



Information Security Management System for Lankem Ceylon PLC

Balanathan Virupasan

BIT Registration number: R091573

Index number: 0915734

Name of the supervisor:

J.H.A.U.S Jayarathne

Academic Year: 2017



**This dissertation is submitted in partial fulfillment of the requirement of the
Degree of Bachelor of Information Technology (External) of the
University of Colombo School of Computing.**

DECLARATION

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for inter-library loans, and for the title and summary to be made available to outside organizations.

B.G.

Date : 2017/11/06

Balanathan Virupasan

(Name of the Candidate)

Countersigned By:

J.H.A.U.S. Jeyaratne

Date : 2017/11/06

J.H.A.U.S. Jeyaratne

(Name of the Supervisor)

ABSTRACT

Lankem Ceylon PLC is a well-established public listed company in Sri Lanka involved in manufacturing and distribution of Agro chemicals, Paints, Industrial chemicals, Bitumen and consumer products. Lankem was created by Royal Dutch Shell in 1964. It was initially set up as an agro chemical business. As Sri Lanka started encouraging local ownership, the company acquired local investors and was listed on the Colombo Stock Exchange.

Lankem Ceylon IT Department adopted ISO/IEC 27001:2013 Information Security Management standard in October 2016. Since the certification is audited and reviewed by Bureau Veritas, UK on regular intervals. Information Security Team should maintain various documents (Policies, Procedures), logs (Backup logs, Disaster recovery drill, Fire Evacuation drill, Server Monitoring logs, Server room access log, and etc.), Information Security Incident reporting and etc. Currently all the information's are maintained in Hard Printed copies and MS Excel.

With the developed web-based Information Security Management System, All the information related to ISO/IEC 27001 will be maintained electronically and organized in a way to enable easy access to Information Security information for Internal IT Auditors, External IT Auditors, Bureau Veritas ISO27001 Auditors, Information Security Team, Information Security Steering Committee and Users.

Developed system achieved the client's expectation of automating certain process of Information Security Management such as Incident management operations can be managed, Information Asset's can be managed in the system, Security Logs can be managed, and ISO 27001 related documents can be managed electronically. System was developed entirely on open-source platform so its aligned with the information technology strategy of the Client.

Benefit of the developed system are Paperless Information security management and Open-Source system with potential of adding new functionality to the system.

ACKNOWLEDGEMENT

First, I would like to thank Late Professor V.K. Samaranayake for implementing this type of degree programme which is very valuable & practical for students to build their future career through Information Technology. I wish to convey my appreciation to the BIT Coordinator of University of Colombo School of Computing (UCSC) and project examination board of Bachelors of Information Technology (BIT) for giving me the opportunity to apply the knowledge gained through the BIT degree program to a real-world environment.

I would like to express my sincere gratitude to my project supervisors Mr. J.H.A.U.S Jayarathne who guided and supported me through the project work to make it as a practical solution. My special thank goes Mr. Dammika Weerasinghe, Asst. General Manager-IT of Lankem Ceylon PLC and staff of IT division for providing me all the necessary details during the project. My sincere gratitude is given to my family members who encouraged me in every time to make this project a success.

Finally, I would like to thank all my friends who helped me in many ways to complete the project. Without their encouragement and support, the completion of this project would never be achieved.

TABLE OF CONTENTS

| | |
|---|-----|
| Abstract | ii |
| Acknowledgement | iii |
| Table of Contents | iv |
| List of Figures | vii |
| List of Tables | x |
| List of Acronyms | xi |
| Chapter 1: Introduction | |
| 1.1 Introduction to Lankem Ceylon PLC | 1 |
| 1.2 Motivation for the Project | 1 |
| 1.3 Objectives and Scope of the Project | 3 |
| 1.4 Structure of the Dissertation | 4 |
| Chapter 2: Analysis | |
| 2.1 Introduction | 6 |
| 2.2 Existing System | |
| 2.2.1 ISO 27001 Documents Management | 6 |
| 2.2.2 Logs Management..... | 8 |
| 2.2.3 Incident Management | 10 |
| 2.2.4 Information Technology Asset Management | 11 |
| 2.3 Use case Diagram for the Existing System | 11 |
| 2.4 Drawbacks of Existing system | 12 |
| 2.5 Feasibility Study | |
| 2.5.1 Technical Feasibility | 13 |
| 2.5.2 Legal/Ethical Feasibility | 14 |
| 2.6 Requirements Gathering | |
| 2.6.1 Requirement Gathering Techniques | 14 |
| 2.6.2 Functional Requirements | 15 |
| 2.6.3 Non-functional Requirements | 16 |
| 2.7 Similar Type Systems | |
| 2.7.1 isoTracker | 17 |
| 2.7.2 OpenDocMan DMS | 18 |

| | | |
|-------------------------------|--|----|
| 2.7.3 | OpenKM | 19 |
| 2.7.4 | Hesk | 20 |
| 2.8 | Overview of Process Models | 20 |
| 2.9 | Process Model for the System | 21 |
| Chapter 3: Design | | |
| 3.1 | Introduction | 24 |
| 3.2 | Alternative Solutions to the System | 24 |
| 3.3 | Justification of Selected Solution | 25 |
| 3.4 | Design Technique | 25 |
| 3.5 | Use-Case Diagram for the Proposed System | 26 |
| 3.6 | Class Diagram for the Proposed System | 33 |
| 3.7 | Database Model for the Proposed System | 33 |
| 3.8 | User Interface Design | 34 |
| 3.9 | Proposed Design Architecture..... | 41 |
| Chapter 4: Implementation | | |
| 4.1 | Introduction | 43 |
| 4.2 | Implementation Environment | 43 |
| 4.3 | Reused Modules or Components | 45 |
| 4.4 | Background of the used Software's | 45 |
| 4.5 | Network Implementation | 47 |
| 4.6 | Use of Design Patterns | 47 |
| 4.7 | Major Code Sections | 48 |
| Chapter 5: Evaluation | | |
| 5.1 | Introduction | 57 |
| 5.2 | Requirements for Testing | 57 |
| 5.3 | Testing Strategies | 57 |
| 5.3.1 | Unit Testing | 58 |
| 5.3.2 | Integration Testing..... | 59 |
| 5.3.3 | Validation Testing..... | 59 |
| 5.4 | Testing Procedures | 60 |
| 5.5 | Test Plan | 60 |

