MSc. Computer Science

Project

PROJECT DOCUMENTATION

Project Title:

Multi Agent Based Framework for Monitoring Servers, Server Room Environment and Devices.

BY:-

Peter Karanja Kahuthia

P58/61720/2010

SUPERVISOR:

Mr. Lawrence Muchemi

A Research Project proposal submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

DECLARATION

This research project, presented on this report is my original work and to my knowledge has not been presented for any other University Award.

Signed: ……………………………………………….. Date……………….

Kahuthia Peter Karanja - P58/61720/2010

This project has been submitted as part of fulfillment of the requirements for the award of Master of Science in Computer Science of the School of Computing and Informatics of the University of Nairobi, with my Approval as the University Supervisor.

Signed: ……………………………………………….. Date……………….

Mr. Lawrence Muchemi

ACKNOWLEDGEMENTS

I wish to acknowledge and thank my Almighty God for giving me the strength, time knowledge, and in summary for enabling me complete my project and Masters in general, it's been a tough but work challenge.

Second I wish to appreciate and Dad (Bedan Kahuthia Gachamba), Mother (Lillian Muthoni Kahuthia) and my wife (Sylvia Muthoni Waweru) for their individual and unique contribution towards my completion of this project.

I would also wish to extend my appreciation to my panelist Dr. Opiyo Elisha, Mr. Andrew Mwaura Kahonge, Dr. Christine Ronge and Mr. Lawrence Muchemi for their dedicated guidance, encouragement, and motivation during the research period.

Finally I wish to recognize my supervisor Mr. Lawrence Muchemi for accepting to create time and be my supervisor, guidance on my project

Abstract

Server and server room administration has always been a daunting task for many administrators and mangers. The server room mainly house many servers and computers which need close monitoring in terms of space and uptime. The environment itself has to be just right for the servers to function normally. That is the temperature has to be right and the humidity has to be normal also. When the mentioned concerns are not as required the services uptime and availability are affected.

Many premium and open source solutions seem to have a structural and execute in a serial way, this has it downside especially with the emergence of new powerful hardware with extended resources like disc space and memory which need to be utilized effectively. The extended use of technology is evident from increase of portable technology like PDA and smartphones thus this necessitates the need for high uptime and service availability and hence the need for effective and efficient monitoring.

In this project we propose an innovative solution for an effective monitoring of server room environment using multi agents. The proposed model uses parallelism technology and background workers in order to achieve effective monitoring. The finding shows that the use of agents will achieve high effective monitoring of devices and server environment and thus increase high uptime and service availability.

Table of Contents

Contents

List of Tables

List of figures

List of Abbreviations

IP: Internet Protocol.

SQL: Structured Query Language.

TCP: Transmission Control Protocol.

MAS: Multi-agent system

LAN: Local Area Network

CPU: Central Processing Unit

IEEE: Institute of Electrical and Electronics Engineers

MBIMSC: Multi-agent based ICT Monitoring System for Consolidated bank

HW: Hardware

SW: Software

OS: Operating System

RRDtool: acronym for **r**ound-**r**obin **d**atabase **tool**

RRD: Round Robin Database

SMTP: Secure mail transfer protocol

1.0 Introduction

A server is a computer or device on a network that manages network resources, [1] to process the needs of the users or other computers on a network. The servers depending on their application can be database servers, application servers, printer servers, domain controllers, mail servers etc. all the mentioned servers need close monitoring to ensure uptime and service availability. Servers are usually housed in a special room called server room or data center with controlled environment.

Server room is a word that is used interchangeably with datacenter and it is a room dedicated for the housing of servers and network devices [2] that are interconnected to host and provide services/resources to users and customers.

Server room environmental conditioning is a major consideration as far as services accesses are concerned. This is due to the sensitivity of computer equipment to heat, humidity and dust, but also the need for very high resilience and failover requirements. Maintaining a stable temperature and humidity within tight tolerances is critical to IT system reliability.

The main problem that have been faced by many for many years is the challenge of close monitoring of the servers' storage space, devices uptime, and server room environmental conditions like temperature, humidity etc. In this project we propose a real time monitoring solution to monitor devices and server room (datacenter) environmental conditions using software agents. The proposed solution sets the threshold of monitored parameters which will be continuously monitored by software agents. The agents will continuously monitor these parameters and generate and sent alerts for any deviations of the set parameters. These alerts are either sent to system administrator to take action, or the agent can be configured to proactively take any corrective action.

For example, in the case of server room environment monitoring an Ethernet thermometer will be installed and the software agent will be continuously monitoring the temperature in the server room. When the temperature goes above the configured limit, alerts will be generated and automatically sent to designated people.

The developed servers and server room administrative assistant agent will be able to satisfactory monitor the devices and be able to monitor the server room environment. The sensors that will be installed in the server room will monitor the environment (temperature) and give output to the agents. The significance of this system is that there will be an automated notification system through alerts. This will eventually lead to reduced system downtime occasioned and hence less risks, less financial risk, reputation risk, stress free workers and a conducive environment for the servers.

1.1 Background

The responsibility of active monitoring is crucial and mandatory in today's world due to the fact that technological advances and also due to changing customer expectations and needs which are continuously evolving. Ends users expect continuous, uninterrupted service time of critical IT systems. The industry acceptable level of High service uptime and reliability should be increased to the industry acceptable level of 99.9% uptime. This has become a benchmark for any acceptance of service and also a prequalification category of any service provider.

System uptime and monitoring automation cannot be achieved by convectional monitoring and use of agents technology need to be deployed in order to optimize monitoring and achieve 99.9% efficiency on monitoring if not more.

We will base our study on a local commercial bank, Consolidated Bank of Kenya as our case study. Consolidated Bank of Kenya Limited was incorporated on 7th December, 1989. This was in an effort to stabilize the financial sector through the acquisition of nine insolvent institutions and thereafter restructuring them into a viable, professionally run commercial bank.

The bank offers one of the widest range of banking products and services in the market today. We realize that a growing business demands a lot of time and energy. We understand these challenges and continuously develop flexible, innovative and convenient financial solutions to help our customers achieve personal and business success.

In order for the bank to achieve this, they have established customer service goals, one of them is providing customer services, Monday to Friday from 9.00am to 4.00pm in most branches and offices. In addition, they aim to provide 24-hour, 7-day access to internet and telephone banking and a network of Automatic Teller Machines.

In order for the systems to be accessible by the customers the bank need to have active monitoring system which should also have the ability to resolve identified issues automatically and also this can only be made possible by having resilient ICT system that is not prone to frequent downtimes from factors that can be prevented or reduced.

Major factors that affect customers are prolonged downtime leading to service unavailability. This leads to various risks e.g. reputation risk, business loss, financial loss etc. We will delve into these details later in the project.

Many organizations have deployed convectional agent less system that use scripts and logs to monitor uptime, example of this systems are Cacti, Nagios and Prisim. These are mainly used system because of the wide capability, application and the fact that they are open source.

1.2 Problem statement

Many mission critical systems are hosted in servers which are housed in a server room) data center). For such systems, lack of availability, reliability, or responsiveness can lead to extensive losses. It is a daunting task for system administrators to manually keep track and monitor both the environmental conditions of datacenter and many devices all the time in order to search anomaly root causes and correct performance anomalies. Therefore, it is imperative to develop automated monitoring system to detect and report any anomaly on real basis to achieve smooth function of systems with a minimum requirement for human intervention.

1.3 Purpose of the project

The purpose of this project is to review the shortcoming of the existing solutions available and proposed an agent based framework that employs agents in monitoring the server room (data center) environment and other devices conditions such as disk space etc. The proposed systems will have the alert system which issues alerts whenever an anomalous event occurs. We will use Consolidated Bank server room (data center) as our case study. The overall aim is to minimize systems downtime by actively monitoring potential threats and making corrective actions and sending notification in case of any fault. This will eventually lead to effective and efficient way monitoring of server room (data center).

1.4 General Objectives

The overall objective of this project is to develop an effective and efficient agent base data center monitoring system that will be able to proactively monitor data center environmental conditions and devices and be able to generate alerts(email and sms) whenever an anomalous even is detected.

1.5 Specific Objectives

1. To develop an agent based framework which actively monitors data center environmental conditions and issues alerts whenever an anomalous condition is detected.
2. To research from the literature the various systems used in monitoring data centers (server rooms).
3. To identify the limitations of existing data center monitoring solutions.
4. To research on how a multi-agent system can help in implementing such solutions.
5. To test and evaluate the performance and effectiveness of the agent based system using Consolidated Bank data center as our case study.

1.6 Research questions

(1) How can we monitor the data center (server room) and react to anomalous conditions from remote locations in a rapid manner?

(2) How can multi agents be used to monitor and alerts systems administrators remotely?

(3) How does our proposed system performance compare with other existing solutions?

(4) What's the best solution for monitoring of temperature & humidity for data center that has smtp feature for notifying by Email?

1.7 Significance of the study

One of the most critical tasks in any data center is that of environmental conditions and device/ equipment monitoring, the servers, and the data center itself. There is quite a bit of planning that goes into choosing an effective monitoring solution and the ongoing management of the equipment within the data center. In addition to traditional IT challenges such as server management, security, and performance, data center owners now must deal with systems uptime of mission critical systems. Effective and efficient monitoring system will prove useful in meeting this critical objective.

Terms

Datacenter – Applicable in monitoring servers and services.

Server room Environment – Ability to monitor servers, devices like routers and server room environment.

Automation – Be programmable to autonomously act to resolved issues

Notification – The agent will able to send email, sms, and sound notification to system administrators.

Reduce trouble shooting – In case of large corporation it will able to pin point the faulty and hence reduce the time used to identify fault.

Give chance to prevent occurrence of issues – This is where due to early notification one can call for the contractors to attend to the issue being reported by the agent before the incident occurs.

1.8 Definition of Terms

Server room – a room that houses mainly computer servers [2]

Data Center – a facility used to house computer systems and associated components, such as telecommunications and storage systems. It generally includes redundant or backup power supplies, redundant data communications connections, environmental controls (e.g., air conditioning, fire suppression) and security devices. [3]

USB modem – wireless GSM modem is a physical unit that enables mobile Internet access and sending and receiving sms by connecting to the mobile operators' towers. [5]

ICT – Information technology

MAS – Multi Agent System

Task parallelism – (also known as **function parallelism** and **control parallelism**) is a form of parallelization of computer code across multiple processors in parallel computing environments

1.9 Assumptions of the Research

For the purpose of this study we have limited our project do the scope that system downtime can only be limited to disc space depletion, server room environment and finally device unavailability.

Disc space check will also be limited to servers based on Linux for the connection protocol that will be used to check disc space will be SSH, server room monitoring will be limited to temperature and humidity and dew point and hardware that will be used will be TH2E from papouch.com.

Device monitoring will be limited to use of ping echo protocol though other methods can be configured and be used to check if device is up and reachable.

2.0 Literature Review and Theory

2.1 Introduction

Agent less system has been used in many monitoring systems and even basic systems are based on such framework. Recently we have seen the changes and increase to system models and architecture. Where system efficiency in use of the system resources is important, we have also seen rise in processing power of computers and this requires systems to be enhanced to ensure optimum use of the hardware to get its optimum value.

This is why we are seeing the use and deployment of agent and multi-agent systems based on this framework or models. This is actively seen on one major popular virtualization environment like VMware.

Virtualization and use of virtualized systems are on the increased and this can inversely project on the use of multi-gent somewhat if not 100%, this is evident from the Gartner report [1] showing the number of percentage in workload on virtual machine see figure 1 below.

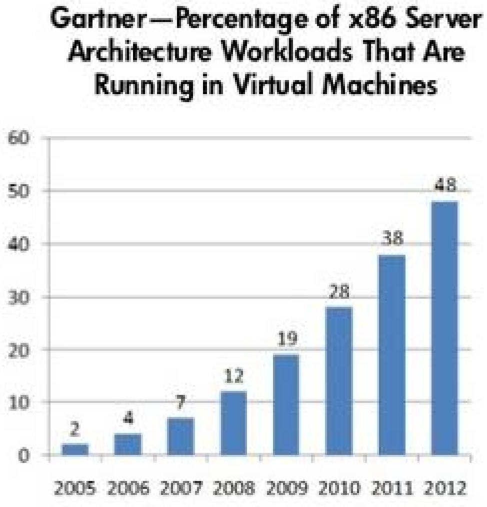


Figure 1: Architecture Workloads that are running in virtual machines

In the past few years we have seen dramatic customer behavioral change in use of technology due to advance portability of technology. And in the case of transaction increase over the years we find that system use over 24/7 has increased and thus the need to active monitor system to avoid downtime.

All monitoring system needs some piece of software to collect data (probe), this data can be got from live probing and reading variables or by reading log files already created by the device being monitored. The collected data need also to be analyzed and compared against acceptable set values and various decisions made and the action taken. Some of the actions that can be done are, resolving the identified issue, sending a notification about the identified problem.

These kinds of monitoring need a dedicated system to enable effective monitoring. We will be discussing various existing system that have been widely used and their technology/ framework and how effective they are, we will also look at the shortcoming and downsides on the same and also listing how my approach on using the agents will help analyze the data and. One of them is munin you configure a server that triggers scripts on the monitored servers. The set of data sources is controlled by what you install in the correct directory on the monitored server. Another one which is the best and widely used is called Cacti (Open source) and relies on SNMP heavily and also allows you to write plugins which then connect out from the central Cacti server to the individual hosts to be monitored. One also doesn’t have to do any SNMP but can report data in a different protocol which is a plus.

Munin is an agent based system as my definition goes. The code necessary to monitor services is not in a central place this employs the agent attributes of autonomy. For Nagios and Cacti it appears to be “true” agent-less, but effectively one program amount of code on the client side as well which will take quite some time.

Before developing the server devices and integrating server room monitoring sensor, work flow and all the planning process need to be done. Work planning is the important thing to make sure the project successfully done. Work planning also will organize activities and coordinate what need to in one project. A Gantt chart will be used to control process during develop this project. A good selection of methodology will give effects to developers to produce a quality project Methodology.

We have decided to research on the below monitoring system which are widely used and have used agentless technology.

