software process. A strategy for software testing must accommodate low-level test that are necessary to verify that a small source code segment can be correctly implemented as well as high –level tests that validate major system functions against customer requirements. Software testing is a verification and validation process.

Verification: Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation: Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Types of Testing

1. **Alpha Testing: -**Testing after code is mostly complete or contains most of the functional and prior to end user being involved. More often this testing will be performed in house or by an outside testing firm in close cooperation with the software engineering department.
2. **Beta Testing: -** Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is released.
3. **Functional Testing: -** Testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the modules performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed do.
4. **Configuration Testing:** -Testing to determine how well the product works with a broad of the hardware/peripheral equipment configurations as on the different operating systems and software.

5. **Pilot Testing: -**Testing that involves the users just before actual release to ensure that users become familiar with the release contents and ultimately accept it. Typically involves many users, is conducted over a short period of time and is tightly controlled.
6. **System Integration Testing**: -Testing a specific hardware/software installation. This is typically performed on a COTS system or any other system comprised or the disparate parts where custom configurations and /or unique installation are the norm .
7. **Software Testing: -**The process of exercising software is with the intent of ensuring that the software system meets its requirements and the user expectations and doesn't file in an unacceptable manner .
8. **Security testing: -**Testing of database and network software in order to keep company data and resources from mistaken/ accidental users, hackers and other malevolent attackers.
9. **Installation Testing: -** Testing with the intent of determining if the product will install on a variety of platforms and how easily it installs .
10. **Compatibility Testing: -**Testing used to determine whether other system software components such as browsers, utilities and competing software would conflict with the software being tested.

Unit Testing

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.)Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

Interface testing:

The practice of component interface testing can be used to check the handling of data passed between various units, or subsystem components, beyond full integration testing between those units. The data being passed can be considered as "message packets" and the range or

data types can be checked, for data generated from one unit, and tested for validity before being passed into another unit. One option for interface testing is to keep a separate log file of data items being passed, often with a timestamp logged to allow analysis of thousands of cases of data passed between units for days or weeks. Tests can include checking the handling of some extreme data values while other interface variables are passed as normal values. Unusual data values in an interface can help explain unexpected performance in the next unit. Component interface testing is a variation of black-box testing, with the focus on the data values beyond just the related actions of a subsystem component.

Unit Testing Advantages:

- Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.
- Reduces Cost of Testing as defects are captured in very early phase.
- Improves design and allows better refactoring of code.
- Unit Tests, when integrated with build gives the quality of the build as well.

White Box Testing

White-box testing (also known as **clear box testing**, **glass box testing**, **transparent box testing**, and **structural testing**) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

White-box testing is a method of testing the application at the level of the source code. These test cases are derived through the use of the design techniques mentioned above: control flow testing, data flow testing, branch testing, path testing, statement coverage and decision coverage as well as modified condition/decision coverage. White-box testing is the use of these techniques as guidelines to create an error free environment by examining any fragile code. These White-box testing techniques are the building blocks of white-box testing, whose essence is the careful testing of the application at the source code level to prevent any hidden errors later on. These different techniques exercise every visible path of the source code to minimize errors and create an error-free environment. The whole point of white-box testing is the ability to know which line of the code is being executed and being able to identify what the correct output should be White-box testing is one of the two biggest testing methodologies used today. It has several major advantages:

1. Side effects of having the knowledge of the source code is beneficial to thorough testing.
2. Optimization of code by revealing hidden errors and being able to remove these possible defects.
3. Gives the programmer introspection because developers carefully describe any new implementation.
4. Provides traceability of tests from the source, allowing future changes to the software to be easily captured in changes to the tests.
5. White box tests are easy to automate.
6. White box testing give clear, engineering-based, rules for when to stop testing.

Black Box Testing:

Black Box Testing, also known as Behavioural Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors

- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

Test procedures:

Specific knowledge of applications code / internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do, but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of how it the software produces the output in first place.

Test Cases

A test case, in software engineering, is a set of conditions under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written - when they are usually collected into test suites.

A test case is a document, which has a set of test data, preconditions, expected results and post-conditions, developed for a particular test scenario in order to verify compliance against a specific requirement. Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution post-condition.