| | |
|---|-----|
| 5.6 Test Data | 61 |
| 5.7 Test Cases..... | 62 |
| 5.9 System Evaluation | 67 |
| Chapter 6: Conclusion | |
| 6.1 Introduction | 69 |
| 6.2 Project Critical Assessment | 69 |
| 6.3 Problem’s encountered | 69 |
| 6.4 Future Improvements..... | 70 |
| REFERENCE | |
| REFERENCE | 71 |
| APENDIX A – System Documentations | 73 |
| APENDIX B – Design Documentation | 76 |
| APENDIX C – User Manual | 84 |
| APENDIX D – Testing Result | 99 |
| APENDIX E – Code Listing | 106 |
| APENDIX F – Client Certificate | 122 |
| GLOSSARY..... | 123 |
| INDEX..... | 126 |

List of Figures

| | | |
|-------------|--|----|
| Figure 2.1 | - Information Security Policy Document Sample Front Page | 7 |
| Figure 2.2 | - Information Security Procedure Sample Front Page | 8 |
| Figure 2.3 | - Server Utilization Monitoring log form | 9 |
| Figure 2.4 | - Backup Log Form | 9 |
| Figure 2.5 | - Incident Reporting Form | 10 |
| Figure 2.6 | - Incident Follow Up Activity Log MS Excel | 10 |
| Figure 2.7 | - Information Technology Asset's List | 11 |
| Figure 2.8 | - Use case of Manual System. | 12 |
| Figure 2.9 | - isoTracker Interface | 19 |
| Figure 2.10 | - OpenDocMan Interface | 19 |
| Figure 2.11 | - OpenKM interface | 19 |
| Figure 2.12 | - HESK interface | 20 |
| Figure 2.13 | - Phases in the Rational Unified Process | 21 |
| Figure 2.14 | - Phases in the Rational Unified Process Cycle | 23 |
| Figure 3.1 | - High Level Use case diagram of Proposed System | 26 |
| Figure 3.2 | -User Management Use Case | 27 |
| Figure 3.3 | -Asset Management Use Case | 30 |
| Figure 3.4 | - Log Management Use Case | 31 |
| Figure 3.5 | - Incident Management Use Case | 32 |
| Figure 3.6 | - Class Diagram for Proposed System | 33 |
| Figure 3.7 | - Database diagram of the Proposed System | 34 |
| Figure 3.8 | - Login Screen of the Information Security Management System | 35 |
| Figure 3.9 | - Home Screen of the Information Security Management System | 35 |
| Figure 3.10 | - User Administration of the ISMS | 36 |
| Figure 3.11 | - New User Creation | 36 |
| Figure 3.12 | - Delete user confirmation dialog screen | 37 |
| Figure 3.13 | - Incident Reporting Screen | 37 |
| Figure 3.14 | - Incident Management Screen | 38 |

| | |
|---|----|
| Figure 3.15 – Asset Creation Screen | 38 |
| Figure 3.16 – Asset Management | 39 |
| Figure 3.17 – System Monitoring Log Entry Screen | 39 |
| Figure 3.18 – ISMS Document upload screen | 40 |
| Figure 3.19 – Asset Register Report Selection | 40 |
| Figure 3.20 – MVC Architecture | 41 |
| Figure 4.1 – Network diagram for ISMS system implementation | 47 |
| Figure 4.2 - MVC Architecture with PHP | 48 |
| Figure 4.3 – Database connection Parameters code | 49 |
| Figure 4.4 – Header layout and User menu code | 49 |
| Figure 4.5 – Pagination Code Segment | 50 |
| Figure 4.6 -Main login page code segment | 51 |
| Figure 4.7 – Login controller Code segment | 51 |
| Figure 4.8 – Authentication system model code segment | 52 |
| Figure 4.9 – Incident Creation UI Code Segment | 53 |
| Figure 4.10 – Incident Creation controller code segment. | 53 |
| Figure 4.11 – Delete incident controller code segment | 54 |
| Figure 4.12 – Asset Management Code segment | 55 |
| Figure 4.13 – Code Segment of Document Upload | 55 |
| Figure 4.14 – Code Segment of Log Creation View | 56 |
| Figure 4.15 – Incident management Sub Module model code segment | 56 |
| Figure 5.1 – Software Testing Levels | 58 |
| Figure 5.2 – Asset Master Test Data. | 61 |
| Figure 5.3 – Questionnaire Sample | 68 |
| Figure A.1 – Importing lankemhub.sql in to database | 74 |
| Figure A.2 – Database parameter configuration | 75 |
| Figure B.1 - High level use case diagram | 76 |
| Figure B.2 – User Management Use Case | 77 |
| Figure B.3 – Asset Management Use Case | 79 |
| Figure B.4 – Log Management Use Case | 80 |

