LAB WATCHER

BY

Muhammad Mudassar Tufail

Department of Computer Science

Quaid-I-Azam University, Islamabad

Session: 2012-2014

LAB WATCHER

Developed By

Muhammad Mudassar Tufail

Supervised By

Mr. Umer Rashid

Department of Computer Science

Quaid-I-Azam University, Islamabad

Session: 2012-2014

In The Name of ALLAH who is the most Merciful and Compensate.

A report is submitted to the Department of
Computer Sciences, Quaid-I-Azam University Islamabad
As a partial fulfillment of the requirements for the award of the
Degree of Master in Computer Science

Dedicated to…

My Loving Parents & Teachers.

DECLARATION

I hereby declare that report is my own work and effort and that has not been submitted anywhere for any reward. Where other source of information has been used they have been acknowledged.

Muhammad Mudassar Tufail

Acknowledgement

I owe my deepest and sincere gratitude to my supervisor **Mr. Umer Rashid** for helping me in doing this project, due to his guidance and help I successfully completed my this project.
The day since I step into Computer Science Department at QAU, I have learnt so much each day and I cannot forget the constant encouragement and support of my teachers throughout my study session.

I would like to thanks **Chairman of Department Dr. Afzal Bhatti**, **Dr. Onaiza Maqbool**, **Dr. Rabeeh Ayaz Abbasi**, **Madam Memoona Afsheen**, **Mr. Syed Muhammad Naqi**, **Dr. Khalid Saleem** and **Dr. Shouaib Karim**. I feel myself lucky to be taught by teachers like them.

I would also like to thank my family and friends for always being there for me when I needed.

Muhammad Mudassar Tufail

Abstract

Computer Labs are usually equipped with interconnected Personal Computers (PCs). The heavy and improper utilization of the PCs in the Lab setting makes them more vulnerable, and prone to the hardware as well as the software based errors. The hardware as well as software based error in PCs in Lab settings will be difficult to diagnose, because they are not properly reported by users. This not only degrades the performance, but also lowers the availability of the PCs in the Labs. The technical staff usually monitors PCs, and diagnoses errors in Lab settings on the regular basis or on user complaints. The continuous monitoring of the PCs in Lab setting, the diagnosis of errors and their reporting is a frustrating and tedious task for the technical staff. It also requires technical skills and human resources. In this project we are interested to design and develop a tool that will monitor PCs in Lab settings for errors, diagnose and report hardware as well as software based errors on real-time bases.

This project includes:

Identification of set of the most common software and hardware based errors that degrades system performance or affects the availability of various types of software services available over the PCs in Lab settings.

Implementation of the software based real-time service that will execute on the PCs in Lab settings. The service will seamlessly monitors, diagnose and report errors occurred in PCs in the Lab settings.

A web based on-line application that provides the detailed view of reported errors to the technical staff on real-time bases. The on-line web application also provided the management of reported software, hardware based errors, hardware statuses and also system specifications of the Lab PCs.

Table of Contents

Chapter 1: Introduction

Chapter 2: Requirements Specification

Chapter 3: Requirements Analysis

Chapter 4: System Detail Design

Chapter 5: Implementation

Chapter 6: Testing

Chapter 7: Conclusions and future work

Chapter 1

Introduction

This chapter provides overview of the purposed system. It describes why system is needed, what problems it deals with, what are its key features, related work which have done so far and finally overview of the report.

1.1 Problem Definition

Computer Labs are usually equipped with interconnected Personal Computers (PCs). The heavy and improper utilization of the PCs in the Lab setting makes them more vulnerable and prone to the hardware as well as the software based errors.

The hardware as well as software based error in PCs in Lab settings will be difficult to diagnose, because they are not properly reported by users.

This not only degrades the performance, but also lowers the availability of the PCs in the Labs. The technical staff usually monitors PCs, and diagnoses errors in Lab settings on the regular basis or on user complaints.

The continuous monitoring of the PCs in Lab setting, the diagnosis of errors and their reporting is a frustrating and tedious task for the technical staff.

Lab Watcher is a useful tool which eases the duties of technical staff for monitoring each and every PC for errors.

Using this tool they can get error reports on real time basis.

1.2 Related Work

There are many applications which monitor computers for problems, for their resource utilization, and also monitor their performance. Some of them discuss below:

Nagios it is an open source powerful monitoring system. It monitors entire IT infrastructure of the organization to ensure systems, applications, services are functioning properly in case of a problem or failure it can alert the technical staff about the issue and allowing them to fix it. It monitors only specific attributes of the windows machines such as: Memory Usage, CPU Load, Disk Usage, Service States, running processes etc. Installation and configuration of this system is also difficult because we need to do extra work of configuration and mostly things are done manually [1]. **Windows Event log viewer** is also a useful tool which shows error logs entries occurs in system during its working. Windows applications write entry in log when information, warning, or error event occurs which is sometime very useful to diagnose and troubleshoot problems from the system. But only log error entries on local machine and administrator of the machine are responsible for checking and diagnosing the problem. Also it only monitors errors events not the resource utilization and performance of the systems, which is very crucial for computer labs [2].

1.3 Motivation

Due to fast growing technologies tasks that were done manually is now done by computers. Different software tools now available to perform different human tasks and they are making human life easier. Computer labs are monitored manually so far, technical staff monitors each and every PC for hardware and software issues, which is quite tedious task for them.

- As every PC in computer lab is connected via local area network to a central server and as every PC connected to a network and has unique system name, so it can become very easy task for technical staff if problems are reported at server side by a PC.
- Service applications running on lab PCs will not interfere other tasks performed by PC users and it'll seamlessly monitors hardware and software problems.

1.4 Purposed Solution

LAB WATCHER is solution for those technical staff; it is a tool which will monitor PCs in Lab settings for errors, diagnose and report hardware as well as software based errors on real-time basis. Lab Watcher is windows based errors as well as warning monitoring system, which not only do work of windows event viewer but also monitor hardware status and actively report problems to technical staff. It uses Win32_Classes of Computer Hardware System, to monitor their availability, error configuration codes and status as well [6].

When error occur it send error description to server side web application where technical staff is responsible for troubleshooting problem from computer. Also System is easily manageable and easy to operate, which will provide ease to technical staff.

It mainly consists of two parts:

1. **PC Monitoring service** which monitors a PC in a lab for hardware as well as software based errors. It not only monitors PC for errors but also report their status.
2. **Online web application** it provide detail view of those error reports to technical staff currently logged in.

1.5 Report Overview

Chapter 2: Requirements Specification

In this chapter Specific requirements of the system are specified comprehensively, also functional as well as non-functional requirements of the system.

Chapter 3: Requirements Analysis

In this chapter requirements analysis is carried out in the form of Use Case diagram and use case description of each use case, also stakeholders of the system are specified.

Chapter 4: System Detail Design

This chapter consists of system design in detail. It contains system class diagram, sequence diagrams which depicts actual flow of the system and also database design as well.

Chapter 5: Implementation

This chapter contains system implementation details such as platform selection, programming language used, important algorithms.

Chapter 6: Testing

This chapter contains User Acceptance Test specification which is built against use cases as described in Requirements Analysis chapter.

Chapter 7: Conclusion and Future Work

This chapter concludes the project and also contains details of future work related to this project.

Definitions, Acronyms and Abbreviations

Definitions of terms, acronyms and abbreviations used in this report are given in this section.

References

At last references related to each chapter which were helpful in system development and documentation are given.

Chapter 2

Requirements Specification

This chapter describes SRS of the system, mainly focused on functional requirements and non-functional requirements, details of system perspectives, system functions, user characteristics, logical database requirements and some general constraints, design constraints as well as dependencies.

2.1 Introduction

This section describes SRS purpose, Objectives, Scope of LAB WATCHER System.

2.1.1 SRS Purpose

Purpose behind writing this SRS is that let other developers understand the basic requirements of LAB WATCHER System. Also this SRS will be helpful to both customers and developer of the system to agree upon requirements of the system to avoid any ambiguities or confusion in future.