Test case template:

A test case can have the following elements. Note, however, that normally a test management tool is used by companies and the format is determined by the tool used

Table No. 4.1.1 : Format of Test Case

Test ID	The ID of the test case
Objectives	Aim of the test case
Steps	Procedure to go for test case
Expected Output	Expected result after running application
Actual Output	Actual result obtained after running application
Pass/Fail Criteria	Final result in terms of Test passed or failed

4.2 Test Cases

4.2.1 System Testing

The table below summarizes the results of system testing:

Table No. 4.2.1.1: System Testing Table

Test Case ID	Date Tested	Tester	Pass/Fail	Severity of Defect	Summary of Defect	Closed prior to Production Release?	Comments
1	22/03/17	Durgesh	Pass	Low	-	Yes	Works for limited scope

4.2.2 User Acceptance Testing

The table below summarizes the test cases employed for user acceptance testing and the test results obtained for each test case:

Table No. 4.2.1.2: User Acceptance Testing Table

Test Case ID	Date Tested	Tester	Pass/Fail	Severity of Defect	Summary of Defect	Closed prior to Production Release?	Comments
2	21/03/17	Shreegeet	Pass	Low	-	Yes	Accepts any input files

4.2.3 Regression Testing

The table below summarizes the test cases employed for regression testing and the test results obtained for each test case:

Table No. 4.2.1.3: Regression Testing Table

Test Case ID	Date Tested	Tester	Pass/Fail	Severity of Defect	Summary of Defect	Closed prior to Production Release?	Comments
3	20/03/17	Vishal	Pass	Low	-	Yes	-

4.2.4 Performance Testing

The table below summarizes the test cases employed for performance testing and the test results obtained for each test case:

Table No. 4.2.1.4: Performance Testing Table

Test Case ID	Date Tested	Tester	Pass/Fail	Severity of Defect	Summary of Defect	Closed prior to Production Release?	Comments
4	19/03/17	Vishal	Pass	Low	-	Yes	-

4.2.5 GUI Testing

The table below summarizes the test cases employed for user acceptance testing and the test results obtained for each test case:

Table No. 4.2.1.5: GUI Testing Table

Test Case ID	Date Tested	Tester	Pass/Fail	Severity of Defect	Summary of Defect	Closed prior to Production Release?	Comments
5	21/03/17	Shreegeet	Pass	Low	-	Yes	-

4.2.6 Unit Testing

The table below summarizes the test cases employed for user acceptance testing and the test results obtained for each test case:

Table No. 4.2.1.6: Unit Testing Table

Test Case ID	Date Tested	Tester	Pass/Fail	Severity of Defect	Summary of Defect	Closed prior to Production Release?	Comments
6	21/03/17	Shreegeet	Pass	Low	-	Yes	-

4.2.7 Integration Testing

The table below summarizes the test cases employed for user acceptance testing and the test results obtained for each test case:

Table No. 4.2.1.7: Integration Testing Table

Test Case ID	Date Tested	Tester	Pass/Fail	Severity of Defect	Summary of Defect	Closed prior to Production Release?	Comments
7	21/03/17	Shreegeet	Pass	Low	-	Yes	-

5. CONCLUSION

5.1 Conclusion

This chapter summarizes the main success of this research work and discusses an about future research work to achieve the ultimate goal in the field of performance of web accessibility, web security, load balancing and collective intelligence. An in depth literature survey was carried out and the critical analysis of the same raised the following major shortcomings and challenges in the web-centric query optimization techniques.

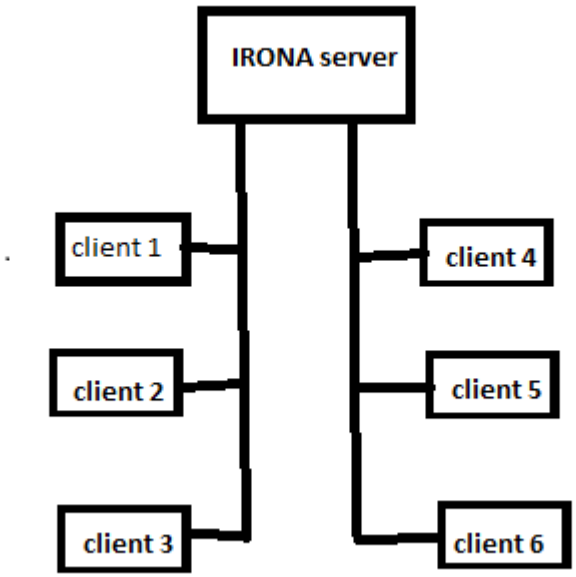


Figure 5.1.1: Architecture of IRONA

An I.R.O.N.A server is present in a network which is responsible to install any software in the infrastructure in the network.