| | |
|---|-----|
| Figure B.5 – Incident Management Use Case | 81 |
| Figure B.6 – Class Diagram for Proposed System | 82 |
| Figure B.7 - Database Diagram | 82 |
| Figure B.8 – Activity Diagram for User Creation | 83 |
| Figure C.1 – Login Screen | 84 |
| Figure C.2 – Logoff menu | 85 |
| Figure C.3 – Report Incident menu path | 85 |
| Figure C.4 – Incident Creation Screen | 86 |
| Figure C.5 – Incident Ticket Number | 87 |
| Figure C.6 - Menu Incident Manage | 88 |
| Figure C.7 – Manage Incidents | 88 |
| Figure C.8 – View Incidents | 89 |
| Figure C.9 – Change Incident | 90 |
| Figure C.10 – Confirm Delete Incident | 90 |
| Figure C.11 – Review incident | 91 |
| Figure C.12 - Incident Log Book | 92 |
| Figure C.13 – Create IT Asset Menu | 92 |
| Figure C.14 – Create IT Asset Screen | 93 |
| Figure C.15 – Manage IT Assets Menu | 94 |
| Figure C.16 – Mange IT Asset Screen | 95 |
| Figure C.17 – Log management | 96 |
| Figure C.18 – Upload Documents | 97 |
| Figure C.19 – User Manage menu path | 97 |
| Figure C.20 – User Manage screen | 98 |
| Figure D.1 – Logging Screen Testing Result | 100 |
| Figure D.2 – Incident Management Test Evidence | 102 |
| Figure D.3 – Asset Module Test Evidence | 104 |

List of Tables

| | |
|--|-----|
| Table 3.1 – Use Case Diagram for Login Process | 28 |
| Table 3.2 – User Management use case | 29 |
| Table 3.3 – Asset Maintenance Use Case | 30 |
| Table 3.4 – Log Management Use Case | 31 |
| Table 3.5 – Incident Management Use Case | 32 |
| Table 4.1 – Development environment hardware and software requirements | 43 |
| Table 4.2 – Production Environment hardware and Software requirements | 44 |
| Table 5.1 – Test Environment Hardware and Software requirement | 61 |
| Table 5.2 – User management module test Cases. | 62 |
| Table 5.3 – Incident Management Module Test cases | 63 |
| Table 5.4 – Asset Management Sub Module Test cases | 64 |
| Table 5.5 – Log management sub module test cases | 65 |
| Table 5.6 – Document Management Module Tests cases. | 66 |
| Table A.1 – Production Environment hardware and Software requirements | 73 |
| Table B.1 – Use Case Diagram for Login Process | 77 |
| Table B.2 – User Management use case | 78 |
| Table B.3 – Asset Maintenance Use Case | 79 |
| Table B.4 – Log Management Use Case | 80 |
| Table B.5 – Incident Management Use Case | 81 |
| Table D.1 – User management module test Results | 99 |
| Table D.2 – Incident Management Module Test Result | 100 |
| Table D.3 – Asset Management Sub Module Test Result | 102 |
| Table D.4 – Log management sub module test results | 104 |
| Table D.5 – Document Management Module Tests Results. | 105 |

List of Acronyms

| | |
|---------|---|
| PLC | - Public Listed Company |
| ISMS | - Information Security Management System |
| ISO | - International Standard Organization |
| ISSC | - Information Security Steering Committee |
| SLA | - Service Level Agreements |
| CISO | - Chief Information Security Officer |
| DMS | - Document Management System |
| PDF | - Portable Document Format |
| ER | - Entity Relationship |
| RUP | - Relational Unified Process |
| GUI | - Graphical User Interface |
| UML | - Unified Modeling Language |
| CPU | - Central Processing Unit |
| RAM | - Random Access Memory |
| PC | - Personal Computer |
| MVC | - Model View Controller |
| WYSIWYG | - What you see is what you get |
| SQL | - Structured Query Language |
| HTML | - Hyper Text Markup Language |
| VPN | - Virtual Private Network |
| URL | - Universal Resource Locator |
| UI | - User Interface |
| CLI | - Command Line Interface |
| PHP | - Hypertext preprocessor |
| SDLC | - Software Development Life Cycle |
| ISO | - International Standard Organization |
| XML | - Extensible Markup Language |
| HTTP | - Hypertext Transfer Protocol |
| XSLT | - extensible Stylesheet Language |

Chapter 1- Introduction

1.1 Introduction to Lankem Ceylon PLC

Lankem was created by Royal Dutch Shell in 1964. It was initially set up as an agro chemical business. As Sri Lanka started encouraging local ownership, the company acquired local investors and was listed on the Colombo Stock Exchange. The company has been prevalent in most industry segments in the country but primarily focused on chemicals and manufacturing.

Over the past 25 years Lankem has diversified into other related businesses. Lankem's Paints and Bituminous Products have been the frontrunners in new product development in their specific industries. Lankem Consumer Products has been in existence since the late 70's, and has been the pioneer in producing mosquito coils, washing machine detergent powders and liquid detergents in Sri Lanka. Today Lankem Consumer Products are a household name in Sri Lanka [1].

Lankem has over the years invested heavily in Information Technology projects to improve the efficiency of the business process.

1.2 Motivation for the Project

Lankem Ceylon PLC is a large enterprise with several subsidiaries. Lankem Ceylon IT Department adopted ISO/IEC 27001:2013 Information Security Management standard in October 2016. Since the certification is audited and reviewed by Bureau Veritas, UK on regular intervals. Information Security Team should maintain various documents, logs, Information Security Incident reporting and etc.

Currently all the information's are maintained in Hard Printed copies and MS Excel.