2.1.2 Objectives

The objectives of the LAB WATCHER System are:

- To Monitor PCs for Hardware & Software based errors.
- To provide management of error reports and PCs information to administrator and technical staff.

2.1.3 Scope

Scope of the system is as follows:

Function & Features:

- This system will provide ease to technical staff working in lab to monitor each & every PC for errors that degrade PCs performance.
- System will actively monitor PCs for most common hardware and software based errors.
- System will report technical staff whenever error occurs in Lab PCs and notify the status of the PCs.
- It will also provide facility to technical staff for managing these error reports.

Data input/output:

- **Inputs:**

  Login details and name of Technical Staff & Administrator.
- **Outputs:**

  Error reports of affected PCs.

  PCs information, such that system specification & hardware statuses.

Constraints:

- Only technical staff & Administrator can view & manage error reports
- Only administrator can manage technical staff and has full access of the system.

2.2 Product Perspectives

The product LAB Watcher System it has one service which is somehow similar to the services running seamlessly on computers. But some of its features make it unique from them.

2.2.1 Concept of Operations

The LAB Watcher system will allow two different methods of operations. The first method of operation will be through LAB PC. This operation will be performed by PC Monitoring Service which will seamlessly monitors the PC, without user intervention, and report these errors to second part of the system. This service will be automatic and startup whenever system boots, or when it is installed and stop whenever system shutdown.

Second method of operation will be through online web application which will be running on server. Operations on this web application will be done by administrator and technical staff. Technical staff is added by system administrator and they can view and manage error reports. While administrator can manage technical staff and have full access of the system.

2.2.2 User Interfaces

- PC Monitoring Service has only user interaction when technical staff or administrator install or uninstall this service on lab computer when it deployed it has no user interface; it will seamlessly monitor the personal computer.
- Online web application has web interface, it is for administrator and technical staff.
- Web interface contains login interface, view of error reports and view of lab computers.

2.2.3 Hardware Interfaces

Followings are the minimum hardware requirements for the LAB WATCHER.

- CPU
- Monitor
- Keyboard, Mouse
- RAM
- Hard Disk
- LAN Connection

2.2.4 Software Interfaces

Following are the minimum software requirements for the LAB WATCHER.

- Windows Operating System: XP/Windows 7/Windows 8
- .NET Framework 4 or on ward
- Firefox
- Chrome
- Notepad++
- Apache Server
- MySQL

2.2.5 Communication Interfaces

The client (LAB PC) and server (Online Web Application) will communicate with each other through local area network connection. For this connection oriented protocols will be useful.

2.3 Product Functions

Followings are the functions performed by the system:

PC Monitoring Service

PC Monitoring Service will perform following functionality:

Hardware Monitoring:

- Detect Keyboard Status and Error Configuration Code
- Detect Mouse Status and Error Configuration Code
- Detect Monitor Status and Error Configuration Code
- Detect RAM Status and Error Configuration Code
- Detect Hard disk Status and Error Configuration Code
- Detect Processor Status and Error Configuration Code
- Detect Hard disk Usage
- Detect RAM Usage

Software Monitoring:

- Detect windows event viewer application log for errors and warnings.
- Detect windows event viewer system log for errors and warnings.
- Detect low virtual memory error and warning.
- Detect low physical memory error and warning.
- Detect low hard disk storage error and warning.
- Detect running and maximum processes an operating system support.

System Monitoring:

- Detect Operating System information and send it to web application.
- Detect Processor Information and send it to web application.
- Detect Hard disk Information and send it to web application.
- Detect RAM Information and send it to web application.
- Detect PC Monitoring Service Status and send it to web application.

Online Web Application

Online Web Application will provide following functionality:

- View of Computers Specifications
- View of Hardware Error Reports
- View of Software Error Reports
- Login to administrator and technical staff.
- Management of technical staff to administrator.
- Generation of reports of errors and computers.
- Update Profile for technical staff and administrator.

2.4 User Characteristics

Main users of this system are LAB Administrator and Technical Staff who work in computer labs. Their responsibilities are to monitor and manage each and every PC in computer lab for errors.

These technical staffs have sufficient knowledge of computer errors troubleshooting. Whenever error reported by this system to them they can easily diagnose and troubleshoot errors to make Lab PCs available to lab users.

2.5 Assumptions & Dependencies

Followings are assumptions and dependencies of LAB Watcher System:

- PC Monitoring Service will be Windows based system only. If in future it will be expended to UNIX or Linux operating system then it should be redesigned.
- Although system deals with most common hardware and software errors occurs in lab settings but in future if up gradation required then system will be updated.

2.6 General Constraints

Following are constraints of developing this system:

- PC Monitoring Service will be seamless to general users of LAB PCs, only technical staff or administrator has rights to install or uninstall, start or stop the service.
- Only LAB Administrator and technical staff can view and manage error reports.
- PC Monitoring Service will not influence the PC performance and user normal tasks.
- PC Monitoring Service will only monitor hardware and software based errors and also system specifications that are mentioned in product function section of this chapter.

2.7 Specific Requirements

This section describes functional as well as non-functional requirements of the system. Also it describes design constraints and logical database requirements.

2.7.1 Functional Requirements

Functional requirements are those requirements which system must provide:

- System should provide a login form where users (administrator and technical staff) can login to the system and also have logout option.
- System should provide a setup wizard using which users (administrator and technical staff) can install or uninstall the PC Monitoring Service.
- System should provide staff management facility to administrator so that administrator can add, view, edit and delete technical staff.
- System should provide view and management of Error reports so that users (administrator and technical staff) can view and delete error reports.
- System should provide update profile feature to users (administrator and technical staff).
- System should provide generate report facility to users (administrator and technical staff) so that they can view specific error reports and computers information.

2.7.2 Non-Function Requirements

Followings are non-functional requirements of the system:

2.7.2.1 Performance

- System will be highly efficient and will be event based.
- System will not affect overall performance of the LAB PC.

2.7.2.2 Reliability

- System will be reliable and actively monitors for hardware and software errors.

2.7.2.3 Availability

- PC monitoring service will be available and start running whenever system boot, and will be available until pc shutdown.
- Online web application will be available to administrator and technical staff round the clock whenever server is running.

2.7.2.4 Security

- PC monitoring service will be highly secure only users with administrator rights on the PC can start, stop or install, uninstall this service.
- Online web application has login authentication before accessing dashboard.

2.7.2.5 Maintainability

- System will be easily maintainable, administrator and technical staff can easily manage (start/stop) service, view and delete error reports.

2.7.3 Logical Database Requirements

MySQL database will be used for following purpose:

- Login authentication of administrator and technical staff.
- Storing user (Administrator and Staff) records.
- Storing error reports.
- Storing PC Information.

Chapter 3

Requirements Analysis

This chapter describes detail requirement analysis of the System in the form of Use Case Identifications, Use Case Diagram and Use Case Description of each use case.

3.1 Stake Holders

Stake holders of the LAB WATCHER System are:

- Administrator
- Technical Staff of LAB

3.2 Use Case Identification

- Login
- Logout
- Run PC Monitoring Service.
- View Reported Errors.
- Delete Reported Errors.
- View Computers Information.
- Generate Reports.
- Update Profile.
- Add LAB Staff.
- Edit LAB Staff.
- View LAB Staff.
- Delete LAB Staff.