5.2 Future scope

The future of I.R.O.N.A.

In earlier days we used to install any software in our corporate infrastructure manually, but today this is not the case. In this project we have implemented it so that any software can be install on any system using its IP address remotely. We require that the client needs to be running our client software. Later in the future, this project can be used to install a software on multiple systems simultaneously.

The socket Services Interoperability presents an effective means where by existing, perhaps loosely defined, system functionality can be adapted to operate in a web services paradigm. Through the use of Service Delegates, the details associated with directly interfacing with local system functionality are encapsulated and effectively isolated from reusable framework components. Semantic web presents design incorporates technologies such as inference engines, rule-based systems, web services and service-oriented architectures to provide the needed infrastructure to support meaningful interoperability among context-based systems. In order to facilitate the interoperability, has developed a set of technologies, standards, and interface protocols, for interoperability of data, information, and systems over the web. The web service technology and standards are widely accepted and used by the for interoperability among grid systems. The solution to interoperability presented in the research goes beyond traditional web services architectures by supporting the representational disparity typically exhibited by context-oriented systems. Rather than constraining interoperating systems to common representations, the interoperability bridge provides a mechanism for managing the potentially complex representational translation between interoperating systems.

The main contribution of this work is the introduction and use of Network service to achieve interoperability in a heterogeneous environment. The implementation of it already been carried out. Java acts as a distributed middleware to facilitate the interoperability of the whole system with the support of distributed technologies. An overall assessment was also executed to test how well to address the interoperability problem with the support of cross platform software. This assessment consists of testing on applications developed in diverse programming languages, system platforms and database system

Proposed Interoperability Architecture: The Interoperability View in order to capture specific reusability of communication protocol among systems, subsystems and/or components, avoiding the use of a Data transformation. The Interoperability takes advantage of well defined system use cases. The choice of activating services depends on the use case that the invoker's event needs. The framework focuses on the logical and Interoperability Architectural views. The proposed approach is built on the following insights i. Monitoring the correct deployment of system requirements; ii. Offering an effective interoperability capability iii. iii. easily applied to other components, subsystems or systems; and iv. Providing self-adaptive services at run-time according to received external events. The framework architecture is structured in three modular views. They represent the entire system based on different concern areas, which are:

i. Logical (functionalities),

ii. Interoperability (reusability)

iii. Technical (infrastructure / feasibility).

REFERENCES

[1] Network Automation http://www.networkautomation.com/

[2] Ansible https://www.ansible.com/network-automation

[3] http://searchnetworking.techtarget.com/Networkautomation-tools-Should-you-build-or-buy

[4] Application Server https://en.wikipedia.org/wiki/Application_server

[5] http://www.javaworld.com/article/2077354/learn-java/app-server-web-server-what-s-the-difference.html

[6] Application Server in JAVA https://www.youtube.com/watch?v=DhYPP221iVw

Socket Programming https://www.tutorialspoint.com/java/java_networking.html

ACKNOWLEDGEMENT

We would like to place on record our deep sense of gratitude to **Prof. S. B. Kalyankar,** HOD-Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for his generous guidance, help and useful suggestions.

We express our sincere gratitude to **Prof. P. M. Katkar**, Dept. of Computer Science and Engineering, Deogiri Institute of Engineering and management Studies Aurangabad, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We are extremely thankful to **Dr. Ulhas D. Shiurkar,** Director, Deogiri Institute of Engineering and management Studies Aurangabad, for providing me infrastructural facilities to work in, without which this work would not have been possible.

Name of Students **Signature(s)**

1. Vishal Panchal
2. Durgesh Rathod
3. Shreegeet Dixit