1. Documents

- Information Security Policy
- Procedures and Guidelines
- Forms
- SLA/Maintenance Agreements
- ISSC Meeting Minutes

2. Logs Management

- Backup Logs
- Server Utilization Logs
- Background/Batch Jobs Monitoring logs.
- Disaster Recovery Drill logs
- Fire Evacuation Drill logs
- Server room Access logs
- User Access Matrix Review Logs
- Storage Disposal Logs.

3. Information Security Incident Reporting and Risk Mitigation actions.

4. Information Technology Assets Maintenance.

Maintaining and retrieving all these documents, Logs and incidents in Folders, Hard copies and MS Excel has become a nightmare for Chief Information security officer and Information Security Team and whenever Audit or Review happen retrieving the hard copy documents and excels are very difficult. Also, management is more interested in paper less work environment.

To overcome the above-mentioned issues management has decided to go for a computerized system where all the information related to Information Security are maintained for easy retrieval, Centralized documents storage and digitalize information to align with their Information Technology strategy.

1.3 Objectives and Scope of the Project

Objective of this project is to build a Web based Information Security Management system where Information Security Policy, Procedure & Guide lines, IT Related Forms, SLA/Maintenance Agreements documents can be uploaded and maintained.

logs such as Backup logs, Server utilization logs, Disaster recovery drill logs, Server room access logs and etc can be maintained electronically in the system where ever hard evidence is required it will be scanned and upload to the system and also Reviewing and approval process will be also system based where CISO can review the log in the system.

Incident reporting system where users and information security team can log the security related incidents to system, follow-up action can be updated, and solution can be maintained for record purpose. Also, this system will be used for IT helpdesk operation also.

Information security asset management is important part of ISO 27001 requirement. Asset maintenance which include creation, Asset Classification, Asset retirement, Asset Disposal and reports.

Summary of the Scope

- Document Management Module for maintaining Policy, Procedure and other documents.
- Log management module for maintaining logs such as backup logs, Background logs etc.
- Helpdesk Module to manage Security related incidents and will be used as IT Helpdesk.
- Asset Management Module to manage Information Technology assets.
- User Authentication Module

1.4 Structure of the Dissertation

The dissertation consists of six main chapters to provide overall idea about the Information Security Management System for Lankem Ceylon PLC. Below are the remaining chapters after introduction chapter.

Chapter 2 - Analysis

The analysis chapter describes the existing system of the particular domain. Apart from that, this chapter will include functional and nonfunctional requirement gathering, and will use top level use case diagrams to describe the usage of the system when applicable.

Chapter 3 - Design

This Chapter will describe the process involved in the design phase, such as design methodologies of the system, tools and techniques used and the database design. This chapter also contains use case diagrams and narratives. Also poses with graphical user interfaces to describe the key features of the system.

Chapter 4 - Implementation

This implementation chapter defines the development procedures and the methodologies of the system. Further describes the deployment environment as well as development tools and technologies. This chapter also provides the code snippets to show how client requirements have implemented.

Chapter 5 - Evaluation

This chapter shows how to test the system using sample data and the output of the system. Especially this phase includes test cases as a testing procedure to ensure this system is tested in all the user perspectives and ready for future requirements.

Chapter 6 – Conclusion

This is the Final chapter of the dissertation & this includes the re-evaluation of project and describes future requirements, enhancement capabilities and recommendations.

References

All the URLs references and necessary quotations which helped to write this report are contained in this section.

Appendix

Consists of seven topics; each topic describes the system in detail. This section has been written in detail for the interested parties to learn about the system.

Chapter 2 - Analysis

2.1 Introduction

The principal objective of the systems-analysis phase is the specification of what the system needs to do to meet the requirements of end users. It is the top most important phase of the software development life cycle. The definition of the Analysis is

“A systems analyst researches problem, plans solutions, recommends software and systems, at least at the functional level, and coordinates development to meet business or other requirements. Although they may be familiar with a variety of programming languages, operating systems, and computer hardware platforms, they do not normally involve themselves in the actual hardware or software development. Because they often write user requests into technical specifications, the systems analysts are the liaisons between vendors and information technology professionals. They may be responsible for developing cost analysis, design considerations, staff impact amelioration, and implementation time-lines [2]”

2.2 Existing System

Presently Lankem Ceylon using manual system for information security management other than web based for incident management system (Open source HESK Helpdesk).

Policies, Procedures, Forms, SLA/Agreements are in either Printed document or MS Word/PDF documents. Logs are maintained manually in log books. Assets are maintained in MS Excel workbook.

2.2.1 Documents (Policies, Procedures, Forms and SLA's)

Currently all the Information Security policies, procedures, forms, and SLA's are maintained in Printed copies illustrated in figure 2.1 and 2.2 and MS Word documents stored in Shared folder. Updates to the policies and procedures are done on MS word and versions also maintained in the document.