3.3 Use Case Diagram

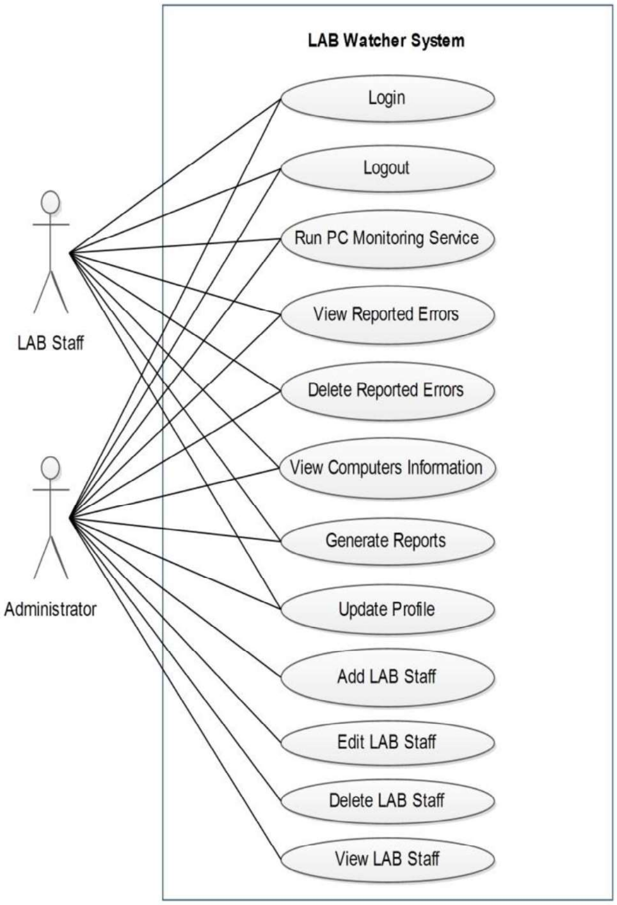


Figure 1. Use Case Diagram

3.4 Use Case Descriptions

“All use cases are written in detail where detail includes use cases description, Primary actor, pre-condition, post-condition, success scenario and extensions [6]”.

Use Case ID	Use Case# 1
Use Case Name	Login
Stakeholder	Administrator, Technical Staff
Pre-Condition	Login page is open and User is member of the System.
Flow of Events	1. User enters email address. 2. User enters password. 3. User select login.
Success Scenario	All Information entered is valid and access is granted to User.
Exceptions	If correct email and password is not entered error message is displayed.
Post-Condition	System will show dashboard page to user.

Use Case ID	Use Case# 2
Use Case Name	Logout
Stakeholder	Administrator, Technical Staff
Pre-Condition	User is logged in the System.
Flow of Events	1. User select logout.
Success Scenario	User successfully logged out.
Exceptions	None
Post-Condition	User successfully logged out from the System.

Use Case ID	Use Case# 3
Use Case Name	Run PC Monitoring Service
Stakeholder	Administrator, Technical Staff
Pre-Condition	Computer is running and connected to server with LAN connection
Flow of Events	1. User select setup option. 2. User follows setup wizard instructions. 3. User select finish option to complete installation.
Success Scenario	PC Monitoring Service will be started successfully.
Exceptions	Error will occur during installation and setup will stop and service will be removed.
Post-Condition	PC Monitoring Service will be started.

Use Case ID	Use Case# 4
Use Case Name	View Reported Errors
Stakeholder	Administrator, Technical Staff
Pre-Condition	User is logged in the system.
Flow of Events	1. User select error management option. 2. User select all software/hardware errors 3. User select view option
Success Scenario	System shows error message and update error message status as viewed.
Exceptions	System does not update status of error message as viewed.
Post-Condition	Error message is viewed.

Use Case ID	Use Case# 5
Use Case Name	Delete Reported Error
Stakeholder	Administrator, Technical Staff
Pre-Condition	User is logged in the system.
Flow of Events	1. User select error management option. 2. User select all software/hardware errors 3. User select delete option
Success Scenario	System shows success message that error message deleted successfully.
Exceptions	System shows error message that error message did not deleted successfully.
Post-Condition	Error message is removed from the system.

Use Case ID	Use Case# 6
Use Case Name	View Computers Information
Stakeholder	Administrator, Technical Staff
Pre-Condition	User is logged in the system.
Flow of Events	1. User select resource monitoring section 2. User select view connected computers option.
Success Scenario	System will list of connected computers and their information.
Exceptions	System will display error message that no computer is connected and no record found.
Post-Condition	List of computers and information is displayed

Use Case ID	Use Case# 7
Use Case Name	Generate Reports
Stakeholder	Administrator, Technical Staff
Pre-Condition	User is logged in the system.
Flow of Events	1. User select report generation section 2. User select error type 3. User select computer 4. User select search option
Success Scenario	System will list of errors against the selected computer and of selected type.
Exceptions	System will display message that no record found for this computer.
Post-Condition	List of errors and computers information displayed

Use Case ID	Use Case# 8
Use Case Name	Update Profile
Stakeholder	Administrator, Technical Staff
Pre-Condition	User is logged in the system.
Flow of Events	1. User selects update profile option. 2. User edit name. 3. User edits password. 4. User confirms password 5. User selects Update Record option.
Success Scenario	System will display success message that user record updated successfully.
Exceptions	System will display following exceptions on in-correct entry of record: o Wrong password entered. o User record can’t be updated.
Post-Condition	User record updated successfully.

Use Case ID	Use Case# 9
Use Case Name	Add LAB Staff
Stakeholder	Administrator
Pre-Condition	User is logged in the system.
Flow of Events	1. User select user management section. 2. User select add user option. 3. User enters name of the technical staff. 4. User enters email address. 5. User enter password. 6. User confirms password. 7. User selects Add Record option.
Success Scenario	System will display success message that new user record added successfully.
Exceptions	System will display following exceptions on in-correct entry of record: o User already added in the system. o Wrong password entered. o Invalid email entered. o User record can't be added in the system.
Post-Condition	New user record added successfully.

Use Case ID	Use Case# 10
Use Case Name	View LAB Staff
Stakeholder	Administrator
Pre-Condition	User is logged in the system.
Flow of Events	1. User selects user management section. 2. User select view users option
Success Scenario	System will display list of LAB Staff exists in the system.
Exceptions	System will display message that no LAB Staff record exist.
Post-Condition	User has viewed the LAB Staff.

Use Case ID	Use Case# 11
Use Case Name	Edit LAB Staff
Stakeholder	Administrator
Pre-Condition	User is logged in the system.
Flow of Events	1. User selects user management section. 2. User select view users option 3. User select edit user option of a particular LAB Staff. 4. User edit name of the LAB staff. 5. User edits email address. 6. User edits password. 7. User confirms password 8. User selects Update Record option.
Success Scenario	System will display success message that LAB Staff record updated successfully.
Exceptions	System will display following exceptions on in-correct entry of record: o User already exists in the system. o Wrong password entered. o Invalid email entered. o User record can’t be updated.
Post-Condition	LAB Staff record updated successfully.

Use Case ID	Use Case# 12
Use Case Name	Delete LAB Staff
Stakeholder	Administrator
Pre-Condition	User is logged in the system.
Flow of Events	1. User selects user management section. 2. User select view users option 3. User select a particular LAB Staff record 4. User select delete record option. 5. User confirm delete of record.
Success Scenario	System will display success message that record deleted successfully.
Exceptions	System will display error message that record can’t be deleted.
Post-Condition	LAB Staff record has been removed from the system.

Chapter 4

System Detail Design

This chapter consists of system design in detail. System design is basically the plan of how the project under consideration will become a reality i.e. in executable form. System design also contains system class diagram, sequence diagram which depicts actual flow of the system and also database design as well.

4.1 Software Architecture

Software architecture intuitively denotes the high level structures of a software system. It represents software elements, there relationship between each other, and their properties as well.