2.2 Structure of an agent

It is possible to define the structure of a very simple agent program mathematically as an agent function [2] which maps every possible percepts sequence into a possible action the agent can perform or to a coefficient, feedback element, function or constant that affects eventual actions:

$$f : P^* \rightarrow A$$

Agent function is an abstract concept as it could incorporate various principles of decision making like calculation of utility of individual options, deduction over logic rules, fuzzy logic, etc. [3]

2.3 Agent modeling

2.3.1 Simple reflex agent model

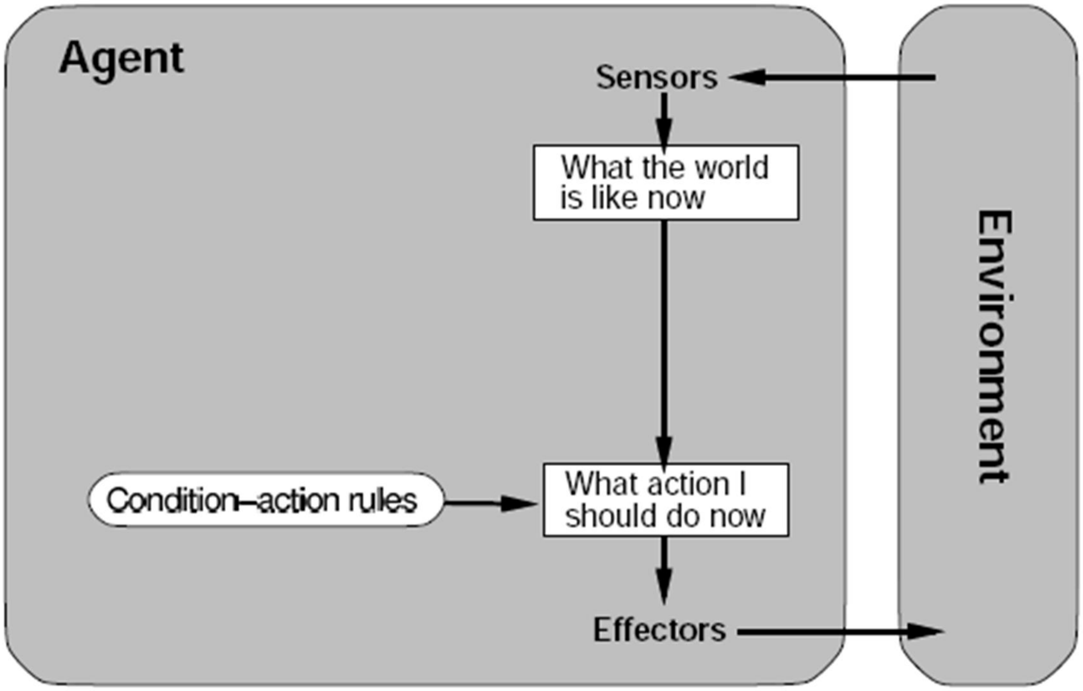


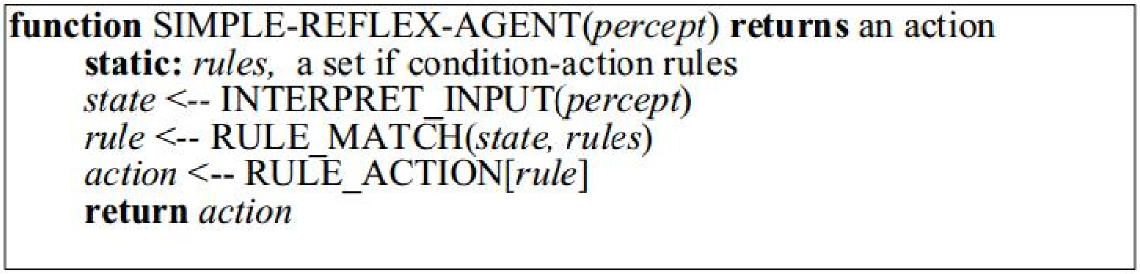
Figure 2: Simple reflex agent model

With the simple reflex agent the agent reaction or action is based on the current perception of the agent, or the world and not depending on the past percepts.
The simple reflex agent has a library of such rules so that if a certain situation should arise and it is in the set of Condition-action rules the agent will know how to react with minimal reasoning.
An example of condition action rule is
If *condition* **the** *action*

A practical example is

If hand is in fire **then** pull away hand

Simple reflex agents are simple to work with but have very limited intelligence
[4]



```
function SIMPLE-REFLEX-AGENT(percept) returns an action
      static: rules,  a set if condition-action rules
      state <-- INTERPRET_INPUT(percept)
      rule <-- RULE_MATCH(state, rules)
      action <-- RULE_ACTION[rule]
      return action
```

2.3.2 Goal-based agent Model

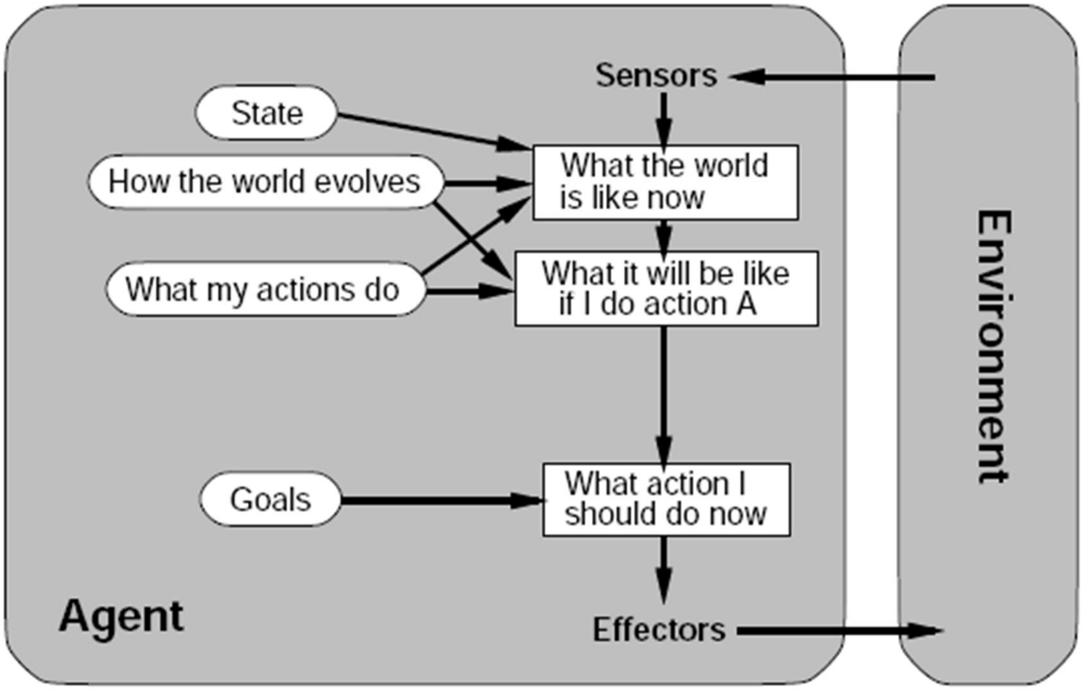

Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Environment
Goals
What action I should do now
Agent
Effectors

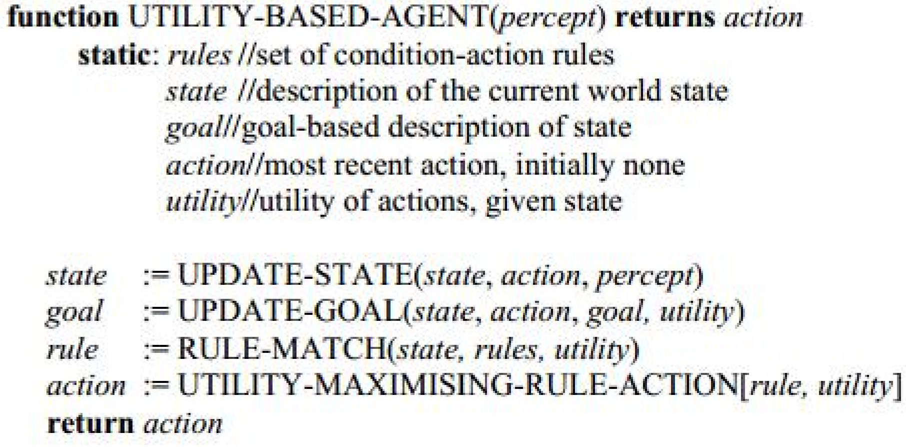
Figure 3: Goal-based agent model

We can describe Goal-based agents to be able to further expand on the capabilities of the model-based agents, by using designed final state "goal" information. In our case goal information could best describe a situation that is or are desirable. This allows the agent a way to choose among multiple possibilities, actions selecting the one which reaches a preset goal state.

We consider an intelligent agent to have a set of goals with desirable situations which should be achieved. The agent can use these goals together with a set of actions and their predicted outcomes to see which action(s) achieve our goal(s).

In order to achieve the goals it can take 1 action or many actions. Search and planning are two subfields in AI devoted to finding sequences of actions to achieve agent’s goals. [4]

An example of pseudo-code is



```
function UTILITY-BASED-AGENT(percept) returns action
    static: rules //set of condition-action rules
            state //description of the current world state
            goal//goal-based description of state
            action//most recent action, initially none
            utility//utility of actions, given state

  state   := UPDATE-STATE(state, action, percept)
  goal    := UPDATE-GOAL(state, action, goal, utility)
  rule    := RULE-MATCH(state, rules, utility)
  action  := UTILITY-MAXIMISING-RULE-ACTION[rule, utility]
  return action
```

2.4 Task parallelism

In order for the agents to be efficient on performing its designated tasks effectively we have implemented tasks parallelism to ensure that serial execution of tasks and monitoring is eliminated. This will greatly improve on monitoring time and efficiency.

Potential speed-up of an algorithm on a parallel computing platform is given by Amdahl's law [5] it states that a small portion of the program which cannot be parallelized will limit the overall speed-up available from parallelization. A program solving a large mathematical or engineering problem will typically consist of several parallelizable parts and several non-parallelizable (sequential) parts. If $\alpha$is the fraction of running time a program spends on non-parallelizable parts, then:

$$\lim_{P\to\infty} \frac{1}{\frac{1-\alpha}{P}+\alpha} = \frac{1}{\alpha}$$

Is the maximum speed-up with parallelization of the program? If the sequential portion of a program accounts for 10% of the runtime ($\alpha = 0.1$), we can get no more than a 10× speed-up, regardless of how many processors are added. This puts an upper limit on the usefulness of adding more parallel execution units. ”in our application we only need basic parallelism for the agent to perform designated tasks.

The pseudo code below illustrates task parallelism:

```
Program:
...
If CPU="a" then
  Do task "A"
Else if CPU="b" then
  Do task "B"
End if
...
End program
```

2.4.1 Benefits of Task parallelism

- ✓ Save time and/or money
- ✓ Provide concurrency
- ✓ Reduce Limits to miniaturization

2.4.2 Some General Parallel Terminology

Node - This is a standalone that comprise of multiple CPUs/processors/cores, memory, network interfaces, etc.

Task - A logically discrete section of computational work or instruction.

Synchronization - The coordination of parallel tasks in real time, very often associated with communications. Often implemented by establishing a synchronization point within an application where a task may not proceed further until another task(s) reaches the same or logically equivalent point.

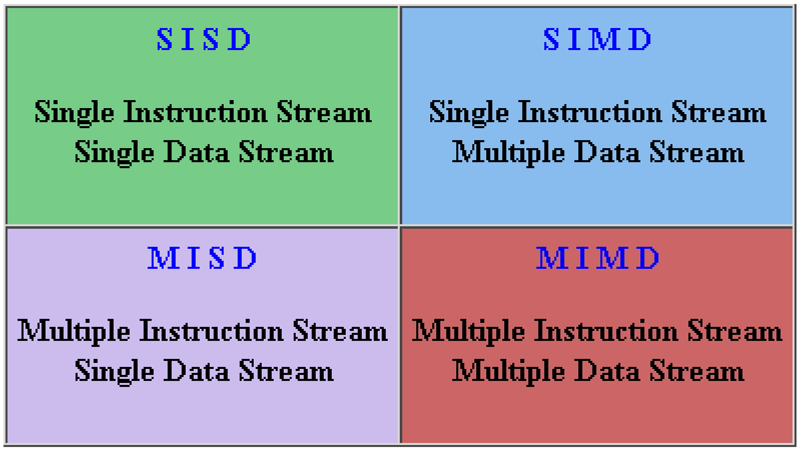
Parallel Overhead - The amount of time required to coordinate parallel tasks, as opposed to doing useful work. The overheads can include:

- Task start-up time
- Synchronizations
- Data communications
- Task termination time

2.5 Flynn's Classical Taxonomy

There are different ways to classify parallel computers one of the mostly widely used classification is the since 1966 is called Flynn's Taxonomy.

Flynn's taxonomy classification distinguishes multi-processor computer architectures by how they can be classified along the two independent dimensions of Instruction Stream and Data Stream. Of which each of these dimensions can have only one of two possible states: Single or Multiple.



SISD Single Instruction Stream Single Data Stream	SIMD Single Instruction Stream Multiple Data Stream
MISD Multiple Instruction Stream Single Data Stream	MIMD Multiple Instruction Stream Multiple Data Stream

Figure 4: Flynn's taxonomy

In our case we will use the SIMD that’s is single instruction stream multiple data stream

2.5.1 Single Instruction, Multiple Data (SIMD)

This is a type of parallel computer

□ Single **Instruction:** All processing units execute the same instruction at any given clock cycle

□ Multiple **Data:** Each processing unit can operate on a different data element

We will have as single instruction stream but have multiple data that will be possibly be executed in parallelism see figure 5 below.

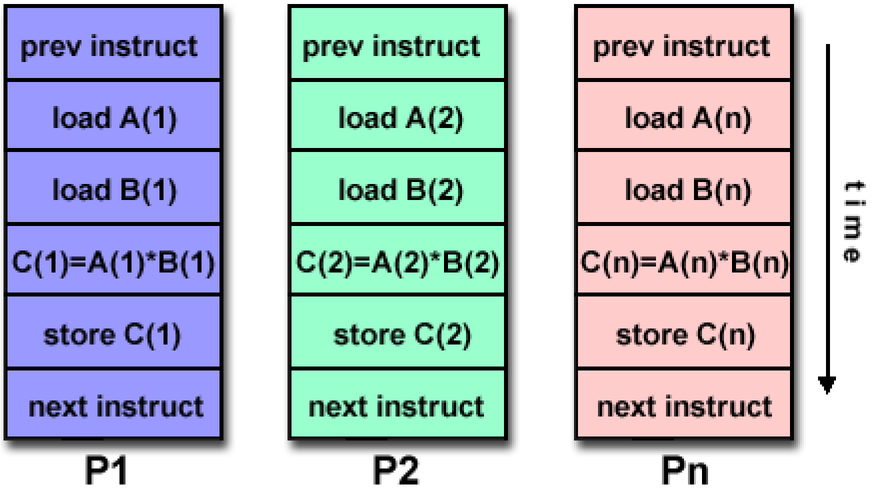


Figure 5: SIMD execution stream

2.6 BackgroundWorker

The use of background worker helps in the clean execution of tasks that would and can take long without affecting the user interface or window form executing the task.

BackgroundWorker class in VB .Net provides an easy way to run time-consuming operations on a window form as a background thread. The BackgroundWorker class enables one to check the state on the task being run and provides even the capability for one to cancel the operation.

2.6.1 Procedure for a BackgroundWorker

i. At the class level, create an instance of the BackgroundWorker class.

ii. Specify whether you want the background operation to allow cancellation and to report progress.

iii. Create an event handler for the background worker's DoWork event.

iv. Create an event handler for the background worker's ProgressChanged event.

v. Create an event handler for the RunWorkerCompleted event.

vi. Add the event handlers to the BackgroundWorker instance's events.

vii. Start running the background operation by calling the RunWorkerAsync method.

viii. Cancel the background operation by calling the CancelAsync method.

2.7 Fact and Finding

2.7.1 Case Study 1: Alchemy Eye Server Monitoring System

This Alchemy Eye Server monitoring System is a product produce by Alchemy lab that advertised in ServerFiles.com which is a server software directory for network administrator and IT professionals. It used Windows as its platform.

Alchemy Eye is a network monitor that continuously monitors network server availability and performance. In the event of network errors, this system can alert the network administrator by mail before the problem get serious out of hand. If servers go down, Alchemy Eye will notify the network administrator or operator in charge and writes a detailed log file.

On any type of server, Alchemy Eye monitors server functions using a variety of protocols and services, including ICMP (ping) network monitoring, NT Event Log computer monitoring, HTTP(s) /FTP URL internet monitor (with or without contents filters). To alert the network administrator about server malfunction events or in the case of a network failure this network monitoring system can send message alert info and window popup.

2.7.2 Case Study 2: WildMetrix 2005 Edition for Server

This WildMetrix 2005 Edition is produce by Ascendview LLC Company. It delivers a server, application and network monitoring solution that provides IT professionals with a tool to monitor entire system infrastructures, pinpoint performance issues and increase productivity. The scalable and expandable design of WildMetrix 2005 Edition allows licensing options that create a solution that can grow as infrastructure monitoring needs to grow. WildMetrix 2005 Edition's architecture provides a tool in a centralized format to monitor and manage geographically separated servers, networks and applications.

This system is using windows as its platform. WildMetrix 2005 Edition also affords the segregation of data while monitoring and administration can be done from an angle console. Pre-built Monitoring Policies deliver results that include preconfigured alert thresholds.