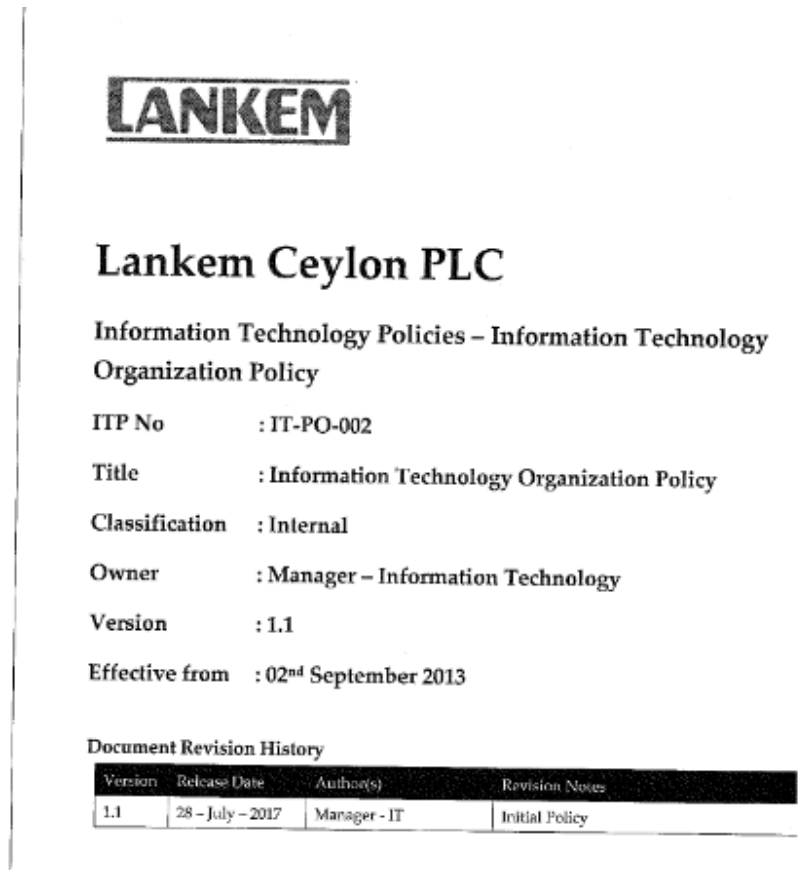


Figure 2.1 – Information Security Policy Document Sample Front Page

Lankem Ceylon PLC
Information Security Management System
Records Control Procedure

Document Number: LKM – IS – PR – 015

Classification: Internal

Owner: Chief Information Security Officer (CISO/GM-IT)

Effective Date: 01/01/2016

Document History:

| Release Date | Revision No | Author(s) | Description | Recommended By | Approved By |
|--------------|-------------|------------|-----------------|---|----------------------------------|
| 01/01/2016 | 1.0 | CISO/GM-IT | Initial Version | Information Security Steering Committee | Director/Chief Financial Officer |
| | | | | | |

Approval by CISO/GM-IT

.....
Signature

.....
Date

Figure 2.2 – Information Security Procedure Sample Front Page

2.2.2 Logs Management

Currently logs are maintained in printed forms illustrated in figure 2.3 and 2.4. where System / Information Security administrators fill manually and later reviewed by IT Manager. Various type of monitoring logs is maintained by Lankem.

| SAP SERVER UTILIZATION | | | | | | |
|------------------------|-------------|------|------------|--------|---------|-----------|
| DATE | SERVER NAME | DISK | FREE SPACE | USED % | REMARKS | SIGNATURE |
| 27/05/17 | Dev | 01 | 200 | 50.63 | 50% | - |
| | | 02 | 100 | 50.63 | 50% | - |
| | | 03 | 100 | 50.63 | 50% | - |
| | | 04 | 100 | 20.29 | 80% | - |
| 01/06/17 | Dev | 01 | 140 | 50.71 | 63% | - |
| | | 02 | 140 | 50.71 | 63% | - |
| | | 03 | 140 | 50.71 | 63% | - |
| | | 04 | 140 | 50.71 | 63% | - |
| 01/06/17 | PRD | 01 | 200 | 216.18 | 28% | - |
| | | 02 | 200 | 200.26 | 25% | - |
| | | 03 | 200 | 230.29 | 23% | - |
| | | 04 | 200 | 115.86 | 61% | - |
| 17/06/17 | Dev | 01 | 100 | 50.63 | 50% | - |
| | | 02 | 100 | 50.63 | 50% | - |
| | | 03 | 100 | 50.63 | 50% | - |
| | | 04 | 100 | 20.28 | 80% | - |
| 01/06/17 | Dev | 01 | 140 | 50.71 | 63% | - |
| | | 02 | 140 | 50.71 | 63% | - |
| | | 03 | 140 | 50.71 | 63% | - |
| | | 04 | 140 | 50.71 | 63% | - |

Figure 2.3 - Server Utilization Monitoring log form

ONLINE BACKUP

SERVER : ECC PRODUCTION

BACKUP SHOULD BE TAKEN: DAILY

| YEAR | MONTH | DAY | DATE | STATUS | REMARKS | SIGNATURE |
|------|-----------|-----------|--------|--------|------------------|-----------|
| 2017 | JULY | MONDAY | 3-Jul | ok | - | - |
| | | TUESDAY | 4-Jul | ok | - | - |
| | | WEDNESDAY | 5-Jul | ok | - | - |
| | | THURSDAY | 6-Jul | ok | - | - |
| | | FRIDAY | 7-Jul | ok | - | - |
| | | MONDAY | 10-Jul | ok | - | - |
| | | TUESDAY | 11-Jul | ok | - | - |
| | | WEDNESDAY | 12-Jul | - | not schedul | - |
| | | THURSDAY | 13-Jul | ok | - | - |
| | | FRIDAY | 14-Jul | ok | - | - |
| | | MONDAY | 17-Jul | Error | Error in back up | - |
| | | TUESDAY | 18-Jul | ok | - | - |
| | | WEDNESDAY | 19-Jul | ok | - | - |
| | | THURSDAY | 20-Jul | ok | - | - |
| | FRIDAY | 21-Jul | ok | - | - | |
| | MONDAY | 24-Jul | ok | - | - | |
| | TUESDAY | 25-Jul | ok | - | - | |
| | WEDNESDAY | 26-Jul | ok | - | - | |
| | THURSDAY | 27-Jul | ok | - | - | |
| | FRIDAY | 28-Jul | ok | - | - | |
| | AUGUST | TUESDAY | 1-Aug | ok | - | - |
| | | WEDNESDAY | 2-Aug | ok | - | - |
| | | THURSDAY | 3-Aug | ok | - | - |
| | | FRIDAY | 4-Aug | ok | - | - |
| | | MONDAY | 7-Aug | not ok | not schedul | - |
| | | TUESDAY | 8-Aug | ok | - | - |
| | | WEDNESDAY | 9-Aug | ok | - | - |
| | | THURSDAY | 10-Aug | ok | - | - |