There are three main components of the system, which includes:

1. Windows Service
2. Web Application
3. Database

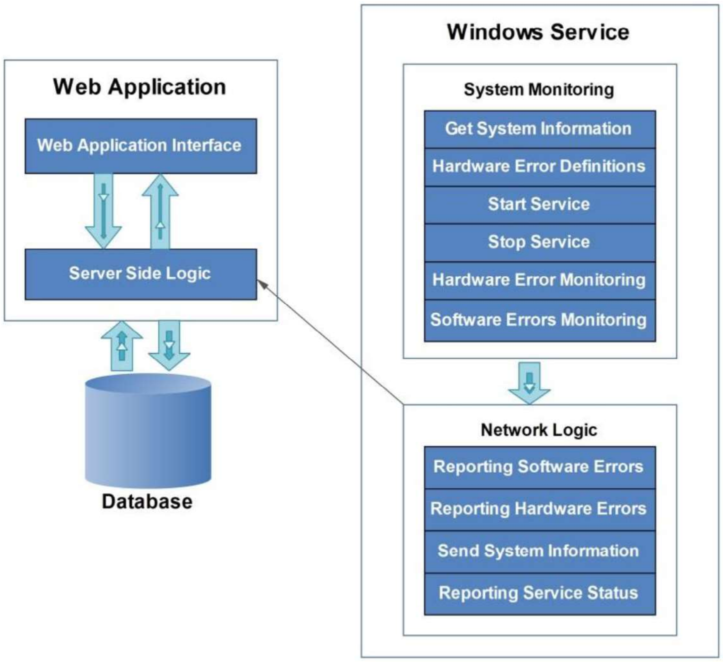


Figure 2. Software Architecture Diagram

4.2 Database Design

In this system database is used for storing Administrator and Technical Staff login details PC’s information, software and hardware errors reports.

Following are entities of the system:

- User
- Computer
- Error

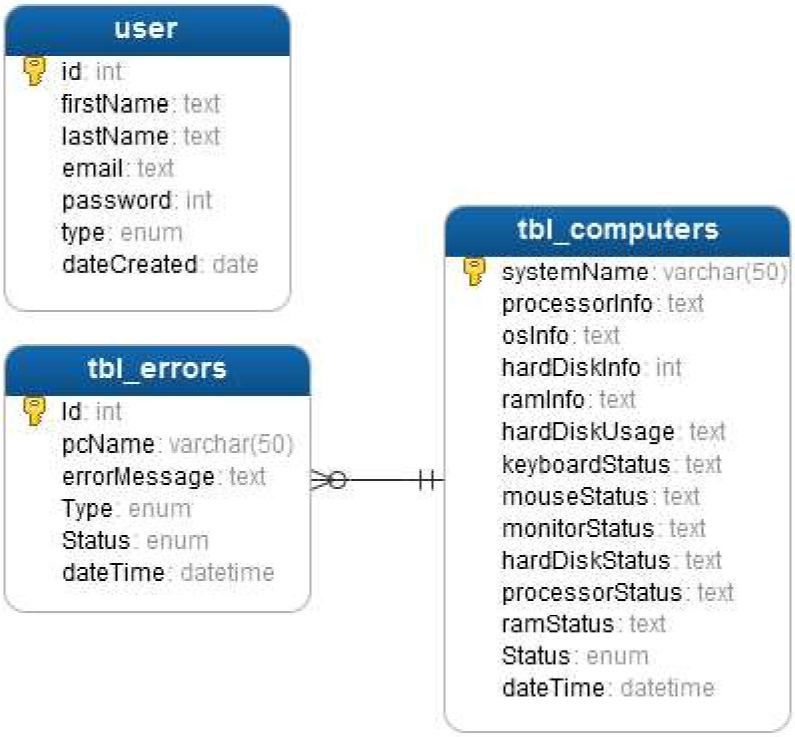


Figure 3. Physical Database Design

4.3 Class Diagram

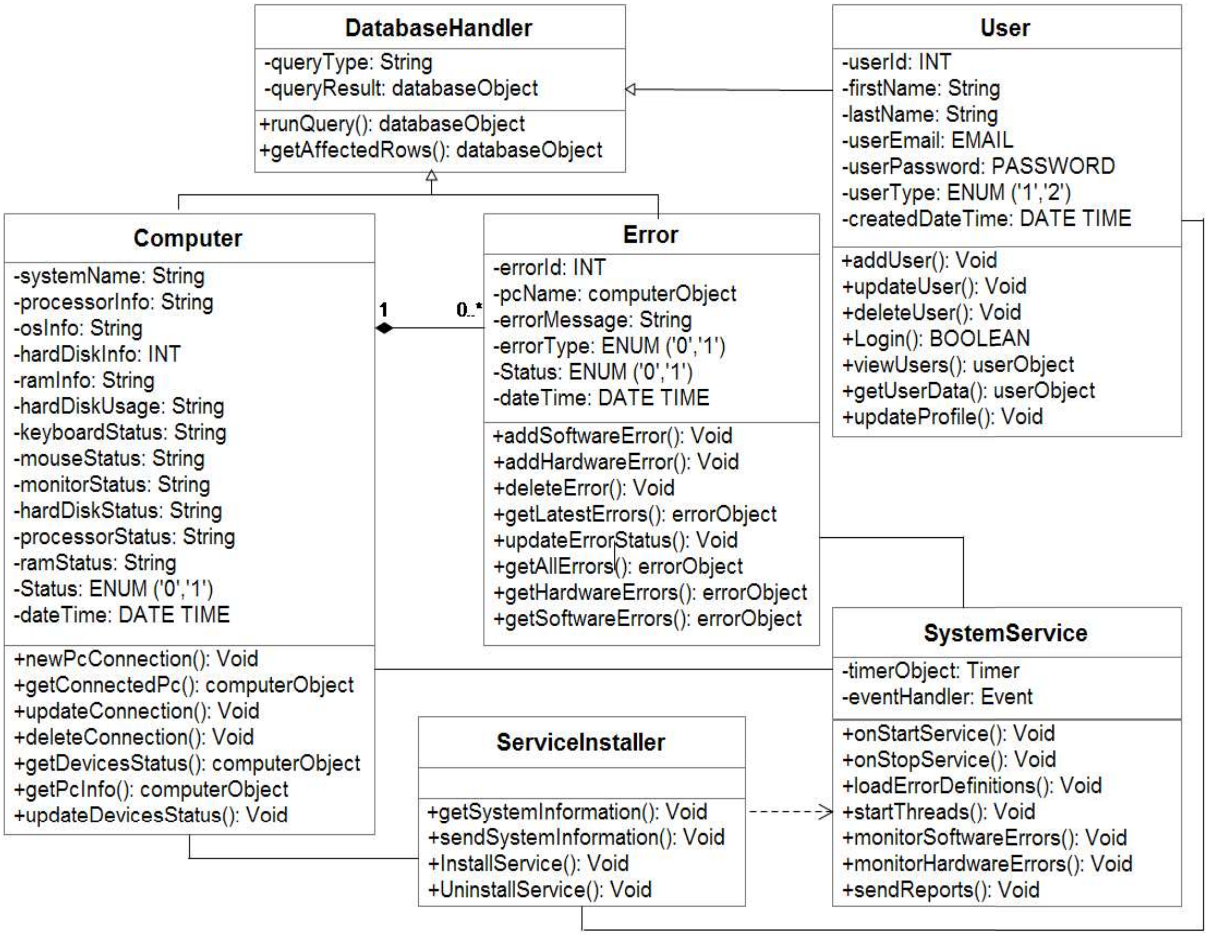


Figure 4. Class Diagram

4.4 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.

Sequence diagram of some important use cases are here.