Server Monitoring Sound Notification System (SMSN) planned to be run in a basic network environment not in a separated geographical server. Therefore, it doesn't need features promotes in WildMetrix 2005 which built to adapt to a large location and areas. Status notification can be send through email to network administrator according to network condition.

2.7.3 Case Study 3: HWg-STE Ethernet thermometer

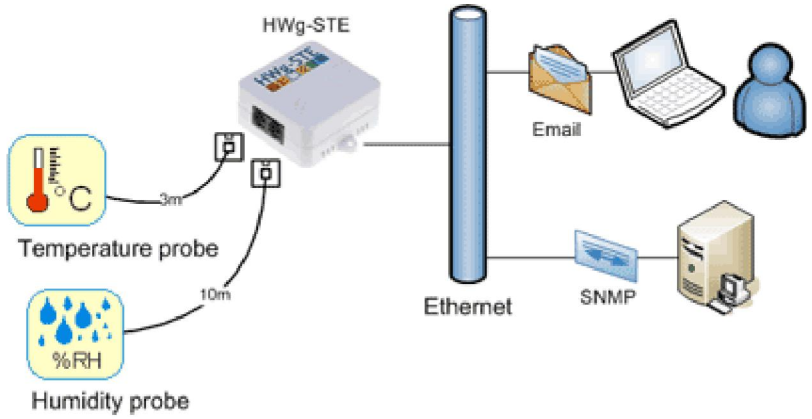


Figure 6: HWg-STE Ethernet thermometer framework

HWg-STE web thermometer is configured using a built-in web server. If the temperature (humidity) exceeds a specified threshold, an e-mail alert is sent. Free windows software HWg-PDMS for graphical overview and MS Excel export included.

- The HWg-STE IP Thermometer is fully SNMP compatible.
- Temperature monitoring, E-mail alert to overheating or temperature drop
- Windows software, AC power adapter and temperature probe included
- Simple installation, supports DHCP

2.7.4 Case Study 4: TH2E - Ethernet thermometer and hygrometer

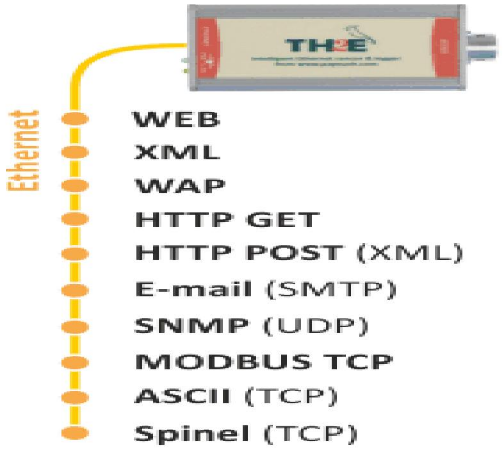


Figure 7: TH2E - Ethernet thermometer and hygrometer

TH2E is a Thermometer and Hygrometer that can calculate a dew point. It can store measured-out values into its internal memory, send the measured-out values to your server, send e-mails when a threshold is exceeded, TH2E remembers extremes and it has an internal WEB server. TH2E is fully configurable via its WEB interface. SNMP and MODBUS TCP are also supported.

2.7.5 Case Study 5: RRDtool Monitoring System

RRDtool is based on open source technology and aims to handle time-series data that's being generated on a host computer/Server like network bandwidth, temperatures, CPU load, etc. The data collected is stored in a round-robin database (circular buffer), thus the system storage footprint remains constant over time.

RRDtool includes tools to extract RRD data in a graphical format, for which it was originally intended. It uses daemon that receives updates to existing RRD files.

The composition of RRDtool daemons include:

Rrdtool - Round Robin Database Tool

Rrdcached - Data caching daemon for rrdtool

Librrd - RRD library functions

Rrdbuild - Instructions for building RRDtool

Rrdcgi - Create web pages containing RRD graphs based on templates

Rrdcreate - Set up a new Round Robin Database

Rrddump - Dump the contents of an RRD to XML format

Rrdfetch - Fetch data from an RRD.

Rrdfirst - Return the date of the first data sample in an RRA within an RRD

Rrdflushcached - Flush the values for a spcific RRD file from memory.

Rrdgraph - Round Robin Database tool graphing functions

rrdgraph_data - Preparing data for graphing in rrdtool graph

rrdgraph_examples - Examples for rrdtool graph

rrdgraph_graph - Rrdtool graph command reference

rrdgraph_libdbi - Fetching data for graphing in rrdtool graph via libdbi

rrdgraph_rpn - About RPN Math in rrdtool graph

Rrdinfo - Extract header information from an RRD

rrdlast - Return the date of the last data sample in an RRD

Rrdlastupdate - Return the most recent update to an RRD

Rrdresize - Alters the size of an RRA and creates a new .rrd file

rrdrestore - Restore the contents of an RRD from its XML dump format

rrdtune - Modify some basic properties of a Round Robin Database

rrdupdate - Store a new set of values into the RRD

rrdxport - Export data in XML format based on data from one or several RRD

2.7.5.1 How Does Rrdtool Work?

Data Acquisition

Data acquisition is done by RRDtool and lets you update the log file at any time you want. RRDtool have the capability to automatically interpolate the value of the data-source (DS) at the latest official time-slot (interval) and write this interpolated value to the log. The original value you have supplied is stored as well and is also taken into account when interpolating the next log entry.

Consolidation

RRDtool offers a solution to the need to query grouped data or data that is related through its data consolidation feature. When setting up a Round Robin Database (RRD), you can define at which interval this consolidation should occur, and what consolidation function (CF) (average, minimum, maximum, total, last) should be used to build the consolidated values, You can define any number of different consolidation setups within one RRD. They will all be maintained on the fly when new data is loaded into the RRD.

Graphing

RRDtool allows you to generate reports in numerical and graphical form based on the data stored in one or several RRDs. The graphing feature is fully configurable.

2.8 Discussed Monitoring solutions shortcoming

Most open source solutions are implemented using structural programming and will execute in a serial way meaning they do not use or adapt to the new hardware that are available which are now more powerful and has many resources.

You will also find that most have plug-ins that can be accepted but it will take an experienced programmer or an administrator to have such a solution running and maintained/supported in day today running.

Most solution lack one of shelf solution that’s integrated to provide a complete monitoring solution that will monitor equipment, services, server room environment and resolving identified issues deployed on agent technology.

2.9 Conclusion

We will develop a solution that will utilize the use of agents’ technology to monitor equipment, services and server room environment and as per my review above and depending with the cases analyzed above there is no complete heuristic solution that addresses heat, humidity monitoring coupled with monitoring of servers, devices and resolving faults that have been identified by the monitoring agents.

The solution will aim to couple system monitoring, faults resolution and temperature and humidity server room environment monitoring. The solution is aimed to improve system uptime for consolidated bank

Monitoring temperature, humidity, server spaces and devices availability is done manual by System administrators and ICT managers. When downtimes are experienced due to these factors they lead to service unavailability resulting to risks and financial loss.

The main issues that affect consolidated bank as my case study are:-

- Air conditioner fails and result to high temperatures which lead to some servers freezing and other shutting down the end result is unavailability of service to customers thus resulting to various risks e.g. operational risk, reputational risk etc.
- Some servers/ routers fail or shutdown this result to unavailability of service.
- Dues to disk space limitations, logs and backup need to be cleaned up and most times this goes undone and thus resulting to system downtime.
- Faulty device sometime take time to identify.
- Do not have a system to monitor the current temperature
- Do not have a specific signal be show if temperature high.
- Waste time to always go to server room to monitoring the temperature.

2.10 Assumptions

- Heat sensor can vary the sensing range smoothly.
- The agent will work 100%
- There is an acceptable temperature range
- In server environment temperature will be the only monitored variable.
- Limit server downtime to available space depletion.
- Server uptime will be limited to ping and echo response though other protocols can be supported.

3.0 Methodology

3.1 Overview

The study involves the developing of a working solution developed using waterfall model and utilizing agents technology, to enhance hardware utilization we will used technology like task parallelization and backgroundworkers using visual basic .NET 2010.

3.2 Research Methodology for Waterfall development model

Waterfall development, also known as the Systems Development Lifecycle Model (SDLC), is an approach to software development that breaks a project into finite phases.

To develop this project, the first phase needed is Planning Phase, this phase focus to collect all the source required, do a work flow target of project that will done. Second phase is analysis, the analysis phase involve the requirement definition and analyzes the problem of monitoring and making corrections. The third phase is research, this phase more focus to research work. The fourth phase is design, in this phase the focus is translate the specification requirement into a set of notations that represent the architecture of probes and agent system. The five phases is implementation of, designs translate into hardware and code. Testing is last phase in this project, to make sure all the functionality is successfully implement as intended.

Figure 1.2 show the phase for developing multi-agent based Monitoring System for Consolidated bank server devices and Server Room

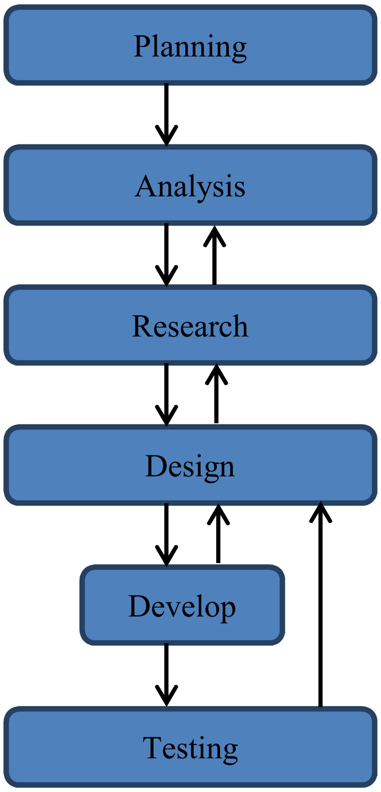


Figure 8: Phase for develop Project

Planning - The process for this phase is:

- Define what resource (hardware, software) need use to development this project
- Decide what tools need use to develop this project
- Plan where to find the resource
- Estimate fully cost to complete this project
- Estimate timeline throughout develop this project
- Planning what expected output from this project.

Analysis - it must analysis all journals the project and focus the whole aspect.

The focus should be on:

- Analysis that user and what situation are suitable to used whenever the project complete.
- Analyze what current problem before and after develop project and what solution to solve that.

- Analysis what function can produce and what benefit if used that.
- Analysis the project it whether can assist and support consolidated bank ICT department.
- Analyze whether the project have value beyond consolidated bank.

Analyze which hardware is easy and fast to integrate and develop.

3.3 System Conceptual Model

The developed system have the following conceptual framework, it basically shows the system overview of what components are composed of the system and how they are interconnected in order to perform monitoring effectively and efficiently.

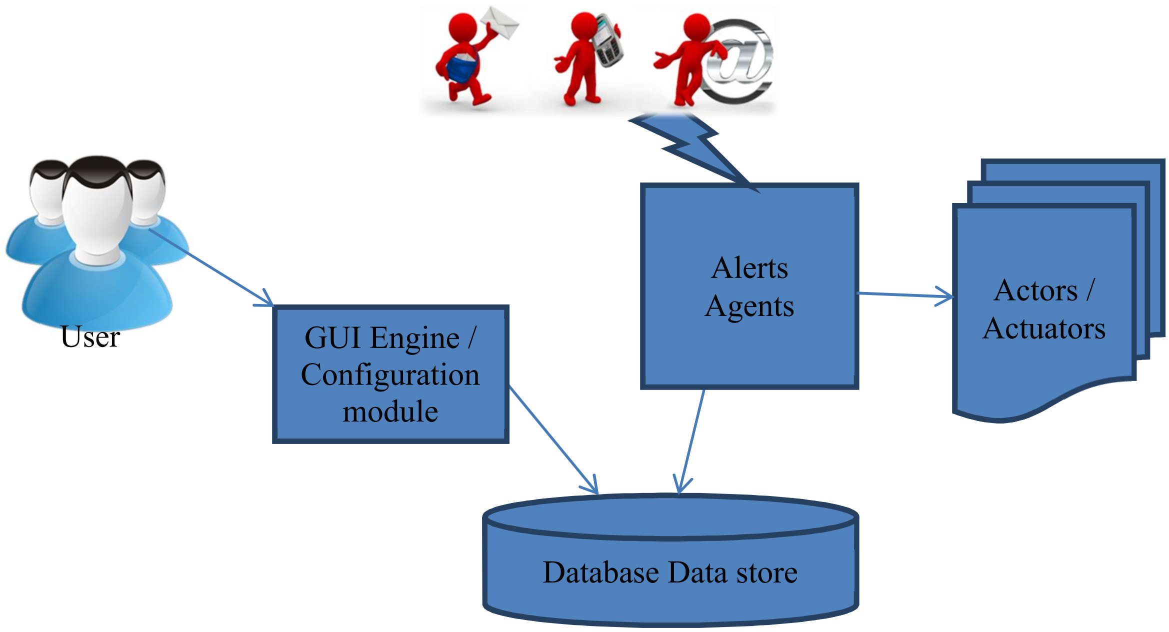


Figure 9: Conceptual model

3.3.1 User

This is the person who has been given access to the system and has the ability to login and create or make changes to the system configuration.

3.3.2 Configuration Module

This is the panel or the heart of the application where all entries are made in that device and system parameters are set using this module.

3.3.3 Agents composition

This can be simplified to be the overall composition of all agents that are composed in the system.

3.3.4 Database

All data that is collected and entered through configured through the configuration module all of it is stored in the database, which can be any relation database buy in our case we will be using MySQL database.

3.3.5 Hardware

Hardware will be considered to be any physical device that the agents will be interaction with, to get a reading or check status of the device.

3.3.6 Alerts Channel

Alerts channel can be all available channels that the agents will use to communicate or delver alerts to users, incase on trouble or notification.

3.4 Agents Interactions Model

Conceptual Model of the agent interactions is like this and basically shows in brief agents interactions see figure 10, each agent's definition and functionality has been described below.