Figure 2.4 – Backup Log Form

2.2.3 Incident Management

As part of ISO27001 requirement security related incidents such as Virus Attack, Spamming, Hacking, Physical security breaches etc. should be reported and logs of follow-up actions should be maintained. Currently there is a Form (illustrated in figure 2.5) for reporting the incidents and follow up actions are maintained by System Admin/Information Security Admin in MS Excel (illustrated in figure 2.6). Even though they have implemented an IT Helpdesk but still not fully operational.

| LANKEM INCIDENT REPORTING FORM (LKM-IS-FM-010) | |
|--|------------------------------------|
| Incident ID: | |
| Name of Incident Reporter (IR): | Email of IR: Telephone of IR: |
| Reported by: | Incident Reporting time: |
| Incident Date: | Business Unit/ Process affected: |
| Incident Occurrence time: | Physical location of the Incident: |
| Duration of the Incident (if | |
| Incident description: | |
| Initial classification of the incident: | |
| Incident Type : | |
| Incident Recorded by (on behalf of IR): | |
| Sign off: | |
| Date: | |

Figure 2.5 Incident Reporting Form

| Incident ID | Incident Reporter(IR) | Incident Description | Incident Type | Impact Level | Assigned person | Action Taken | Escalated to Vendor | Root Cause |
|-------------|-----------------------|----------------------|---------------|--------------|-----------------|--------------|---------------------|------------|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

Figure 2.6 Incident Follow Up Activity Log MS Excel

2.2.4 Information Technology Asset Management

As part of ISO27001 requirement Information Technology assets should be maintained properly. Currently assets are maintained in MS excel as shown in figure 2.7. Asset acquisition, Retirement, Transfers and Disposal details are maintained in the same MS Excel workbook.

| Asset No. | Asset | Asset Type | Asset Description | Asset Owner | Asset Custodian | Confidentiality Rating (1-3) | Integrity Rating (1-3) | Availability Rating (1-3) |
|-----------|--|-------------|---|-------------|--------------------------------|------------------------------|------------------------|---------------------------|
| | Laptop use by GM IT | Laptop | To store critical documents related to IT department | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| | PC use by GM IT | Workstation | For Day to day operations | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| | PC use by Manager IT | Workstation | For Day to day operations | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| | Laptop use by manager IT | Workstation | To store critical documents related to IT department | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| | PCs use by IT staff (11) | Workstation | For day today IT operations | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| LCCPU0151 | LDAP win server | Server | LDAP active directory server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| LCCPU0362 | Anti Virus Server-HP Pro | Server | Anti- Virus Server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| LCCPU0145 | Call Billing server-Dell | Server | Call Billing server – PABX | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| LCCPU0192 | AS400 - I Series | Server | AS400 Application server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| LCCPU0416 | IBM Blade Server | Server | 3 blades (BI Server / HR System server / Solmon Server) | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| LCCPU0319 | LDAP server-HP Pro | Server | LDAP Server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |
| LCCPU0318 | LDAP Backup server-HP ProLiant MC110G6 | Server | Backup LDAP Server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | |

Figure 2.7 – Information Technology Asset’s List

2.3 Use Case Diagram of The Existing System

A Detailed study was carried out to understand the functionalities of the existing manual system. Since there is no computerized system available only higher-level use case diagram was created. Refer the design chapter for the proposed detail use case diagrams. Figure 2.8 describe the existing system in use-case diagram.