Use Case 3: Run PC Monitoring Service

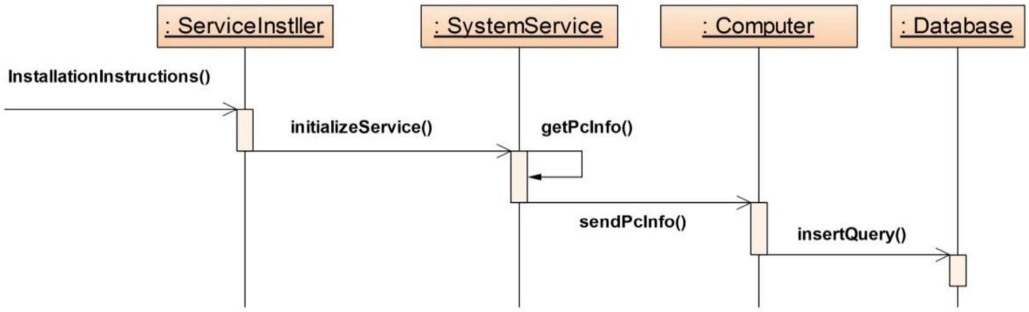


Figure 5. Sequence Diagram

Use Case 4: View Reported Errors

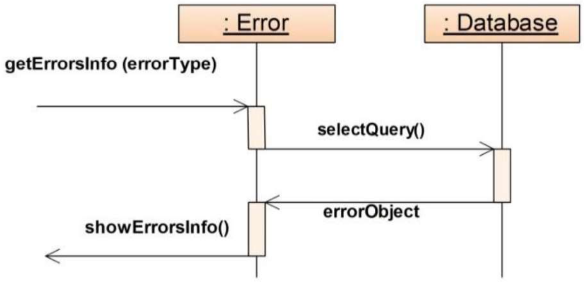


Figure 6. Sequence Diagram

Use Case 5: Delete Reported Errors

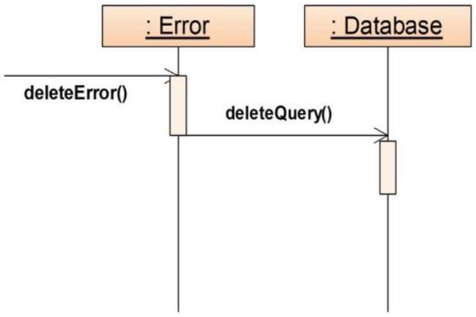


Figure 7. Sequence Diagram

Use Case 6: View Computers Information

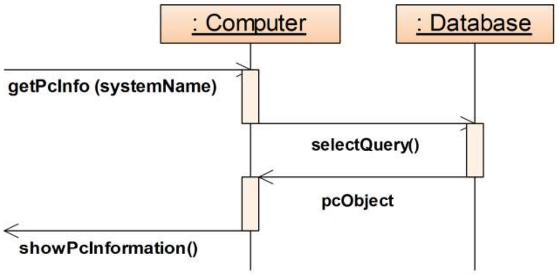


Figure 8. Sequence Diagram

4.5 Design Highlights

As there is different design approaches to develop the same application, my approach is totally Object oriented based. The reason of this approach is that by using object oriented approach reusability is increased. We can make one class and then use its instances again and again for different functionality.

Secondly is as our first part of the system is totally windows based so we can use existing windows APIs and classes for our different functions.

Finally as we want that our window service should be very light and very fast so that it doesn’t affect or degrade PC original speed and processes. For this threading and parallel execution of our code is the best option.

Chapter 5

Implementation

This chapter is related to the system implementation. This chapter mentions the tools, platform; API's used to develop the system. Some codes screen shots are also given at the end.

5.1 Introduction

After the system detail design phase, the system implementation phase comes. In this phase we decide that how we develop our system and which are the best techniques to use for implementation.

5.2 Platform Selection

As most of the computer LABs are equipped with windows based operating system so windows platform is selected for developing of PC Errors Monitoring Service.

Windows services are applications which seamlessly execute in the computer systems and perform certain tasks. These types of services are highly efficient when user has needed not to interact with the application directly and want to perform certain tasks behind the scene without user intervention.

And for our web application apache server and MySQL database is required to run server side scripts and to present the results in the form of web based interface.

5.3 Programming Language Selection

For LAB WATCHER System two main programming languages are selected. These two programming languages are different and used for different purposes.

1. **C-Sharp (C#.NET)**

For the development of PC Error Monitoring Service C#.NET is selected and developed windows service which fulfils the needs of PC Error Monitoring Service. Win32_Classes are used for monitoring and retrieving system specifications and system status.

Why C#?

C# is an elegant, simple, object-oriented programming language that allows developers to develop vast range of applications. C# also gives the capability to build durable and reliable system level applications. Win32_Classes are efficiently utilized with C# which are used to monitor and retrieve system specifications and monitor them for errors. Another reason for choosing C# is that Windows Service development is very easy in C# and it is easily configurable and deployable to the system [9].

2. **PHP**

For the development of Online Web Application PHP is used. By using PHP classes the information posted by the Windows Service were fetched and stored in the database, and then by simple database queries that information is fetched from database and presented to the user. Also management of the error reports, computer information and users is provided by developing Dashboard for administrator and technical staff. For this purpose PHP, HTML, CSS, Ajax and jQuery is used in combine.

Why PHP?

PHP is easy and web pages are easy to write and maintain because the source code and HTML are together [10].

5.4 Software Used

Followings are the tools which are used in the development of the LAB WATCHER System:

- **Notepad++**

Notepad++ is a text and source code editor for Windows based operating systems.
One advantage of Notepad++ over the built-in Windows text editor, (Notepad) is tabbed editing, which allows working with multiple open files and also perception and indentation of the source code is also maintained properly in Notepad++ [11].

- **Visual Studio 2010 IDE**

It is an integrated development environment (IDE) from Microsoft, Which is used to develop console and graphical user interface applications along with Windows Forms applications, Web sites, Web applications, and Web services [12].

- **MySQL**

MySQL is an open source relational database server. It was first developed by Sun Microsystems includes and currently being managed by Oracle Corporation. It is open source database software whose main function is to store and retrieve data as requested by other software applications. Such as web application [13].

- **XAMPP Server**

XAMPP Server is an open source cross platform web server package. It consists of Apache HTTP Server, MySQL Database Server and other utilities. X in XAMPP Server means cross platform means this can work on all platforms whether its windows based or Linux based system [13].

5.5 APIs Used

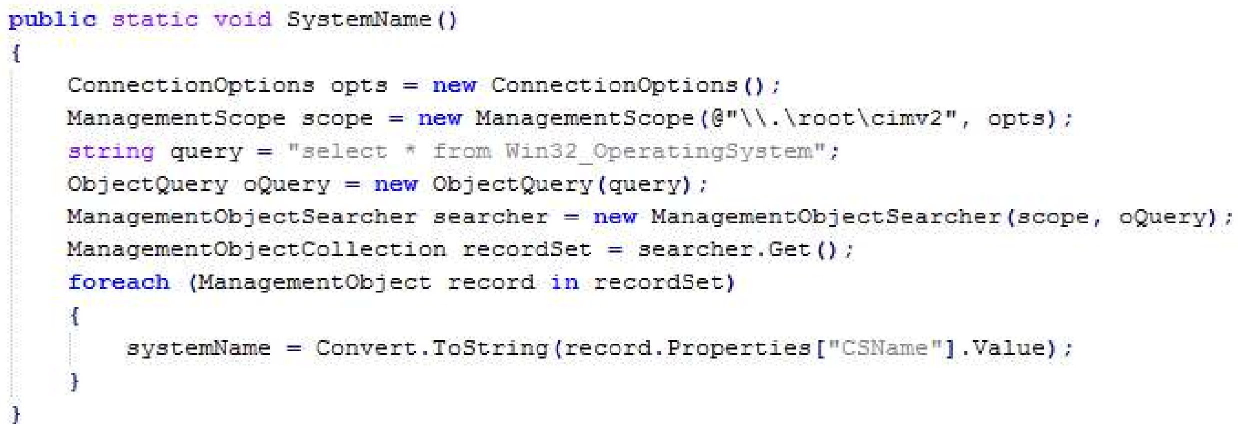
Following are APIs used in the development of LAB WATCHER System:

- **Win32_Classes** such that Computer System Hardware Classes are used for Hardware related objects, Operating System Classes are used for operating system related Objects [14].
- **WMI Queries,** WMI Queries use WMI Query language (WQL) a subset of SQL. The query string is stored in a .NET Framework query class. We can invoke these query classes ManagementObjectSearchers class constructors and get the information we want. For example: SELECT * FROM Win32_LogicalDisk WHERE FreeSpace < 2000000 [15].