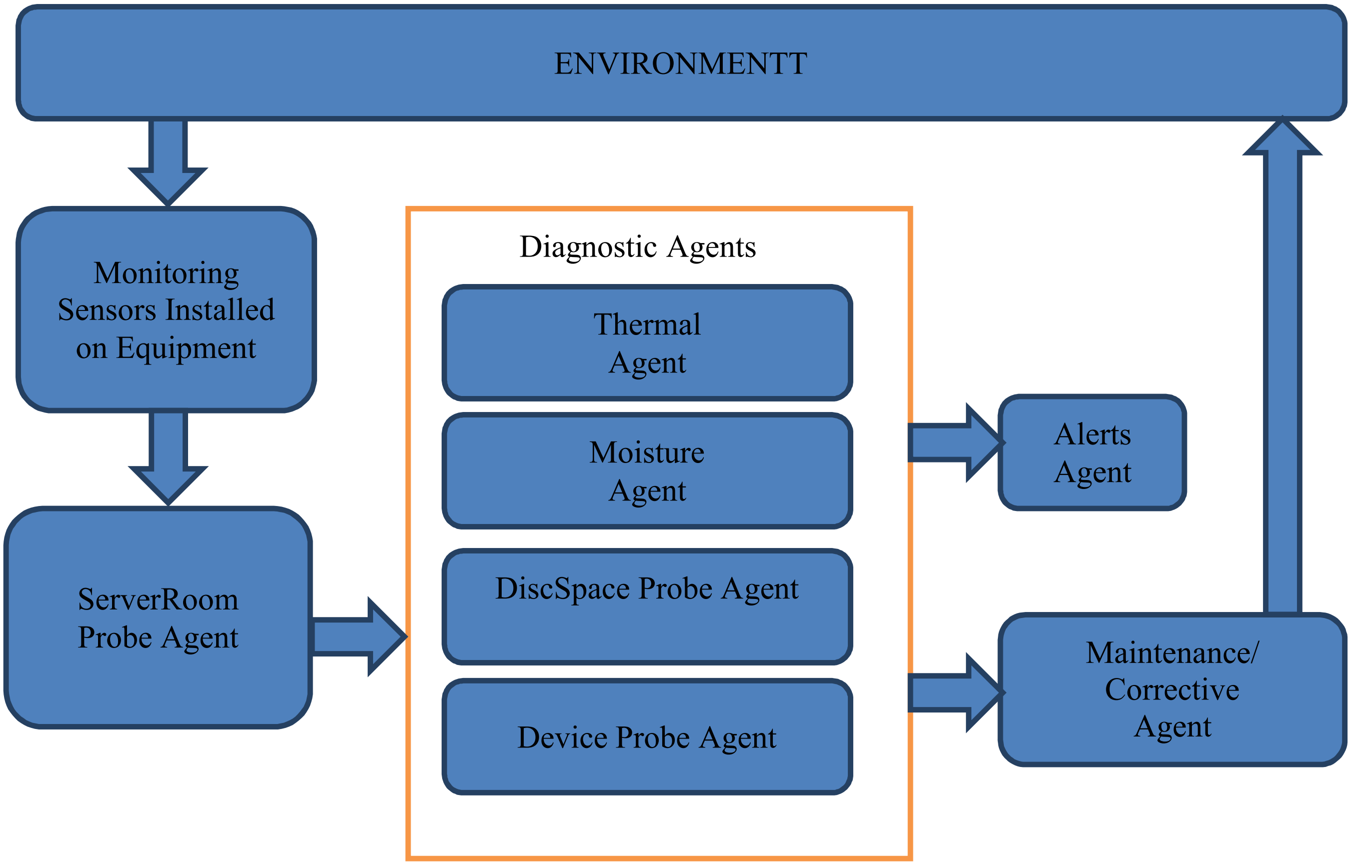


Figure 10: Agents' interactions model.

- **Device-Probe-Agent**

Device probe agent uses the set probing mechanism/protocol which at the moment is limited to Ping/echo protocol at intervals of 2 seconds, when a device is found not to respond and a timeout occurs then the device being monitored is flagged to be unreachable and thus down.

- **Server Room-Probe-Agent**

Server room probing agent connects to the Ethernet thermometer and gets the current values that are being reported, the read values are in XML format. The reading is fragged as the current values and limits are compared to identify if they are above set limits.

- **DiscSpace-Probe-Agent**

DiscSpace probe agent connects using two set protocols i.e. Telnet and SSH but we have limited the robots to SSH due to its secure nature, the agents are also programmed to connect to Unix based servers where there are preprogrammed commands on how to check disc space.

- **Alerts-Send-Agent**

Alerts send agent is responsible for generation of email and sms alerts for devices that have exceeded monitoring limits for variables being read. The agent is also able to flag the notification whether it has been sent successfully and retry to send the notification if it had failed earlier.

- **Resolution-Agent**

Resolution agent act on disc space probe agent reading and try to resolve disc space readings, the devices has been flagged whether it can be resolved automatically once the issue has been identified. The agent performs the resolution by executing commands that are set for the device.

A good project needs a good manageable process. Project Methodology is used to define fundamental principles, rules and how you are going to complete the project. In developing process, all aspect needs to be attentive. Different methodologies can support work in different phase of the system life cycle.

For developing multi-agent based Monitoring System for Consolidated bank server devices and Server Room Waterfall model will used because it very compatible with this project. Waterfall development makes it easy to keep the project under control. It limits the amount of cross-team interaction that occurs during development, it's relatively easy to estimate, and allows for greater ease in project management since plans aren't constantly being revised.

3.5 Research Methodology for the System Evaluation

We will evaluate the system using the System Usability Scale (SUS) this evaluation will help us determine if the system has met its objective as intended.

The System Usability Scale (SUS) was released by John Brooke in 1986. It was originally created as a "quick and dirty" scale for administering after usability tests on systems like VT100 Terminal ("Green-Screen") applications. SUS is technology independent and has since been tested on hardware, consumer software, websites, cell-phones, IVRs and even the yellow-pages

The SUS is a 10 item questionnaire with 5 response options.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The SUS uses the following response format:

Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
□	□	□	□	□

Table 1: SUS respondent table

Scoring SUS

- For odd items: subtract one from the user response.
- For even-numbered items: subtract the user responses from 5
- This scales all values from 0 to 4 (with four being the most positive response).
- Add up the converted responses for each user and multiply that total by 2.5. This converts the range of possible values from 0 to 100 instead of from 0 to 40.

3.6 Data collection Methodology

Data collection is done during the system requirement analysis where we needed to find out how currently monitoring is being carried out by several institutions, this would help determine how important the system is needed and on what areas we should put more focus on.

Later we would conduct another questionnaire to find out how practice is our system and will the potential users accept the system in their day to day monitoring. Also help evaluate how the system has met its requirements.

After that we conducted our research we found that the manual way of monitoring system was being practiced by many companies and very few had automated monitoring system see table 2 below

Monitoring Type	Device Uptime	Server Room	Disc Space
Manual	3	4	4
System	1	0	0

Table 2: Current monitoring methods survey findings.

We sent the questionnaire to a total of four organizations and response was as per table 2 above, by this finding it was clear that a system was required to automate monitoring.

The results were also put in a graph representation see figure 11 below

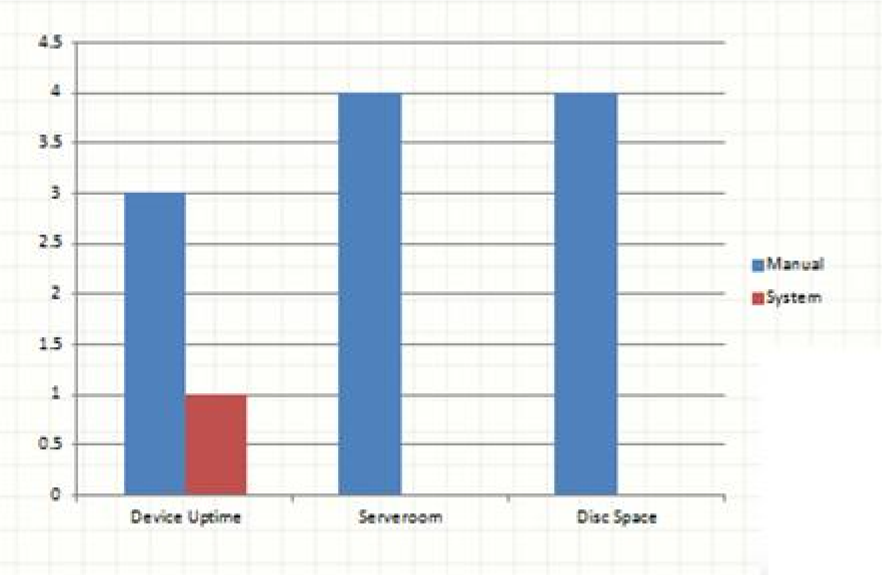


Figure 11: Current monitoring methods survey.

We also went ahead and asked about what issues or incidents result to system down time in the year 2013 for the four institutions and consolidated response is as per the table 3 below.

Cause Of Downtime	Number Of incidents
Server room Temperature too high	16
Server room Humidity too high	0
Disc space full	15
Network Device downtime	20

Table 3: Causes of downtime in the year 2013.

From this finding we find that server room temperature, disc space and monitoring of devices is critical because the major cause of downtime.

Research also showed that high number of downtime was being caused by

- High temperatures in the server room
- Lack of disc space in the servers
- And device uptime

This is evident as shown in the figure 12 below

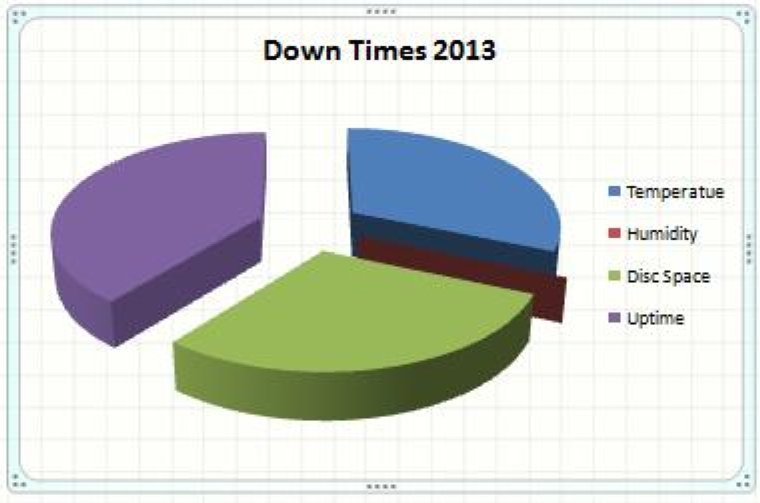


Figure 12: Causes of downtime in the year 2013.

3.7 System Research Methodology

Research is important to decide what source to use.

The focus should be on:-

- Research and decide which parts are easy and fast to development example prototype or system.
- Research the whole project concerning HW, SW, and OS.
- Research which component suitable to development the prototype, example component temperature sensor.
- Research which software suitable to support the prototype and easy create the functional.
- Prepare a questionnaire on key related questions.
- Research what tools need to use during implement the project

3.8 System Design Methodology

This focuses on the design process. In this project need three parts to design.

The designs focus on:

Categories	Description
Hardware	-Select the best HW sensor for monitor temperature -Select the best HW sensor for monitor humidity -Select the best sending sms modem -Select the best HW for hosting the system
Software	- Design agent framework - Design interface system (Visual Studio) to communicate with HW and receive the input from probes. - Design Software to monitor Servers. - Design functional agents to send alerts. - Design reactive agents to react on rules set
Device Communication	- Decide and design what device suitable to use for communication channels on alerts.

Table 4: MBIMSC Design Process

3.8.1 System Database Design

The database is designed in MySQL workbench database design, the have several relationship within the database that is **one to one** and **one to many**. This will ensure data integrity see figure 13 which shows table relationships.

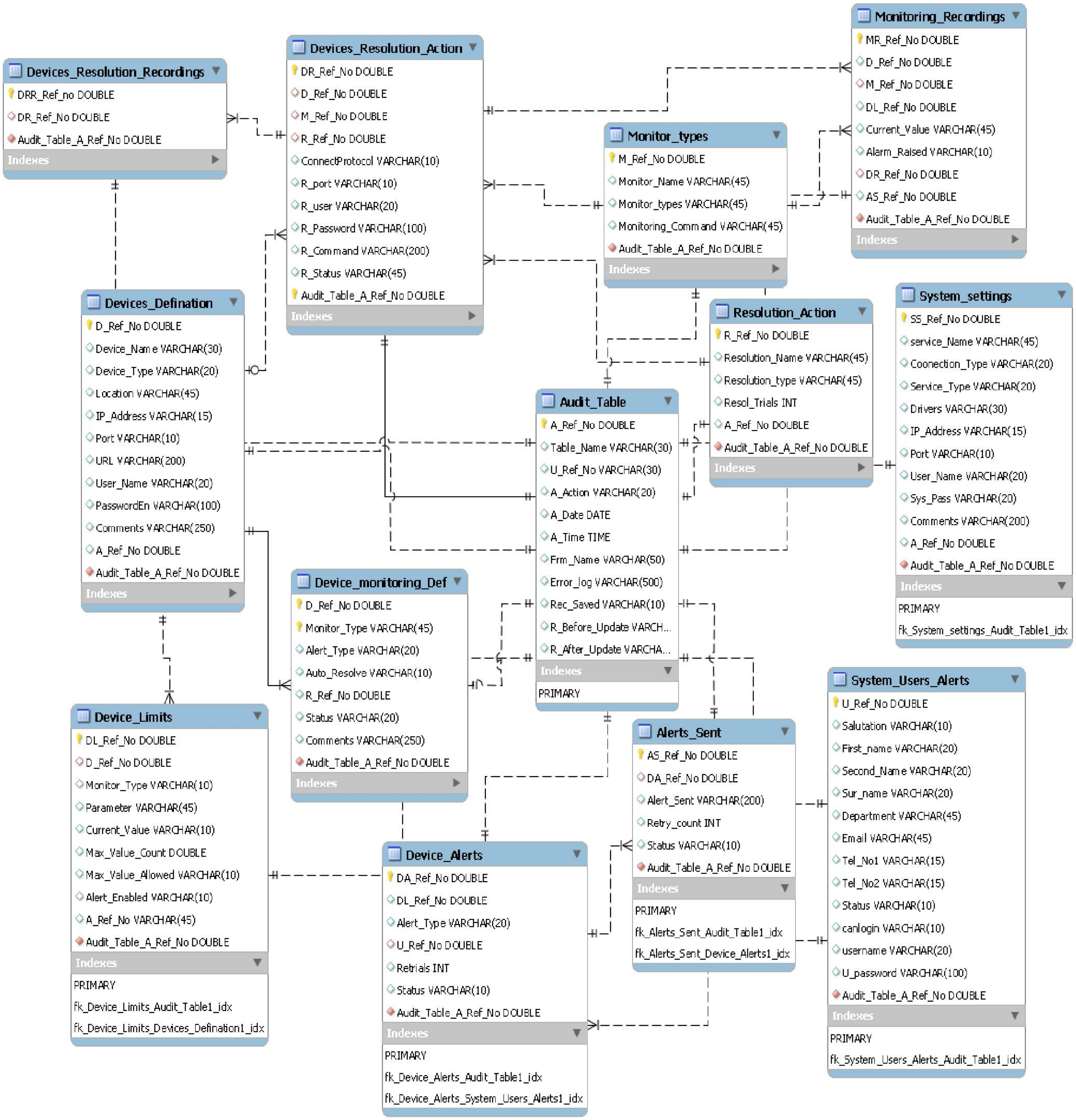


Figure 13: Entity Relationship Diagram

3.9 System Develop Methodology

The process develop and integrate monitor temperature/humidity sensor can separate into few part. The process to develop focus on:-

- Process implement framework

- Develop one agent system to communicate with HW sensors
- Develop one agent for sending alerts
- Have an agent that will be reactive to faults or issues identified
- Develop one system using Visual studio 2008 .net to provide interface for initial setup configuration. This will also provide interface for adding new devices to be monitored.

3.10 System implementation

How Agents work

All agents have been configured to use parallelism and use of background workers this is to avoid from hanging or window freezing and also the agent is able to execute task in parallel.

3.10.1 Device-Probe-Agent

Device probe agent uses the set probing mechanism/protocol which at the moment is limited to Ping/echo protocol at intervals of 2 seconds, when a device is found not to respond and a timeout occurs then the device being monitored is flagged to be unreachable and thus down.

3.10.2 Server Room-Probe-Agent

Server room probing agent connects to the Ethernet thermometer and gets the current values that are being reported, the read values are in XML format. The reading is fragged as the current values and limits are compared to identify if they are above set limits.

3.10.3 DiscSpace-Probe-Agent

DiscSpace probe agent connects using two set protocols i.e. Telnet and SSH but we have limited the robots to SSH due to its secure nature, the agents are also programmed to connect to Unix based servers where there are preprogrammed commands on how to check disc space.

3.10.4 Alerts-Send-Agent

Alerts send agent is responsible for generation of email and sms alerts for devices that have exceeded monitoring limits for variables being read. The agent is also able to flag the notification whether it has been sent successfully and retry to send the notification if it had failed earlier.

3.10.5 Resolution-Agent

Resolution agent act on disc space probe agent reading and try to resolve disc space readings, the devices has been flagged whether it can be resolved automatically once the issue has been identified. The agent performs the resolution by executing commands that are set for the device.