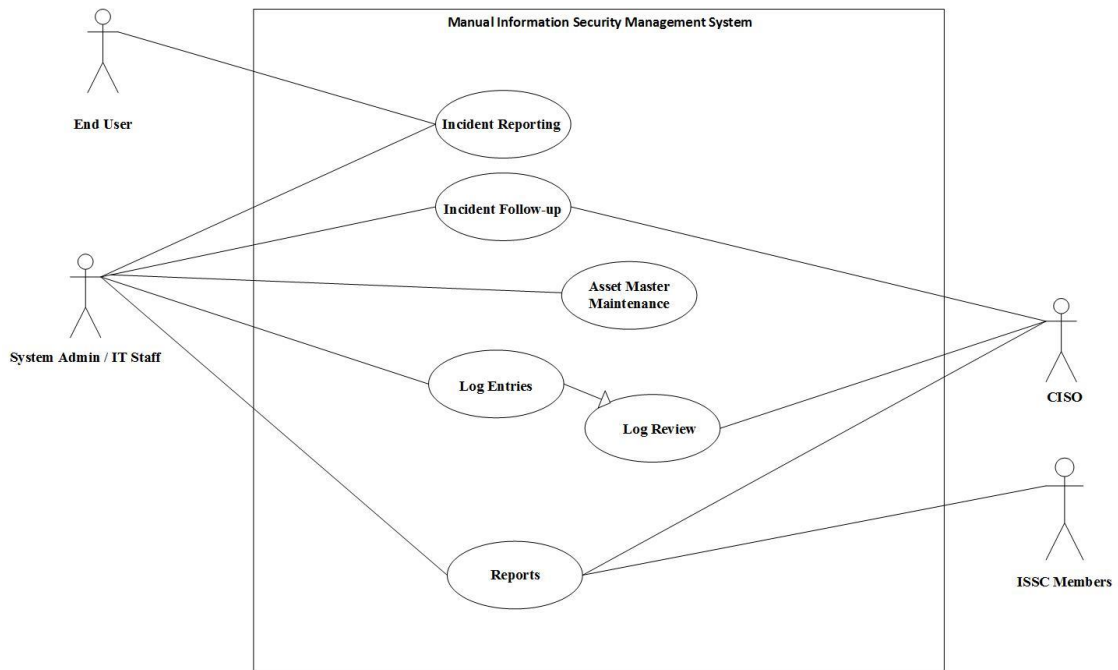


Figure 2.8 – Use case of Manual System.

2.4 Drawbacks of Existing System

Below are the key drawbacks identified from the existing manual Information Security Management system. Drawbacks of the systems were identified after carefully going through the existing system process and interview with CISO, System Administrator, IT Manager, Information Security Steering committee members and Users.

- Policy documents, Procedures, Request forms and SLA's are kept in Printed documents and MS Word/PDF. Whenever use somebody wanted to view the policy or any other document System administrators send printed copy or PDF email this process causing lots of delay and wastage of paper. Whenever there is a version update to the document its emailed to each relevant user in some cases its printed and sent.
- Documents are stored in Shared Folder in a Network attached storage but files are not properly arranged for easy access and retrieval. Printed documents are stored in

box files retrieval of document takes very long time and consuming a very long time.

- Logs are maintained in Papers and stored in boxed files very difficult to view logs and supporting evidence also printed and kept. b'cos of that huge amount of paper wastage and going through the evidence also very difficult for Auditors and Information Security Officer.
- Incident Management reporting – currently there is a system implemented but it doesn't have the ISO27001 incident reporting format and, it's a free web based system very difficult to customize to suite the lankem requirement.
- Asset Management – Assets acquisition, Transfers, retirement and disposals are maintained in Excel which is very difficult and also the changes to the entries are not properly logged.
- None of the process are integrated with each other's so sometimes System administrators must duplicate the same data in several places. Example when asset is transferred asset master should be updated same time transfer log is manually updated.

2.5 Feasibility Study

Feasibility study is a process of finding strengths, weaknesses, opportunities and threads. And analyze how above four facts effects to proposed system. Regarding the system mainly considering about following types,

2.5.1 Technical Feasibility

The proposed system analyzed to check any technical difficulties or Barriers found. If any technical barrier found, then find the alternative solution for the problem as per the advices of the supervisor.

Technical requirements such as Server availability, Capacity availability and other software requirements were checked at client side.

2.5.2 Legal/Ethical Feasibility

Mainly considered about the legal issues of the use of the software after developing the system on windows servers. (Need licensed server operating system)

2.6 Requirements Gathering

Requirement gathering is the process of collecting functional and nonfunctional requirements from the company in order to design the new information system.

2.6.1 Requirement Gathering Techniques

Requirements identification is the most important part in system analysis phase. If the client's requirements are not gathered and defined accurately the whole project will be meaningless, since gathered requirements will not reflect what the client actually wants. It is a well-known fact that most projects fail because of the lack of understanding of the system requirements. To avoid failures in the future, it is recommended to use a well-structured requirement gathering techniques based on the environment [3]. Following are the methods which used to gather requirements in the proposed system.

- Face to Face Meetings and Interviews were held with CISO, System Administrators, IT Manager, ISSC Members and others IT Division Staff.
- Inspected the manual documents, Excel files, ISO27001 guidebook.
- Distributed a questionnaire to ISSC members to gather the feedback on their functional expectation in the new system.

- Captured facts were represented as scenarios and Use-Case diagrams to give a clear picture of the proposed system to CISO and IT Manager to get the confirmation on the requirements.

2.6.2 Functional Requirements

Definition of functional requirements - “In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish [4]. The key functional requirements of the Information Security Management System as follows.

- System should provide the functionality to Upload/Delete/Edit Policy documents, Procedures, ISSC Meeting minutes and SLA’s by CISO, Viewing and Downloading Facility to ISSC Members, IT Staff, System Administrators. Version Management facility should be available for CISO.
- CISO should be able to upload the request forms to the system. End users should be able to view and download the forms and also End users should be able to view and download acceptable usage policy.
- Uploaded SLA’s if relevant to an asset it should be linked to asset. When viewing the asset IT staff and System administrators should be able to view the available SLA’s.
- System administrators and IT Staff should be able to Enter, View, update daily monitoring logs in the system and, they should be able to attach supporting evidence documents to the system. IT manager should be able to view and review the entries.

- IT staff and System administrators should be able to Add, Update, Delete and View assets in the system and, they should be able to transfer the asset to different custodian, User, Location etc. System should track the history of an asset changes and transfers.
- IT Staff, System Administrators, End Users should be able report/View Security related incidents to system. Follow-up action and solution should be updated by CISO in the system.
- Email notification should be sent to the CISO when a new incident is reported in the system and when solution is updated in the system it should notify the reporter.
- System should have four authorization level. 1st level IT Staff, 2nd Level System Administrators, 3rd level ISSC Members and 4th level CISO / IT Manager.

2.6.3 Non-Functional Requirements

Non-functional requirements describe the requirements from non-technical stakeholders and which are not directly produced with the specific functions of the system. In order to complete the success of the system should meet the set of nonfunctional requirements.