5.6 Some Important Code Snapshots

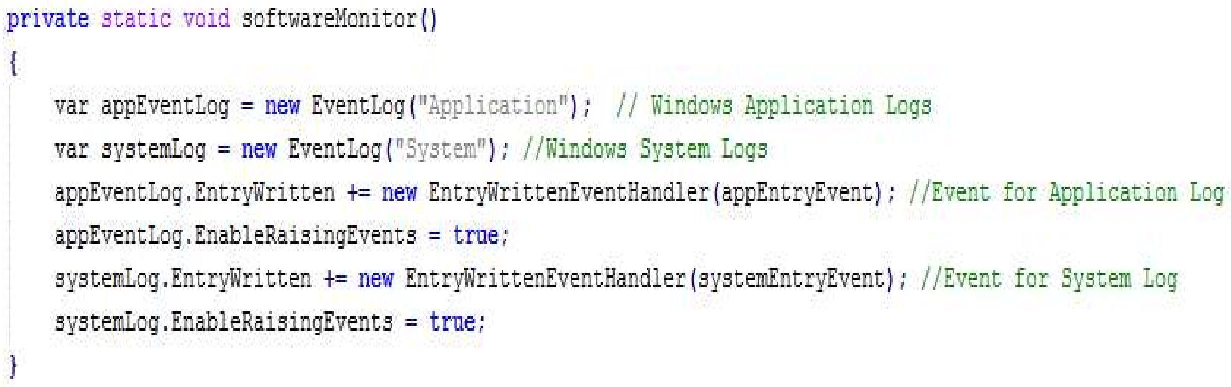
Following are the screen shots of some of the important algorithms used in LAB WATCHER System.

1. **Retrieving System Name in Windows Service**



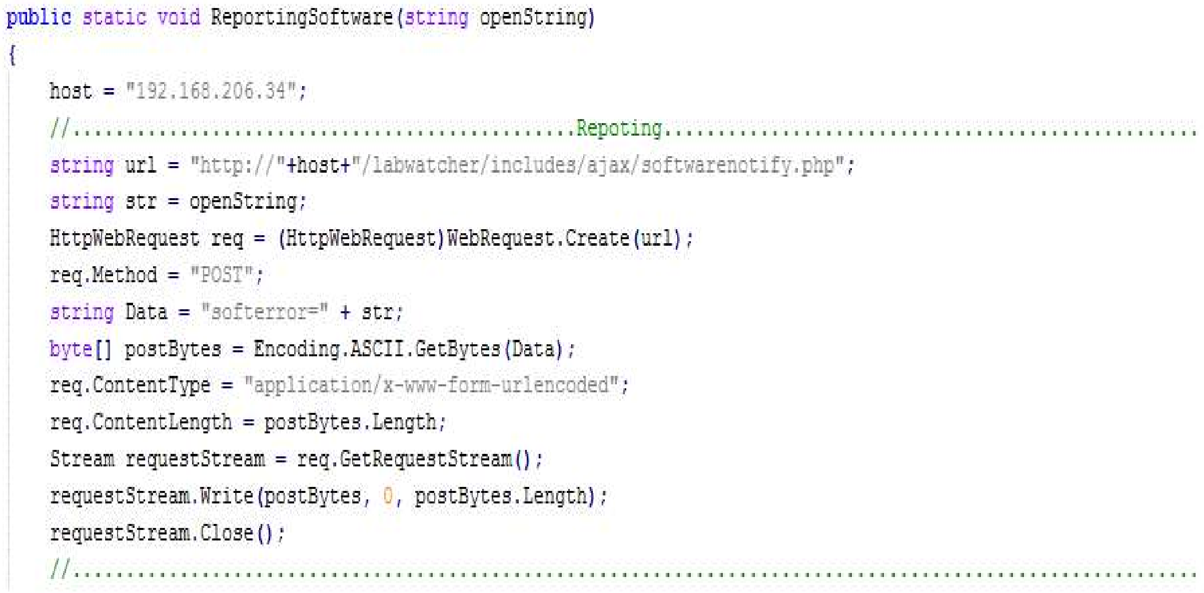
```
public static void SystemName()
{
    ConnectionOptions opts = new ConnectionOptions();
    ManagementScope scope = new ManagementScope(@"\\.\root\cimv2", opts);
    string query = "select * from Win32_OperatingSystem";
    ObjectQuery oQuery = new ObjectQuery(query);
    ManagementObjectSearcher searcher = new ManagementObjectSearcher(scope, oQuery);
    ManagementObjectCollection recordSet = searcher.Get();
    foreach (ManagementObject record in recordSet)
    {
        systemName = Convert.ToString(record.Properties["CSName"].Value);
    }
}
```

2. Software Monitoring



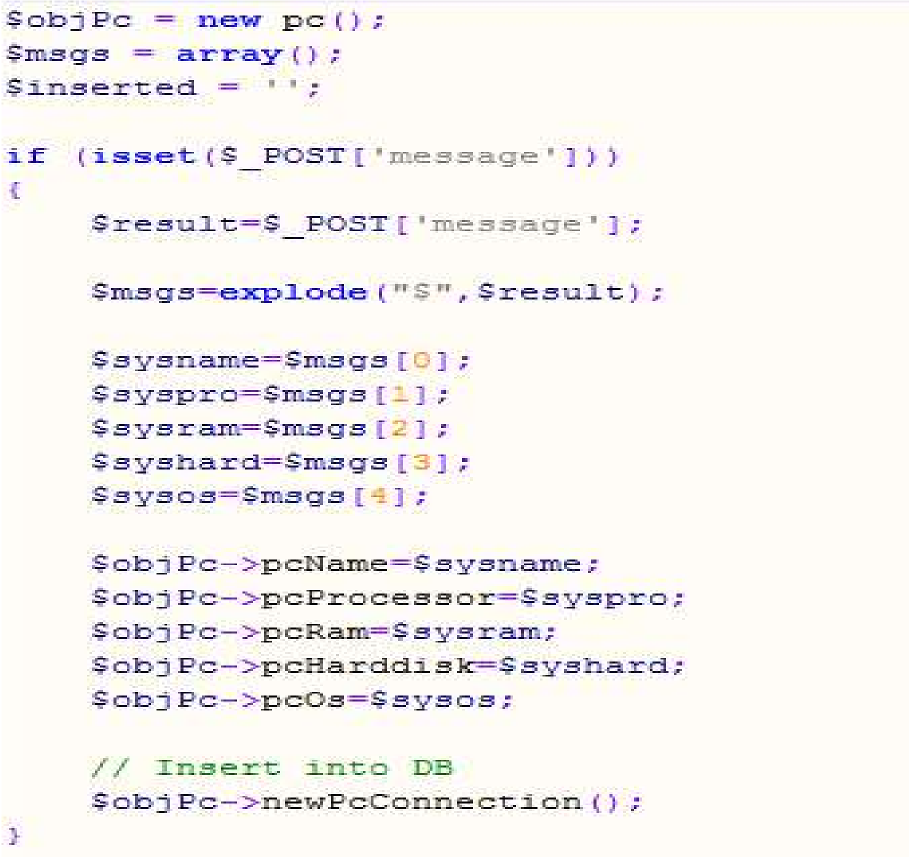
```
private static void softwareMonitor()
{
    var appEventLog = new EventLog("Application");  // Windows Application Logs
    var systemLog = new EventLog("System"); //Windows System Logs
    appEventLog.EntryWritten += new EntryWrittenEventHandler(appEntryEvent); //Event for Application Log
    appEventLog.EnableRaisingEvents = true;
    systemLog.EntryWritten += new EntryWrittenEventHandler(systemEntryEvent); //Event for System Log
    systemLog.EnableRaisingEvents = true;
}
```