3.11 System Testing

The purpose of testing is to make sure the project is successful and that it's functioning as intended to meet the requirements

The testing focus on:-

- The monitoring temperature/humidity sensor can detect changes in the server room.
- The system can probe for devices.
- Check if the system is configurable, and can accept different limits
- The system is able to identify faults.
- Required corrective actions are taken to resolve faults.
- Alerts are generated to notify admins.

3.11.1 How to Test Agents

We tested the agents by using test servers for Consolidated bank and the solution is still in pilot phase, for MSC presentation we have decided to have :-

- ✓ Virtual server running in hosted Virtual machines
- ✓ GPRS Safaricom modem for sending sms
- ✓ Mobile connected via Wi-Fi to act as the network device
- ✓ Ethernet thermometer to read temperature
- ✓ Dlink router that will act as LAN
- ✓ Hair dryer that will trigger increase in temperature
- ✓ Server room Model that will represent a real server room.

3.11.2 How to trigger disc space issue

Have the server running and the DiscSpace-Probe-Agent running, On the partition being monitored introduce temporally files that will result is space depletion. This should trigger an over limit reading for that particular partition.

The reading will be populated into the database and an alert triggered to send for the device designated to assigned users. The Alerts-Send-Agent should generate an alert based on the alert type set either email or sms.

If the device has been set to auto resolve then Resolution-Agent should raise a resolution action.

3.11.3 How to test server room environment

To test temperature increate in the server room follow the below key points:-

- Have the Ethernet thermometer powered and the ServerRoom-Probe-Agent running
- Check that the current reading does not exceed the limit set for temperature
- Have the thermometer sensor set on the server room model
- Introduce heat by heating the server room model using the hair dryer
- This should trigger an over limit reading

The Alerts-Send-Agent should generate an alert based on the alert type set

3.11.4 How to trigger device downtime

This is to test the device monitoring agent, follow the key steps below to test device monitoring capability.

- Connect the computer to the DLINK Wi-Fi and the Device-Probe-Agent running
- Ensure that the set is for DLINK router and it's on, the set IP should be the one for the router.
- Make sure that the current reading is not timeout
- Allow the agent to run for a minute, check that the agent can detect the DLINK to up.
- Disconnect the DLINK router
- This should trigger a timeout and over limit reading recorded.

The Alerts-Send-Agent should generate an alert based on the alert type set

3.12 Project Deliverables

- Tested working program that demonstrating the agent capability to monitor servers and devices uptime, identify faults, rectify faults and monitor the server room environment.
- The program is able to generation alerts like sounds, emails and sm.
- Be able to deliver the generated alerts using different channels.

3.13 Tools and Resources

The resource for this project will be research and materials already published in the field of servers monitoring and multi-agent distributed systems. The development tools that will use and resources are:

- Microsoft Visual studio .net 2008 enterprise edition.
- Stationary and printing, to cost Kshs 4,000.00
- Ethernet SNMP thermometer Kshs 20,000/=
- Huawei Modem E173 USB modem Kshs 4,000/=
- Microsoft Visio - To draw layout of the SW and HW architecture

4.0 Results

After the system was compete and working we needed to know how suitable and practical it is to be used for monitoring in a middle sized institution. The evaluation was done using a survey questionnaire based on the System Usability.

4.1 Survey Results

After the questionnaire we filled and collected we analyzed the data using SUS and also used the Microsoft excel to analyze the results and also represent them statistically.

The questionnaire was sent to 12 respondents who I demonstrated the system to and 10 users were able to return their filled questionnaire.

Results findings were as below see Table 6

Question	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
I think that I would like to use this system frequently.	0	0	3	3	4
I found the system unnecessarily complex.	4	2	2	1	0

I thought the system was easy to use.	0	0	1	2	7
I think that I would need the support of a technical person to be able to use this system.	2	2	4	2	0
I found the various functions in this system were well integrated.	0	0	2	3	5
I thought there was too much inconsistency in this system.	8	1	1	0	0
I would imagine that most people would learn to use this system very quickly.	0	0	1	1	8
I found the system very cumbersome to use.	8	1	1	0	0
I felt very confident using the system.	0	0	1	2	7
I needed to learn a lot of things before I could get going with this system.	6	2	1	1	0

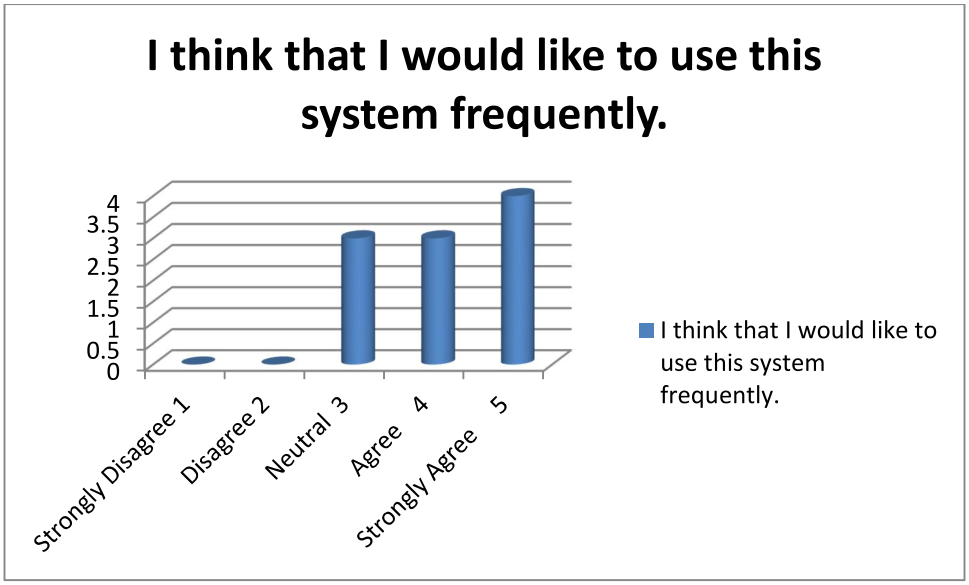
Table 6: Results on SUS findings

4.1.1 Results on Usability of the System

Ease of Use

Majority of the respondents agreed that the system was easy to used and practical, and once one learned how to go about the system it was very easy to even guides others on how to use.

Question 1: I think that I would like to use this system frequently.

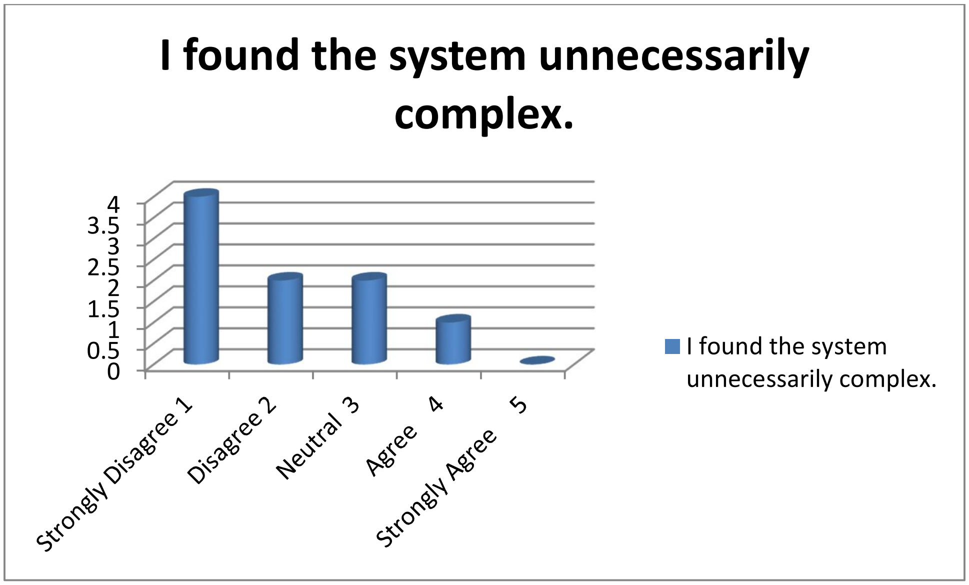


Findings

Many users agreed to that the system would of help to them for they want to use the system frequently.

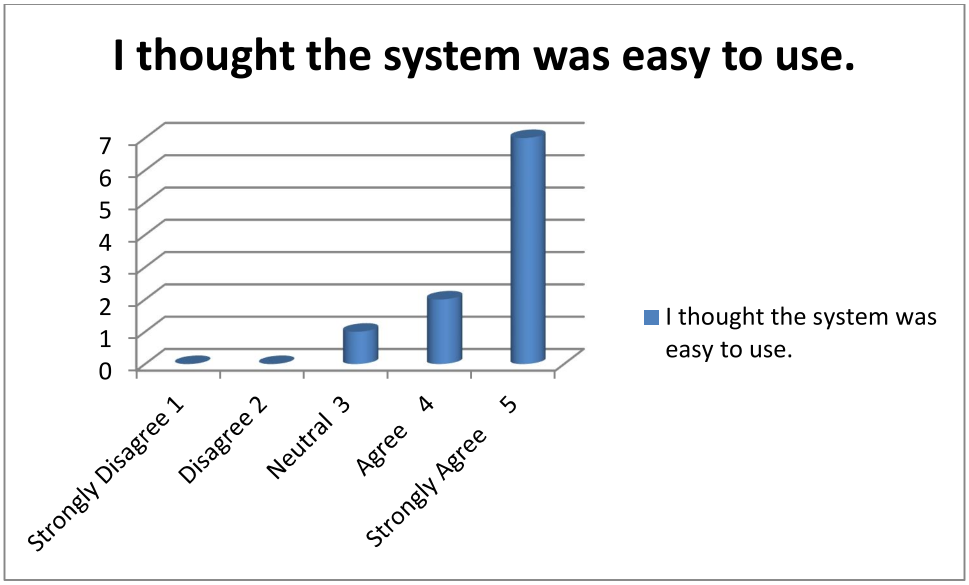
Question 2: I found the system unnecessarily complex.

From the findings below many users found the system to be necessary complex, this would interpret to the possibility of ease to understand the system.



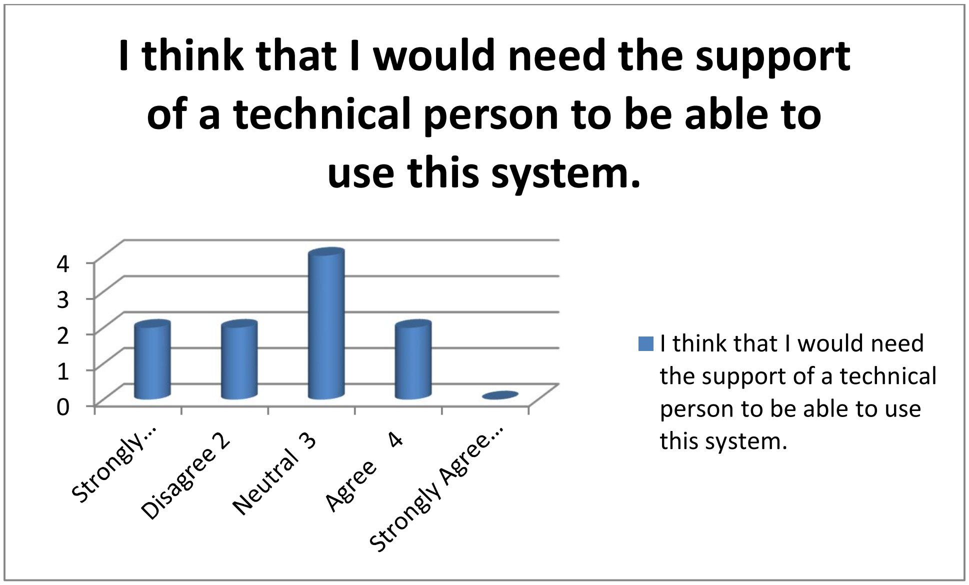
Question 3: I thought the system was easy to use.

Many users found the system easy to use, even by the aspect that there is a control panel to control use of agents which simplifies the agent's administration function, also the system does not need "baby sitting" in that one you setup the devices then that is it.



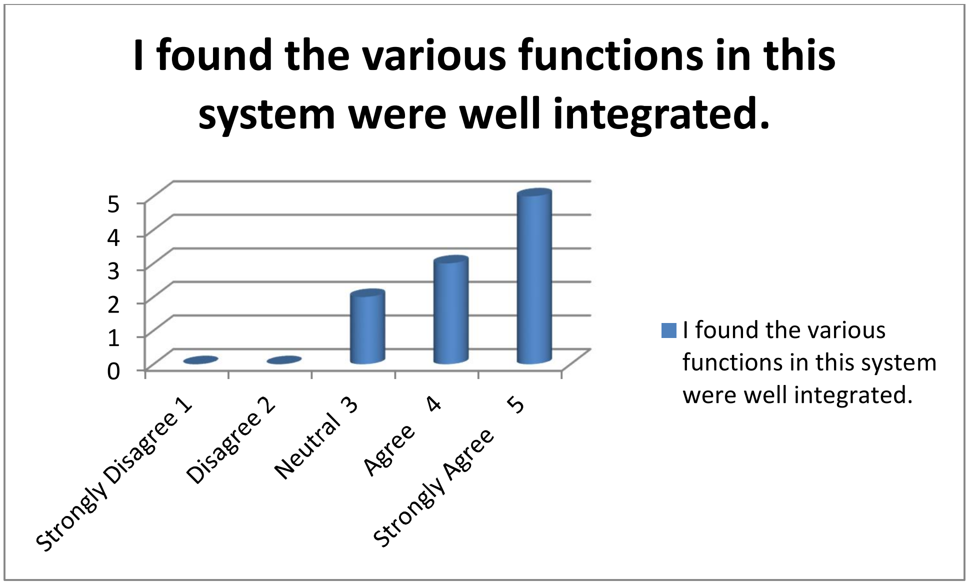
Question 4: I think that I would need the support of a technical person to be able to use this system.

The respondents did not agree to this for the interface has been simplified to a novice user level. By having a graphical user interface which users are familiar with.



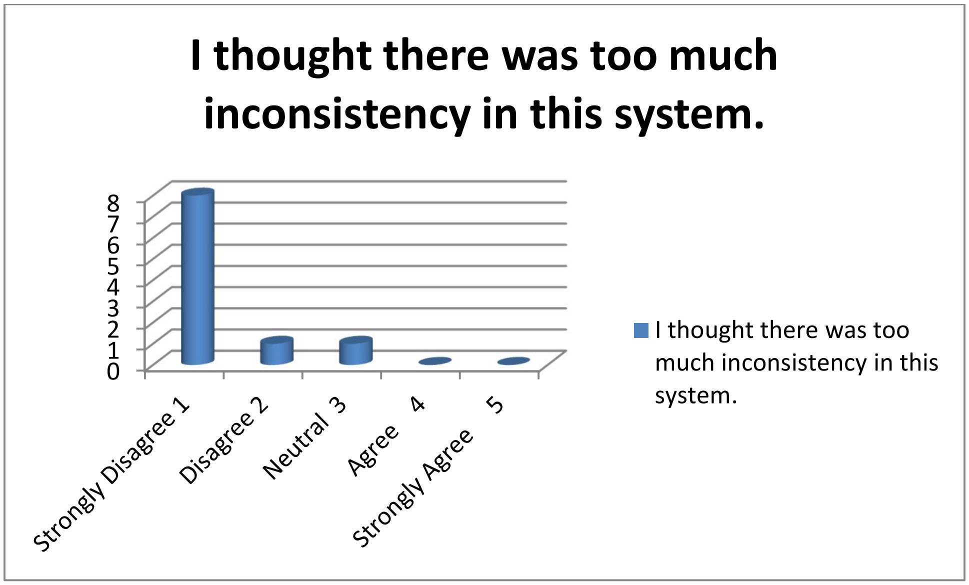
Question 5: I found the various functions in this system were well integrated.

The user configuration module is modular but well integrated that's what most users found as per the response on SUS.



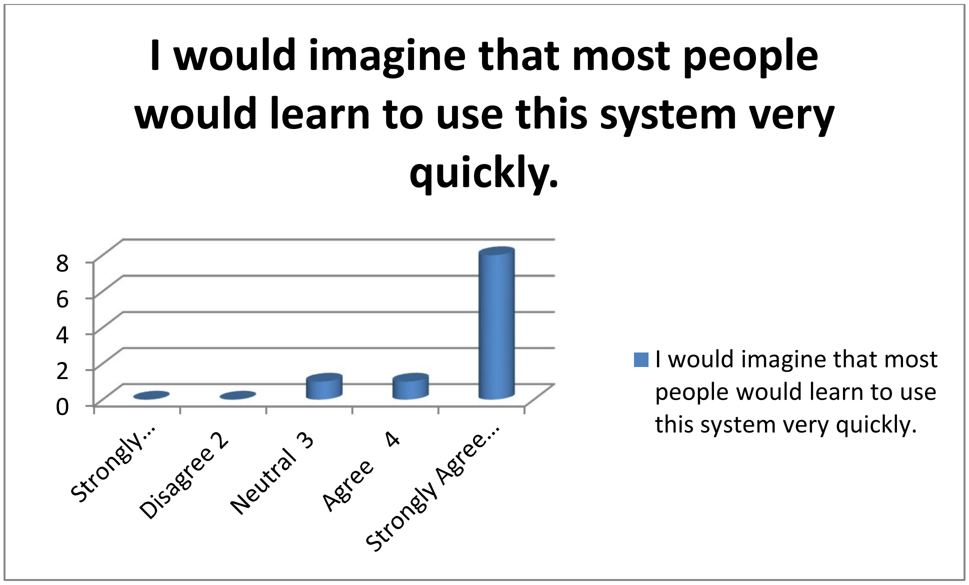
Question 6: I thought there was too much inconsistency in this system.

Majority of users disagreed with this question to mean that they found the system to have consistency which is a good attribute.



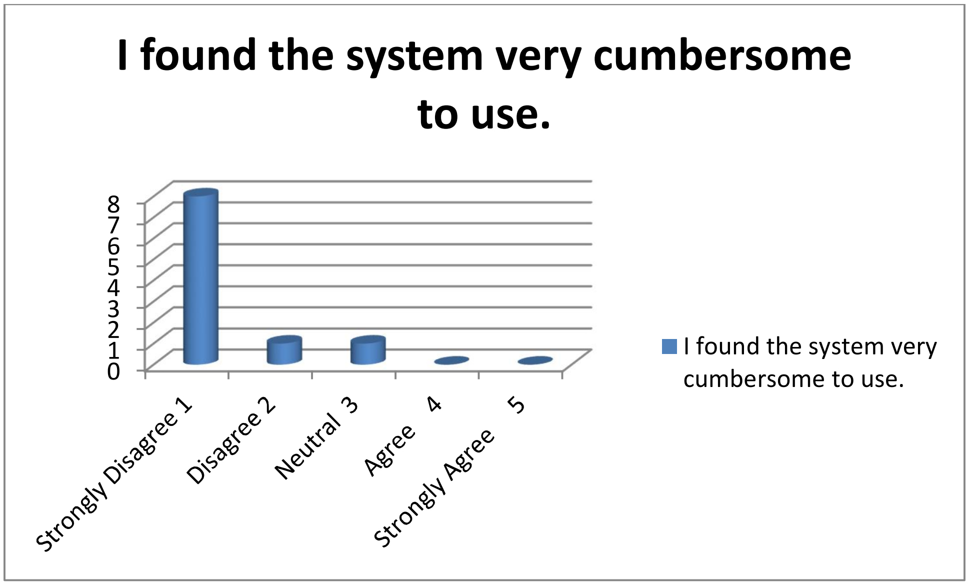
Question 7: I would imagine that most people would learn to use this system very quickly.

The system was found to be easy to learn its ways of operation, and usability which would greatly be accepted by many due to this factor.



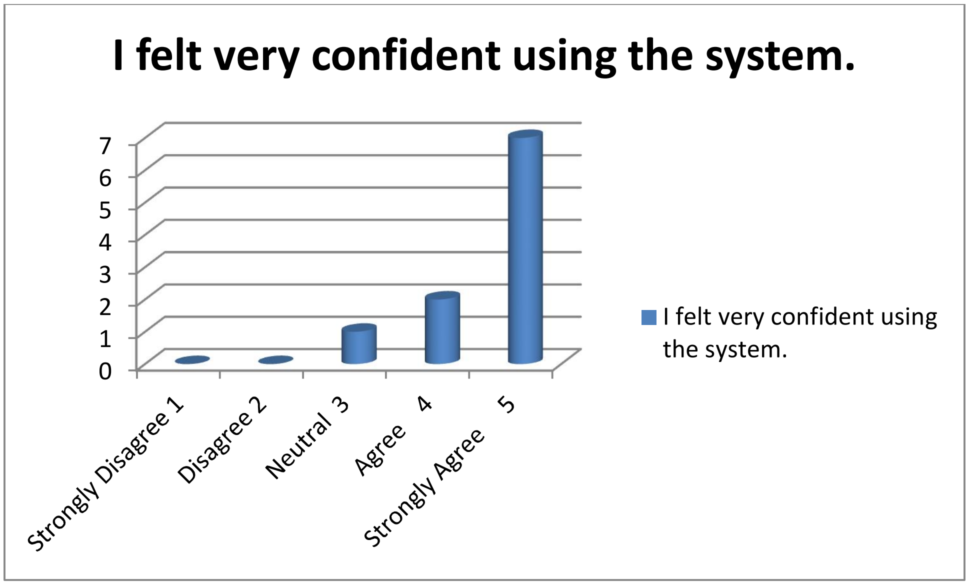
Question 8: I found the system very cumbersome to use.

The system was found to be easy to use for many users disagreed that the system was cumbersome to use, and the more any application is less cumbersome to use the more it's accepted.



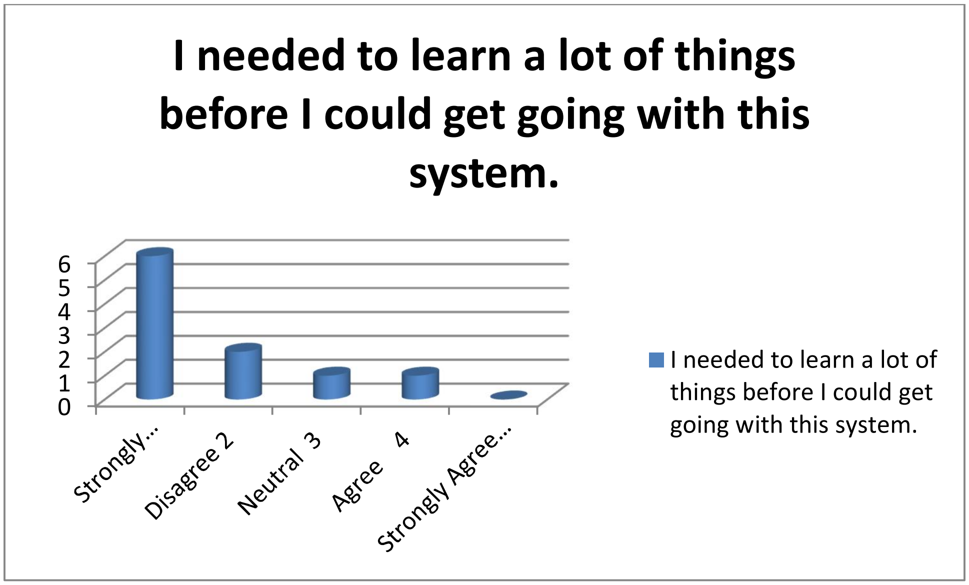
Question 9: I felt very confident using the system.

Despite the application being developed in rather new technology many users strongly agreed that they felt confident to use it and this means that they are drawn toward using the system.



Question 10: I needed to learn a lot of things before I could get going with this system.

The system is scaled down to a point the technical jargons and implementation is not known to the user and users appreciated that they did not need to learn a lot of thing for them to operate the application.



5.0 Conclusion and Recommendation

5.1 Achievements

The following were achieved as per the objectives earlier mentioned objective we were able to develop a tested working program that demonstrating the agent capability to:-

- Monitor servers
- Monitor the server room environment i.e. Temperature, humidity and Dew point
- Monitor devices uptime and identify faults
- Monitor Linux based disc space
- Generate sms and email alerts
- Rectify/Resolve disc space faults

The system is light and able to run on many windows environment and this means it doesn't require many resources to run. This will translate to less financial constrains when one want to adopt such a monitoring solution.

5.2 Challenges

5.2.1 Development related

The were no major challenges that we encountered while developing this system but the, we had a few challenges like the main development environment was on window XP operating system which had faults and had to upgrade to window 7 profession which proved to be very stable.

5.2.2 Hardware procurement

We did also encounter delay when shipping the H2E Ethernet thermometer which we could not locally and it took time to get it here, pay duty and have it cleared.

During the development the laptop crashed and had to rebuild most of the application components like database and lost code.

5.2.3 Data Collection Process

We did also encounter challenged when we sent out the questionnaire, we had to constantly do follow up in order to have the questionnaire filled. There were also some respondents that did not return the questionnaires despite the follow up.

5.3 Recommendations and Future Work

In order to have auto resolve action on server room environment monitoring. I would recommend that the system be enhanced with external relay mechanism that can be triggered to switch on a backup AC installed in the server room.

practical recommendation though we are developing for our case, is inclusion of monitoring services and transaction; this will also be effective in monitoring ATMs. This will be crucial to reports incidents of transaction declines.

There are other possible mechanisms that can be incorporated to enhance the system that is introducing monitoring of services with capability of resolving services down times e.g. web services, server daemons etc.

i. References

[1]. HP Software Solutions url: http://www.slideshare.net/hpsoftwaresolutions/hp-software-universe-managing-virtualized-environments-agentless-and-agentbased-solutions-for-managing-vmware-environments-using-hp-sitescope-operations-manager-agents-and-virtualization-spi

[2] Russell & Norvig 2003, p. 33

[3] Salamon, Tomas (2011). Design of Agent-Based Models. Repin: Bruckner Publishing. pp. 42–59. ISBN 978-80-904661-1-1.

[4] Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Novig

[5] Amdahl, Gene M. (1967). "Validity of the single processor approach to achieving large scale computing capabilities". Proceeding AFIPS '67 (Spring) Proceedings of the April 18–20, 1967, spring joint computer conference: 483–485.

[6]. What is a server? URL: http://www.webopedia.com/DidYouKnow/Hardware_Software/2005/servers.asp

[7]. Server room URL: http://en.wikipedia.org/wiki/Server_room

[8]. Weiss. Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999

[9]. Gsm usb modem. URL: http://mobielinternet.nl/what-is-a-usb-modem-en/

[10]. Ramanath and Gilbert's methodology http://archimede.bibl.ulaval.ca/archimede/fichiers/23343/ch02.html

[11]. Danny Weyns. Architecture-Based Design of Multi-Agent Systems. Verlag Berlin Heidelberg 2010

[12]. J. Kramer, J. Magee, in Self-Managed Systems: An Architectural Challenge. FOSE'07: 2007 Future of Software Engineering, Minneapolis, MN (IEEE Computer Society, Washington, DC, 2007), pp. 259–268

[13]. L. Bass, P. Clements, R. Kazman, Software Architecture in Practice (Addison-Wesley,Boston, MA, 2003)

[14] Durfee, E. H. & Montogomery, T. A., "MICE: A Flexible Testbed for Intelligent Coordination Experiments," Proceedings of the 1989 Distributed Artificial Intelligence Workshop, pp. 25-40, 1989.

[15] T. Drashansky, E. N. Houstis, N. Ramakrishnan, J. R. Rice, “Networked agents for scientific computing,” Communications of the ACM, vol. 42, NO. 3, March 1999, pp. 48-53.

[16] R. Khosla, T. Dillon, “Intelligent hybrid multi-agent architecture for engineering complex systems,” Proceedings of the 1997 IEEE international Conference on Neural Networks, vol. 4, pp. 2449-2454.

[17] Honavar, V., Miller, L., and Wong, J. “Distributed Knowledge Networks” In: Proceedings of the IEEE Information Technology Conference. 1998.

Appendix B

Questionnaire

(1) How are the servers and devices monitored on uptime?

(2) What is the recommended serve room temperature?

(3) What is the recommended serve room humidity?

(4) How many incidences have been recorded this year (2013) on system downtime?

(5) How many incidences have the AC failed and led servers shutting down because of temperatures?

(6) How many incidences have been recorded where humidity was an issue?

(7) What are the associated risks when the server goes down?

(8) What financial loss can are estimated in case of prolonged service downtime?

(9) Have you ever expected downtime due to high temperatures elsewhere apart from the current organization?

(10) Have you ever expected downtime due to high humidity elsewhere apart from the current organization?

(11) What solutions have you implemented to notify against high temperatures in the server room?

(12) What solutions have you implemented to notify against high humidity in the server room?

(13) Do you have notification mechanism in case of severs and devices downtime?

(14) Do you have any automation to resolve system issues?

(15) Would you like to implement a solution that will aid you in monitoring servers, devices and serve room environment?

System Usability Scale Questionnaire

	Question	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
1	I think that I would like to use this system frequently.					

2	I found the system unnecessarily complex.					
3	I thought the system was easy to use.					
4	I think that I would need the support of a technical person to be able to use this system.					
5	I found the various functions in this system were well integrated.					
6	I thought there was too much inconsistency in this system.					
7	I would imagine that most people would learn to use this system very quickly.					
8	I found the system very cumbersome to use.					
9	I felt very confident using the system.					
10	I needed to learn a lot of things before I could get going with this system.					