3. Reporting Software Errors



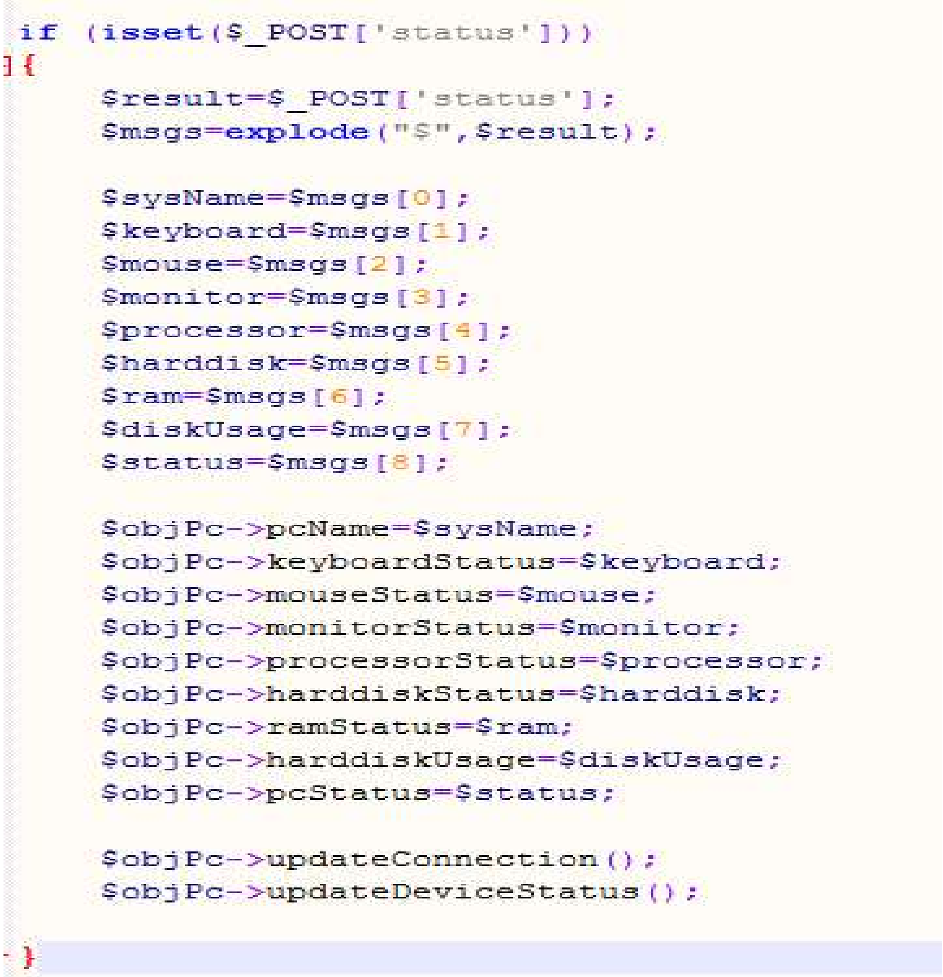
```
public static void ReportingSoftware(string openString)
{
    host = "192.168.206.34";
    //...............................................Repoting..................................................
    string url = "http://"+host+"/labwatcher/includes/ajax/softwarenotify.php";
    string str = openString;
    HttpWebRequest req = (HttpWebRequest)WebRequest.Create(url);
    req.Method = "POST";
    string Data = "softerror=" + str;
    byte[] postBytes = Encoding.ASCII.GetBytes(Data);
    req.ContentType = "application/x-www-form-urlencoded";
    req.ContentLength = postBytes.Length;
    Stream requestStream = req.GetRequestStream();
    requestStream.Write(postBytes, 0, postBytes.Length);
    requestStream.Close();
    //.........................................................................................................
```

4. Adding Computer Information On installation of Service



```
$objPc = new pc();
$msgs = array();
$inserted = '';

if (isset($_POST['message']))
{
    $result=$_POST['message'];

    $msgs=explode("$",$result);

    $sysname=$msgs[0];
    $syspro=$msgs[1];
    $sysram=$msgs[2];
    $syshard=$msgs[3];
    $sysos=$msgs[4];

    $objPc->pcName=$sysname;
    $objPc->pcProcessor=$syspro;
    $objPc->pcRam=$sysram;
    $objPc->pcHarddisk=$syshard;
    $objPc->pcOs=$sysos;

    // Insert into DB
    $objPc->newPcConnection();
}
```

5. Status of Computer Devices and Updating Service Status



```
if  (isset($_POST['status']))
{
        $result=$_POST['status'];
        $msgs=explode("$",$result);

        $sysName=$msgs[0];
        $keyboard=$msgs[1];
        $mouse=$msgs[2];
        $monitor=$msgs[3];
        $processor=$msgs[4];
        $harddisk=$msgs[5];
        $ram=$msgs[6];
        $diskUsage=$msgs[7];
        $status=$msgs[8];

        $objPc->pcName=$sysName;
        $objPc->keyboardStatus=$keyboard;
        $objPc->mouseStatus=$mouse;
        $objPc->monitorStatus=$monitor;
        $objPc->processorStatus=$processor;
        $objPc->harddiskStatus=$harddisk;
        $objPc->ramStatus=$ram;
        $objPc->harddiskUsage=$diskUsage;
        $objPc->pcStatus=$status;

        $objPc->updateConnection();
        $objPc->updateDeviceStatus();

}
```

Chapter 6

Testing

This chapter contains User acceptance testing which are built against the use cases described in Requirements Analysis chapter.

6. 1 Introduction

Testing plays an important role, after implementation of the System to check whether system developed fulfil the requirements stated in Requirements Specification and Requirements Analysis phase. Testing is also useful to uncover requirements related issues. User Acceptance Test is very useful because it provide actual functions, inputs to the system, expected results from the system, actual results, procedures to perform test cases, pass/fail status of the test and date against each test case.

6.2 Test Cases

Test cases contains following information:

- Test Case ID
- Function
- Tester
- Setup
- Procedure
- Expected Results
- Actual Results
- Status (Fail/Pass)

Test Case # 01	Login
Function	This test case verifies that user can login to system successfully.
Tester	User
Setup	Open “Login” page
Procedure	1. Enter email address “abc@example.com” 2. Enter password “123456” 3. Click on login button.
Expected Results	User Dashboard will be shown.
Actual Results	User Dashboard is shown.
Status (Pass/Fail)	Passed.

Test Case # 02	Logout
Function	This test can verifies that user can logout from the system successfully.
Tester	User
Setup	Login to system
Procedure	1. Click logout option from dashboard menu.
Expected Results	Users will logout and login page will be shown.
Actual Results	User logged out from the system and login page shown.
Status (Pass/Fail)	Passed.

Test Case # 03	Run PC Monitoring Service
Function	This test case can verifies that user can successfully run PC Monitoring Service.
Tester	User
Setup	Start the PC
Procedure	1. Runs service setup. 2. Follows instructions and Click “Next”. 3. Clicks “Finish” option to complete installation.
Expected Results	PC Monitoring Service will be installed successfully and started running.
Actual Results	PC Monitoring Service Installed and started successfully.
Status (Pass/Fail)	Passed

Test Case # 04	View Reported Errors
Function	This test can verifies that user can view reported errors successfully.
Tester	User
Setup	Open ‘Dashboard’ main Page
Procedure	1. View latest error message 2. Check the status of error message to view.
Expected Results	Reported error message status will be changed to view.
Actual Results	Reported error message status is changed to view.
Status (Pass/Fail)	Passed

Test Case # 05	Delete Reported Errors
Function	This test case test whether user can delete reported errors from the system successfully.
Tester	User
Setup	Open ‘Errors Manager’ Section
Procedure	1. Click on “All Software Errors” link 2. Selects an error message and click delete option. 3. Confirm delete option
Expected Results	Error message will be deleted from the system and success message will be displayed.
Actual Results	Error message is deleted from the system and success message is displayed that error is deleted.
Status (Pass/Fail)	Passed