Installation Guide

1. The hardware requirements for the system before use must meet the below minimum requirement.

System Software requirements

- MySQL server 5.2
- Install MySQL workbench

Hard ware Requirements

- Window 7 professional or Windows Xp Professional
- 1 GB RAM
- GSM Modem
- GHZ processor speed
- 80 GB hard disk

2. Install the MySQL Server as per the installation guide provided by the software vendor

3. Install the application.

4. Install the GSM modem

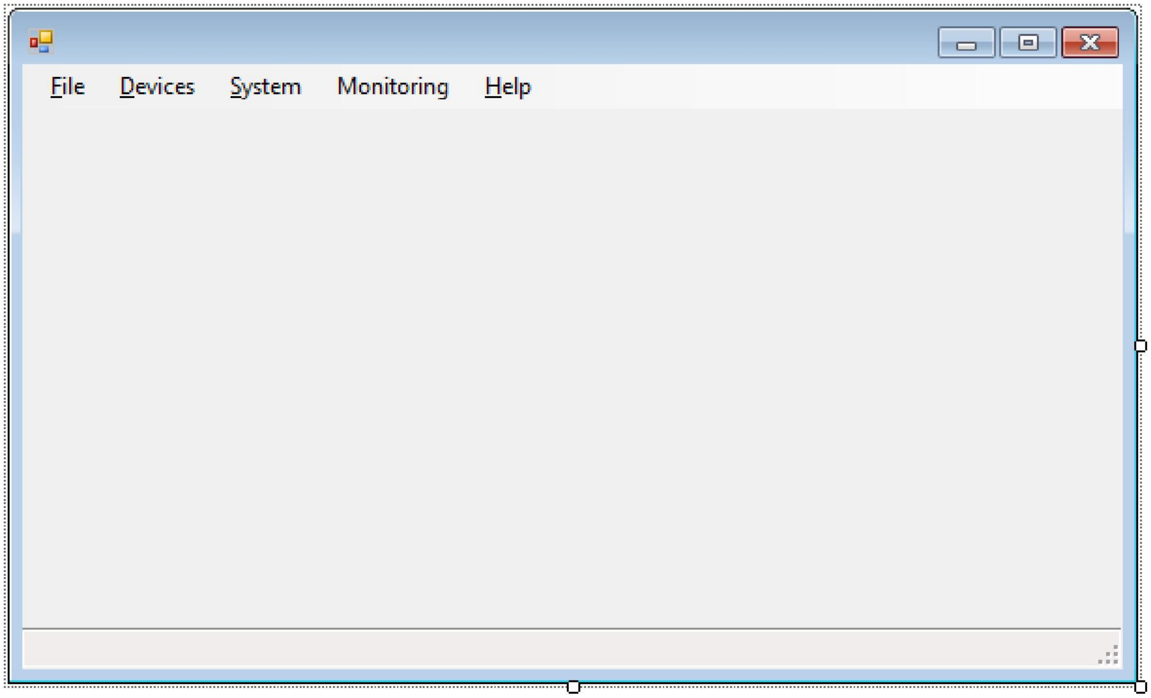
5. Set the minimum default parameters on the configuration files

- Smtp server
- Host MySQL server IP
- MySQL login credentials
- Modem Port

Screen shots of the main configuration application

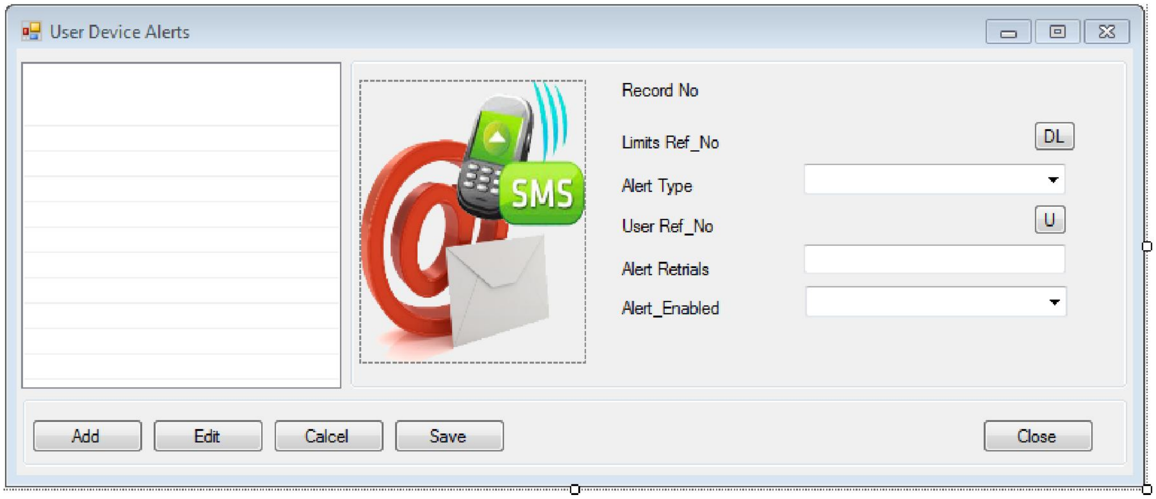
Main System Configuration window

This is the parent window where all menus are to call other forms



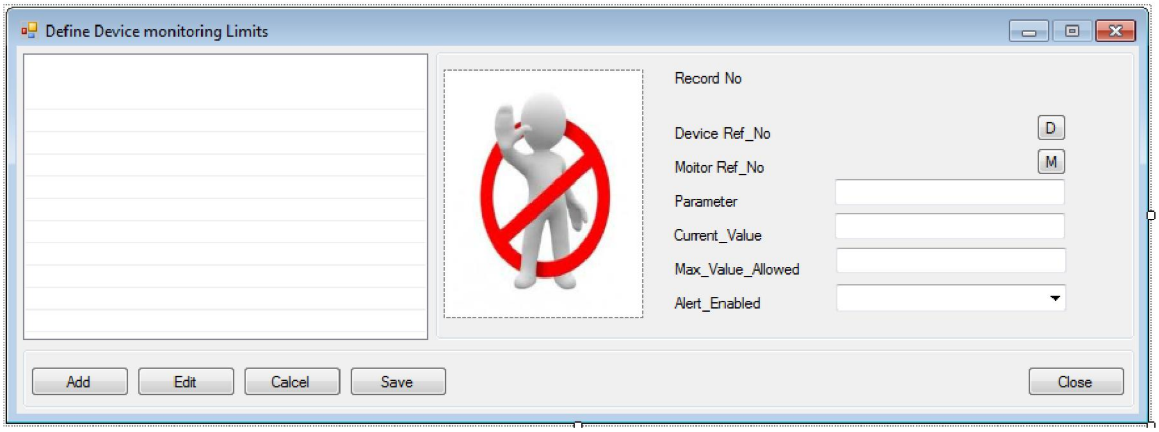
Device Alerts Definition window

This window is used to define alert that are enabled or rose relating to the device defined



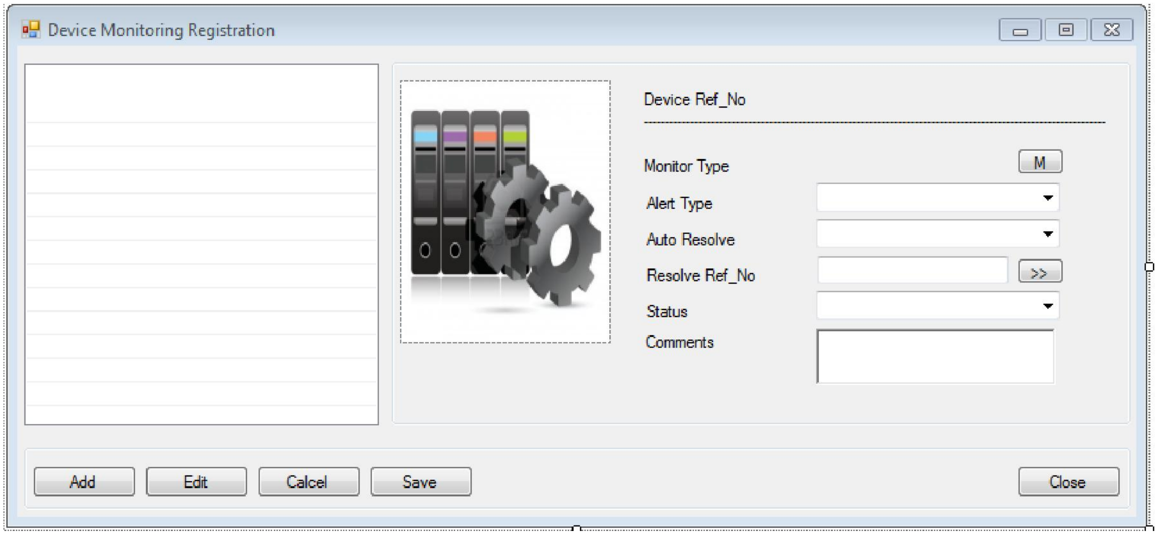
Define Device monitoring Limits Window

Limits or threshold to any variable being monitored are defined here.



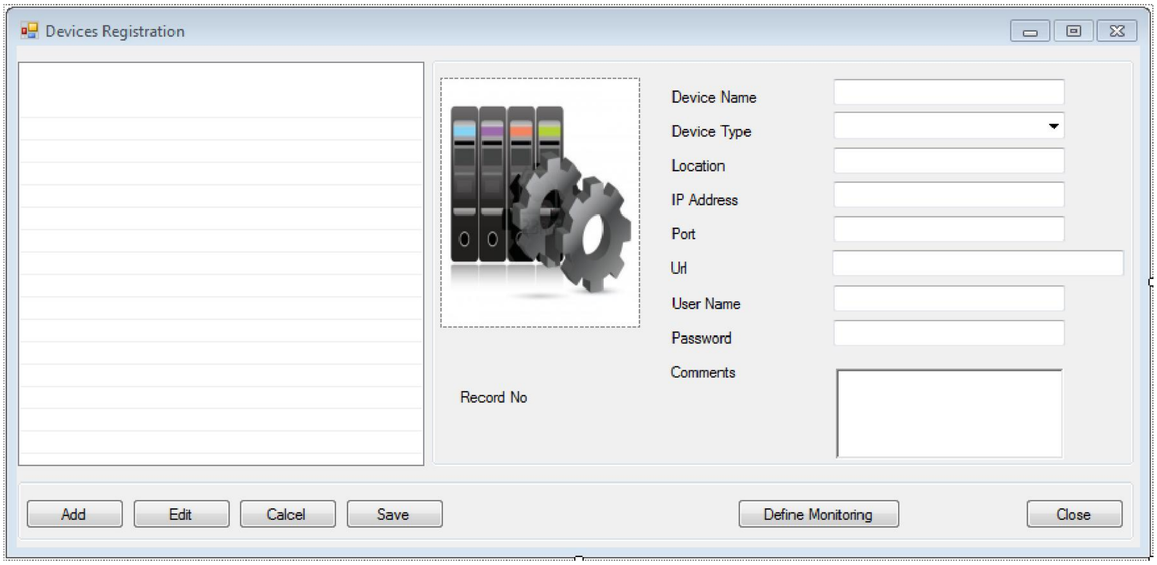
Device Monitoring Registration

The type of monitoring of any device is defined on this window where you can control if that variable will be monitored or not.



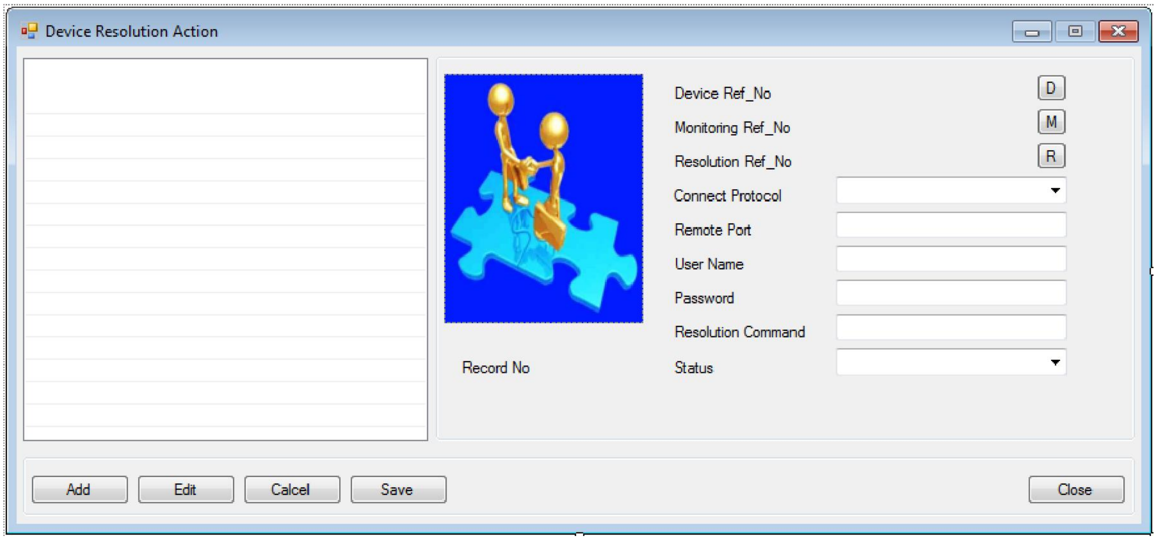
Devices Registration

This window is used to register the devices that can be monitored



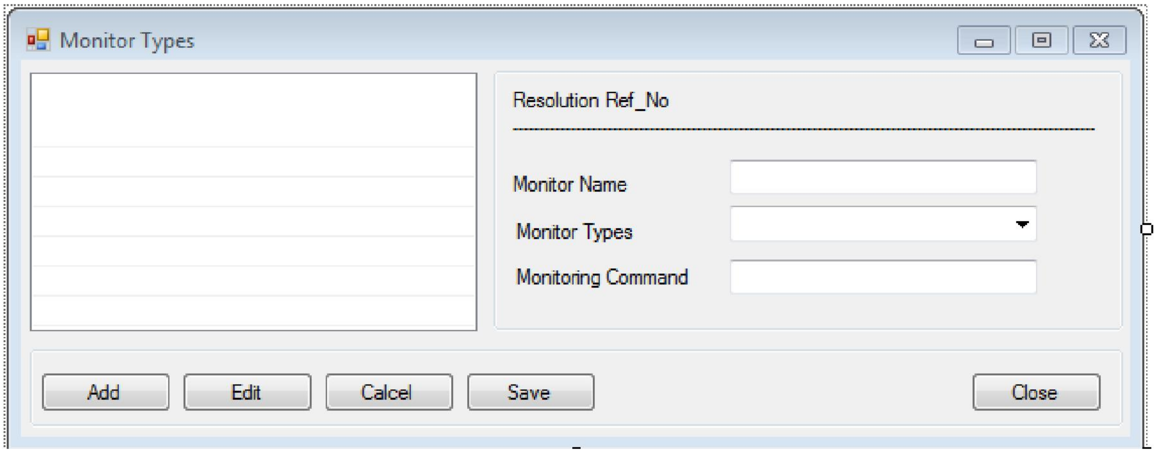
Device Resolution Action

If a fault can be auto resolved by the resolution agent this is where we define this.



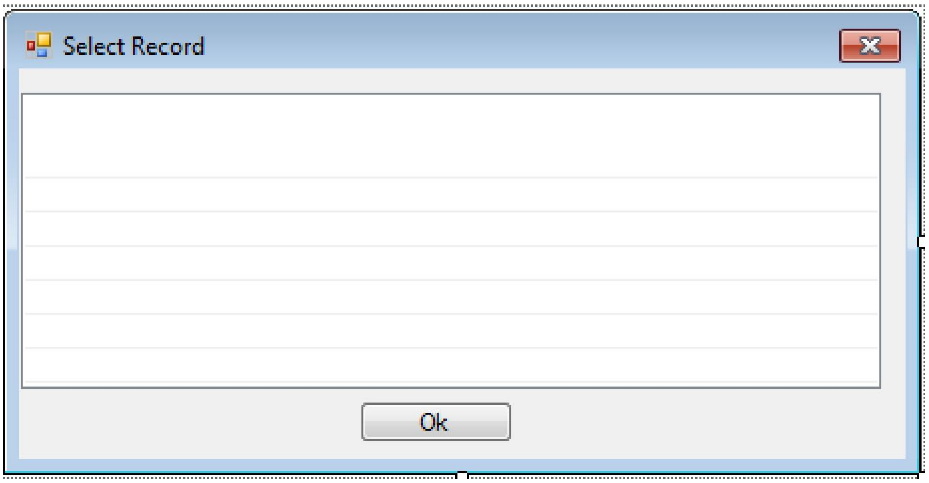
Monitor Types

The possible monitoring types by the agents are defined on this window.



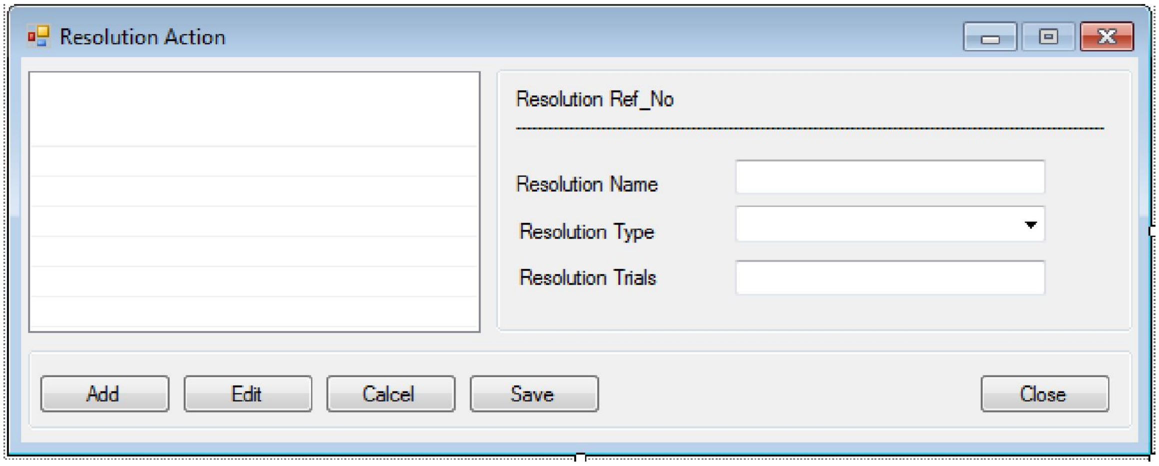
Select Record window

This window is referred to by many agents and is used to query and return data to window forms.