Test Case # 06	View Computers Information
Function	This test case verifies whether user can view computers information successfully.
Tester	User
Setup	Open 'Resource Manager' Section
Procedure	1. Click connected computers link 2. Click computer name to view system specifications 3. User click hardware status link to view hardware status
Expected Results	List of connected computers and their specification and hardware status will be shown.
Actual Results	List of connected computers and their specification and hardware status is shown.
Status (Pass/Fail)	Passed

Test Case # 07	Generate Reports
Function	This test verifies whether user can generate reports successfully or not.
Tester	User
Setup	Open 'Generate Report' page
Procedure	1. Click generate reports link 2. Select error type 'All Errors' 3. Select connected computer 'Mudassar-PC' 4. Click search button
Expected Results	All reported errors of selected computer will be shown.
Actual Results	All reported errors of selected computer shown.
Status (Pass/Fail)	Passed

Test Case # 08	Update Profile
Function	This test case verifies that user can successfully update profile.
Tester	User
Setup	Open ‘Settings’ page
Procedure	1. Click update profile link 2. Update the name to ‘Asim’ 3. Update the password to ‘123456’ 4. Confirm the password to ‘123456’ 5. Click on update record link
Expected Results	User record will be updated successfully and success message will be displayed.
Actual Results	Record updated successfully and success message is displayed.
Status (Pass/Fail)	Passed.

Test Case # 09	Add LAB Staff
Function	This test case verifies that user can add LAB Staff successfully.
Tester	User
Setup	Open ‘User Manager Section’
Procedure	1. Click on add lab staff link 2. Enter “Asim” as first name of the staff 3. Enter “Khan” as last name of the staff 4. Enter ‘abc@example.com’ as the email of the staff 5. Enter ‘123456’ as the password 6. Enter ‘123456’ as the confirm password 7. Press Add User button.
Expected Results	LAB Staff record will be added successfully and success message will be displayed
Actual Results	Record is inserted and success message is displayed.
Status (Pass/Fail)	Passed

Test Case # 10	Edit LAB Staff
Function	This test case verifies that user can successfully edit LAB Staff record.
Tester	User
Setup	Open ‘view users’ page
Procedure	1. Click on edit link for a lab staff 2. Edit first name from “Asim” to “Alam”. 3. Edit last name from “Khan” to “KHAN” 4. Edit email from ‘abc@example.com’ to ‘alam@live.com’ 5. Edit password from ‘123456’ to ‘54321’ 6. Edit confirm password ‘123456’ to ‘54321’ 7. Press Update User button.
Expected Results	User record updated successfully and success message will be displayed.
Actual Results	Record updated successfully and success message is shown.
Status (Pass/Fail)	Passed

Test Case # 11	View LAB Staff
Function	This test case verifies that user can successfully view LAB Staff record.
Tester	User
Setup	Open ‘User Manager Section’
Procedure	1. Click on view users
Expected Results	List of lab staff will be displayed.
Actual Results	List of lab staff is displayed.
Status (Pass/Fail)	Passed

Test Case # 12	Delete LAB Staff
Function	This test case verifies that user can successfully delete LAB Staff record.
Tester	User
Setup	Open ‘User Manager Section’
Procedure	1. Click on view users 2. Select a user and click on delete option 3. Click confirm delete option
Expected Results	Record of the lab staff will be successfully deleted from database.
Actual Results	Record of the lab staff is deleted successfully.
Status (Pass/Fail)	Passed.

Chapter 7

Conclusions and Future work

This chapter concludes the project and also highlights future improvements which can be made in the system to enhance its performance and benefits.

7.1 Conclusion

LAB WATCHER System is the computer labs error and resource monitoring system, which not only monitor hardware and software based errors but also provide some resource monitoring like Hard Disk Monitoring, RAM Status, Keyboard, Mouse, Monitor and Processor status as well.

Beside this core functionality it provides easy management of error reports and computers information and also lab staff management facility to administrator.

Users of this system can also generate or filter error reports and computer information. Also users can update their profile for their own password and name.

Windows Service of the system is easily maintainable and highly secure because only
PC administrator has the privileges to install/uninstall, stop/start the service.
Also service is monitoring the PC seamlessly without affecting normal operations of the computer system hence PC will be available for normal usage to LAB Users.

7.2 Future work

This system can be enhanced further in lots of ways. As presently its main focus is on common hardware and software errors monitoring and some basic resource monitoring, It can be enhanced to real time resource monitoring of hardware of the computer system efficiently and also software processes running on the remote computers. Remote PC Administration facility and more system events monitoring can be added to the system to affectively monitor the computer labs.

Definitions, Acronyms and Abbreviations

Term	Description
LAB	It is acronym for ‘Laboratory’. Here it represent computer lab.
PC	It is acronym for ‘Personal Computer’.
Monitoring Service	It is operating system service which will seamlessly monitors PCs.
Technical Staffs	Lab technical staffs that monitor and trouble shoot lab computers.
Error Reporting	Sending error messages to technical staffs.
Web Application	Website application which runs on server side.
Application Log Errors and Warnings	The Application log contains events logged by applications or programs. For example, a database program might record a file error in the application log.
System Log Errors and Warnings	The System log contains events logged by Operating system components. For example, the failure of a driver or other system component to load during startup is recorded in the system log.
Operating System	System program which administrate computer system and provide useable interface to user. Here in this project we are concentrating Windows Operating System.
RAM	Random Access Memory
CPU	Central Processing Unit
WMI	Windows Management Instrumentation
API	Application Programming Interface
UML	Unified Modeling Language
IDE	Integrated Development Environment
SRS	Software Requirement Specifications

References

Chapter 1

1. Monitoring Windows Machines (2014) in Nagios Core, Retrieved on Dec 27th, 2013 from http://nagios. sourceforge.net/docs/3_0/monitoring-windows.html

2. Event logs information in Windows event viewer (2014) in Microsoft Windows, Retrieved on Dec 29th 2013 from http://windows.microsoft.com/en-us/windows/what-information-event-logs-event-viewer#1TC=windows-7

3. Win 32 Classes (2014) In Microsoft Developer Network. Retrieved on Jan 2, 2014, from http://msdn.microsoft.com/en-us/library/aa394084(v=vs.85).aspx

Chapter 2

4. Software Requirement Specification (2014) in Wikipedia, Retrieved on Jan 14th ,2014 from http://en.wikipedia.org/wiki/Software_requirements_specification

Chapter 3

5. Craig Larman, (2002). Applying UML and Patterns

6. Partap K.J.Mohapatra (2010), Software Engineering lifecycle approach, new age international publishers.

Chapter 4

7. Craig Larman, (2002). Applying UML and Patterns

Chapter 5

8. Why Use C#. (2014), In Microsoft Developer Network. Retrieved on Feb 15th ,2014, from http://msdn.microsoft.com/en-us/library/aa664274(v=vs.71).aspx

9. Why Use PHP. (2014), In PHP Manual, Retrieved on Feb 15th, 2014 from http://php.net/manual/en/

10. Notepad++ (2014), In Notepad++ website, Retrieved on Feb 16th, 2014 from http://notepad-plus-plus.org/

11. Visual Studio (2010), In Microsoft Visual Studio, Retrieved on Feb 18th,2014 from http://www.visualstudio.com/

12. WMI (2014), In Microsoft Developer Network. Retrieved on Feb 25th ,2014 from http://msdn.microsoft.com/en-us/library/aa393964(v=vs.85).aspx

Chapter 6

13. Software Testing (2014), In UAT Test Process, Retrieved on Feb 28th,2014 from http://www.bced.gov.bc.ca/imb/downloads/uat-test-process.pdf

14. Acceptance Test Cases (2014), In Software Acceptance Test Case Writing. Retrieved on March 1st, 2014 from http://www.tbs-sct.gc.ca/emf-cag/acceptance/proc/atp-per-eng.asp

THE END