Resolution Action

This window is used to define resolution actions that the resolution agent is able to perform.

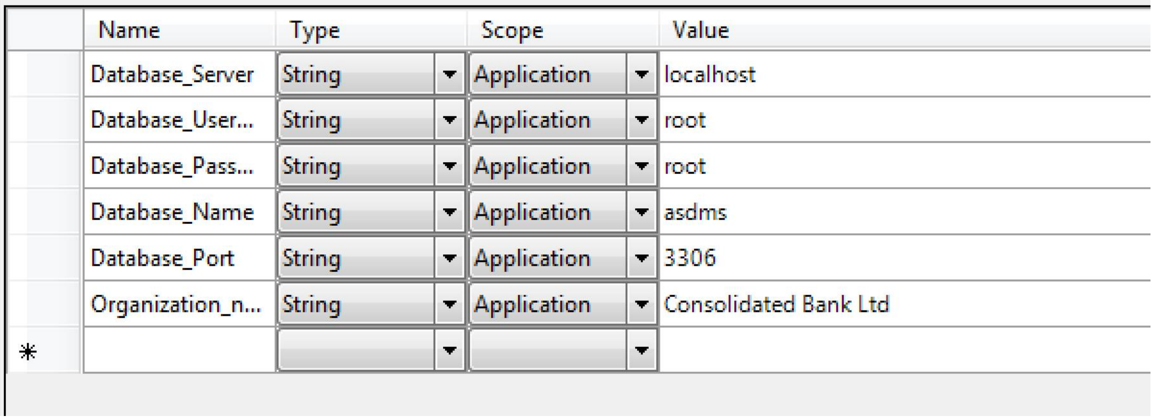


System Resources

The below resources were used on the window forms

Settings.settings

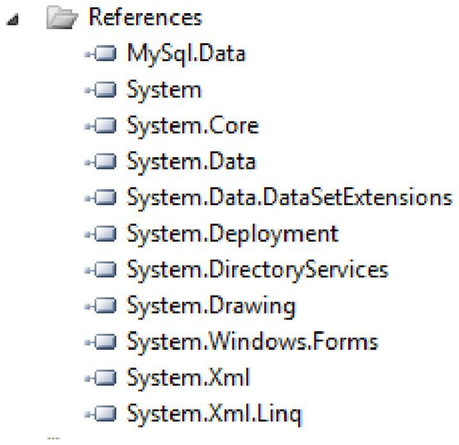
This are some of the system default setting that are required for the application to run.



	Name	Type	Scope	Value
	Database_Server	String	Application	localhost
	Database_User...	String	Application	root
	Database_Pass...	String	Application	root
	Database_Name	String	Application	asdms
	Database_Port	String	Application	3306
	Organization_n...	String	Application	Consolidated Bank Ltd
*				

System References

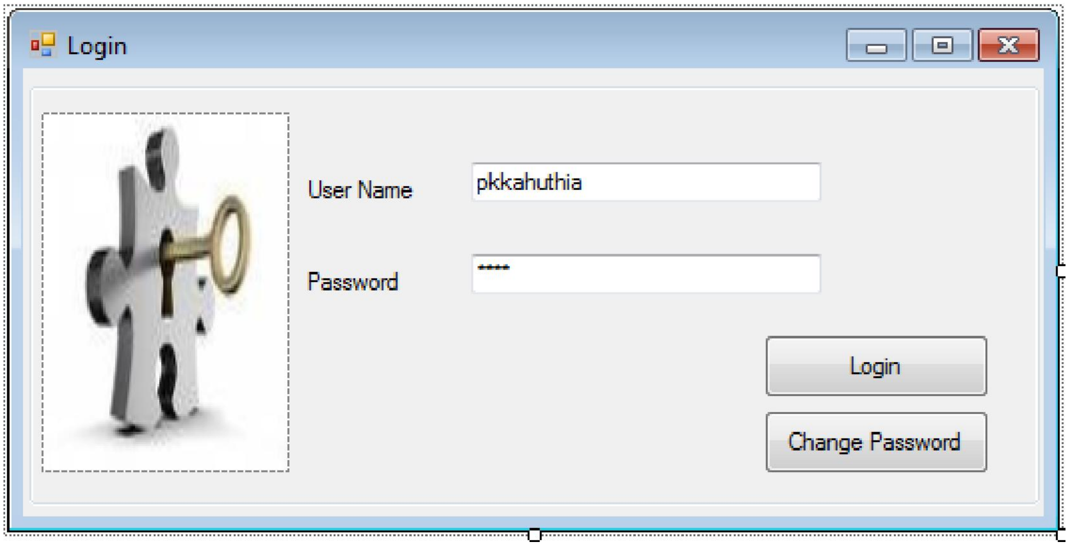
Some of the references that we used during development



```
References
    MySql.Data
    System
    System.Core
    System.Data
    System.Data.DataSetExtensions
    System.Deployment
    System.DirectoryServices
    System.Drawing
    System.Windows.Forms
    System.Xml
    System.Xml.Linq
```

Login Window

Login window is used when getting access to the system, it protect an authorized access to the system.



User Name: pkkahuthia

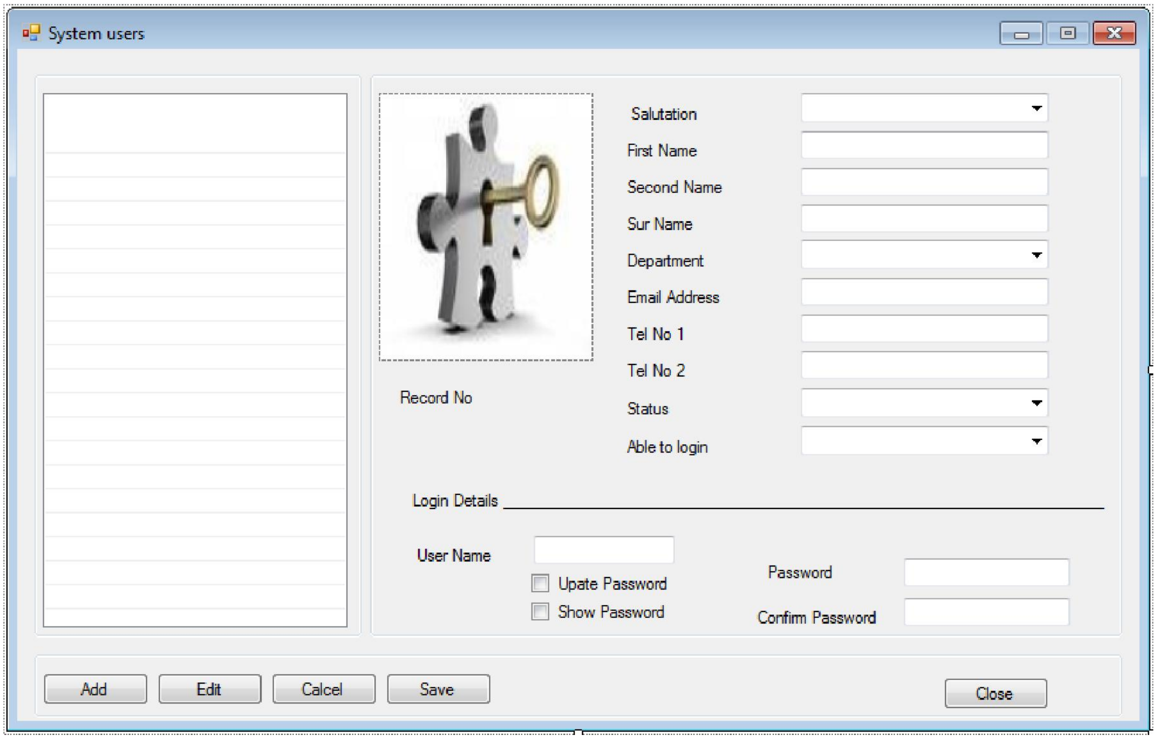
Password: ****

Splash window

This is the first form that appears when one runs the application.

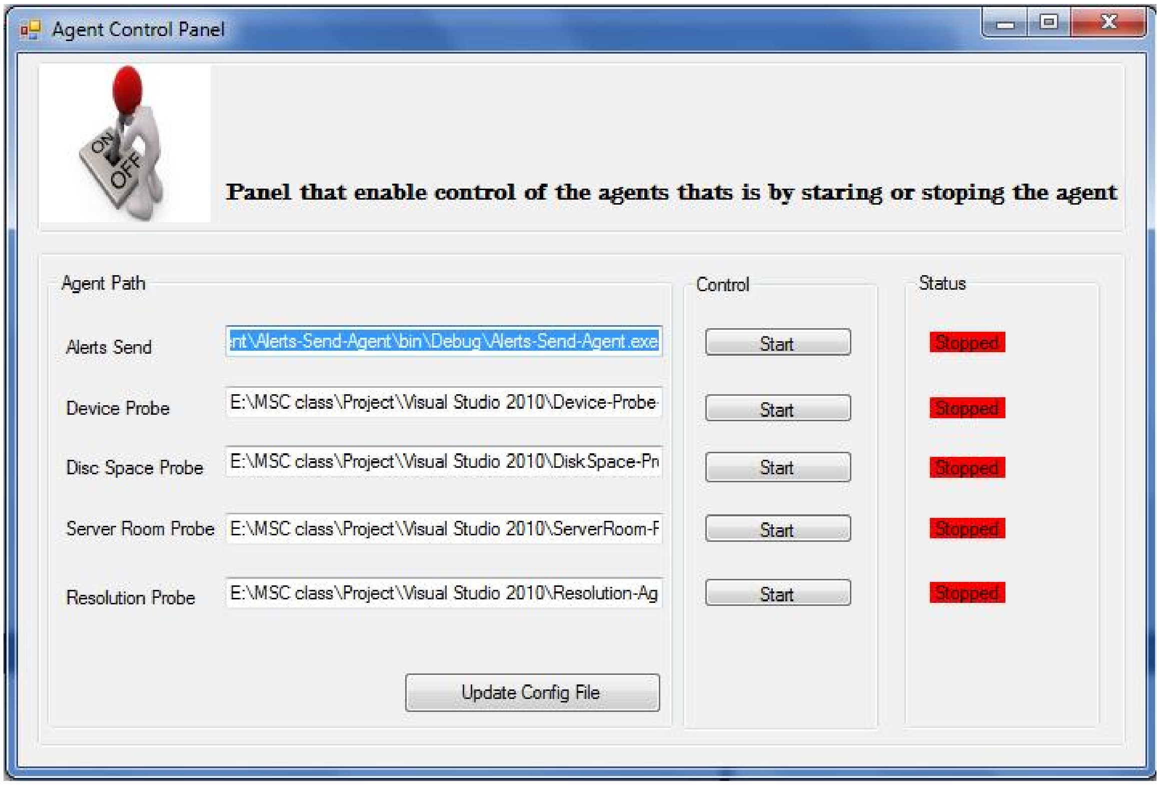
System Users

This is the window that is used to define system users and also users who are set to receive alerts for devices set.



Agent Control Panel

Agents control panel is used to manipulate agents, it can be used to start or stop the agents if need be. The same is also used to check states of each agent.



Panel that enable control of the agents thats is by staring or stoping the agent

Agent Path		Control	Status
Alerts Send	nt\Alerts-Send-Agent\bin\Debug\Alerts-Send-Agent.exe	Start	Stopped
Device Probe	E:\MSC class\Project\Visual Studio 2010\Device-Probe-	Start	Stopped
Disc Space Probe	E:\MSC class\Project\Visual Studio 2010\DiskSpace-Pr	Start	Stopped
Server Room Probe	E:\MSC class\Project\Visual Studio 2010\ServerRoom-F	Start	Stopped
Resolution Probe	E:\MSC class\Project\Visual Studio 2010\Resolution-Ag	Start	Stopped

System Sample Code

BackgroundWorker

```
Private Sub tick(ByVal sender As Object, ByVal e As System.Timers.ElapsedEventArgs)

        If (BackgroundWorker1.IsBusy = False) Then
            ' Start the asynchronous operation.
            Console.WriteLine("Agent initiating Resolution action.......")
            BackgroundWorker1.RunWorkerAsync()
        Else
            Console.WriteLine("Agent is Busy.......")
        End If

    End Sub

Private Sub BackgroundWorker1_DoWork(ByVal sender As Object, ByVal e As
DoWorkEventArgs) Handles BackgroundWorker1.DoWork
        Load_Servers_Probes()
    End Sub
```

Task Parallelizing

```
Tasks.Parallel.Invoke(
                          Sub()
                              Console.WriteLine("Agent Performing resolution action for server.."
& Convert.ToString(IP).ToString)
                              Dim ReadResult = ExecuteSshCommand(R_port,
Convert.ToString(IP).ToString, R_user, R_password, R_Command)

                              If ReadResult = "Succesful" Then
                                  Post_Resolution("" & DR_Ref_No & "")
                              End If
                          End Sub)
```

Decryption procedure

```
    Public Function AES_Decrypt(ByVal input As String) As String
        Dim pass As String
        'define the key
        pass = "ThiSkeyHere4"

        Dim AES As New System.Security.Cryptography.RijndaelManaged
        Dim Hash_AES As New
System.Security.Cryptography.MD5CryptoServiceProvider
        Dim decrypted As String = ""
        Try
            Dim hash(31) As Byte
            Dim temp As Byte() =
Hash_AES.ComputeHash(System.Text.ASCIIEncoding.ASCII.GetBytes(pass))
            Array.Copy(temp, 0, hash, 0, 16)
            Array.Copy(temp, 0, hash, 15, 16)
            AES.Key = hash
            AES.Mode = Security.Cryptography.CipherMode.ECB
            Dim DESDecrypter As System.Security.Cryptography.ICryptoTransform =
AES.CreateDecryptor
            Dim Buffer As Byte() = Convert.FromBase64String(input)
            decrypted =
System.Text.ASCIIEncoding.ASCII.GetString(DESDecrypter.TransformFinalBlock(Buf
fer, 0, Buffer.Length))
            Return decrypted
        Catch ex As Exception
            Return "o"
        End Try
    End Function
End Module
```

Database Connection Procedure

```
  Public Sub ConnectDatabase()
    Dim Database_Name, Database_Server, Database_UserName, Database_Password,
Database_Port As String

    Database_Name = ConfigurationManager.AppSettings.Item("Database_Name")
    Database_Server = ConfigurationManager.AppSettings.Item("Database_Server")
    Database_UserName =
ConfigurationManager.AppSettings.Item("Database_UserName")
    Database_Password =
ConfigurationManager.AppSettings.Item("Database_Password")
    Database_Port = ConfigurationManager.AppSettings.Item("Database_Port")

    Try
      DisconnectDatabase()
      If conn.State = ConnectionState.Closed Then
        conn.ConnectionString = "DATABASE=" & Database_Name & ";" _
         & "SERVER=" & Database_Server & ";user id=" & Database_UserName &
";" _
         & "password=" & Database_Password & ";port=" & Database_Port &
";charset=utf8"
        conn.Open()
      End If
    Catch myerror As MySqlException
      ' MessageBox.Show("Error connecting to the server: " + myerror.Message,
"MySqlConnection Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
      Write_error_log_File("System_global error  ", myerror.Message)
    End Try
  End Sub
```