Definition - “In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture [5]”. The key nonfunctional requirements are below.

Accuracy and Consistency

Important task to keep the high accuracy of the processed data and produce the correct results and the functions of the system should be consistent all the time.

Security and reliability

Important task to keep all the records securely, especially employment data, employee's personal with high security. This system should have timely backup system to back up all the data including employee profiles. System should be able to install without problems.

Usability

Usability of the system should be high level and it should be user friendly environment to all the users.

Reusability and Maintainability

This system should facilitate future enhancements and expandability and should develop with clear reusable source code.

2.7 Similar Type Systems

By studying software systems available in the market today, more experiences could be obtained about how the developing system should be and how the required functionalities should be presented. Following are a few similar systems that were reviewed to design the proposed system.

2.7.1 isoTracker

isoTracker ISO Document Control software (interface shown in figure 2.9) handles the process of document control in a logical and easy to use fashion. isoTracker builds up version information as each new version of a document is imported into the document

control module. It assigns approvers and permits your draft documents to be reviewed and electronically approved before publication [6]. System was developed for ISO 9001 and ISO 14001 some of the functionalities of the system can be used for ISO 27001 document management.

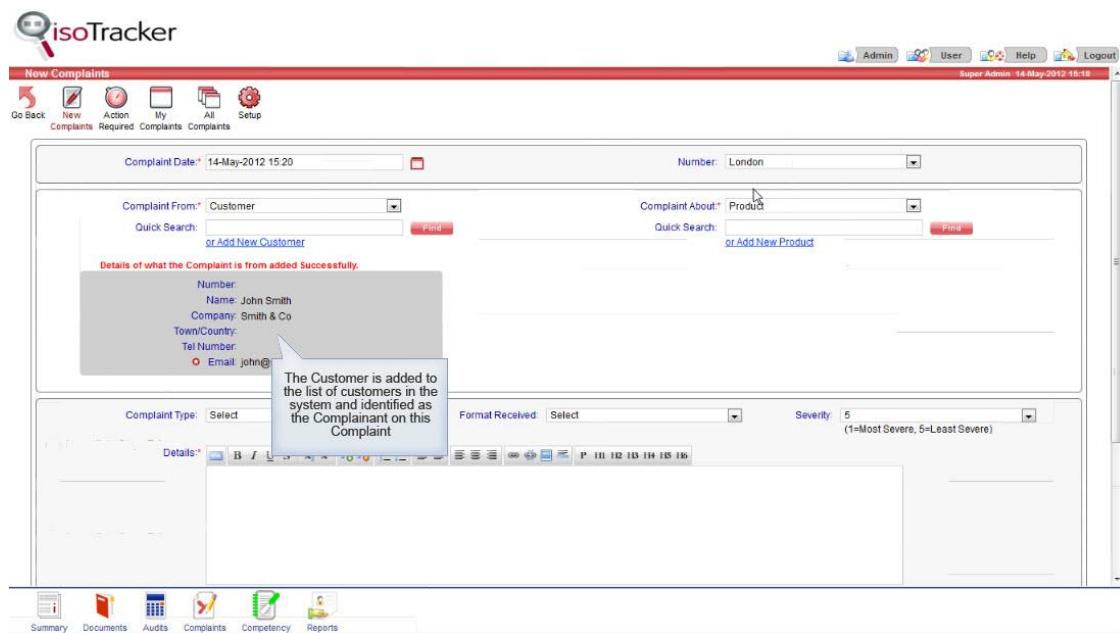
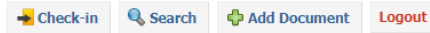


Figure 2.9 - isoTracker Interface

2.7.2 OpenDocMan DMS

OpenDocMan was designed to be used for a variety of situations and solve many problems. Besides the common features of document management that OpenDocMan handles, here is a list of some other things you could do with OpenDocMan [7]. Even though OpenDocMan is a document management system it can be used to store and retrieve the documents, but it doesn't have the capabilities to Manage Logs or Incidents. Figure 2.10 illustrate the interface of the System.



Demo resets once per hour

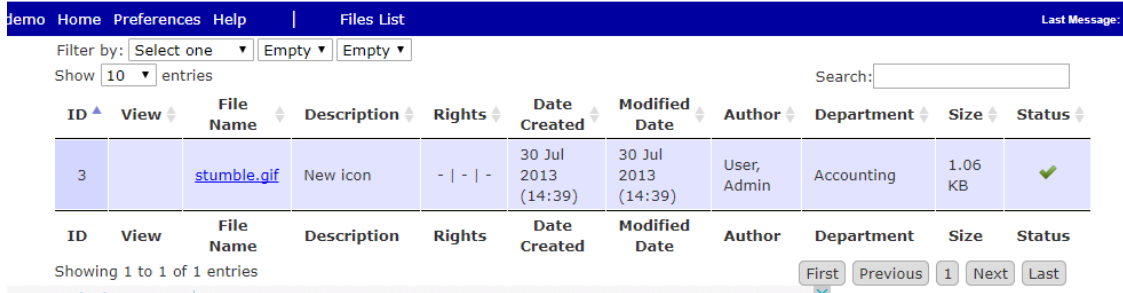


Figure 2.10 - OpenDocMan Interface

2.7.3 OpenKM

OpenKM is a document management software (Figure 2.11) that integrates all essential document management, collaboration and an advanced search functionality into one easy to use solution. The system also includes administration tools to define the roles of various users, access control, user quota, level of document security, detailed logs of activity and automation setup [8].

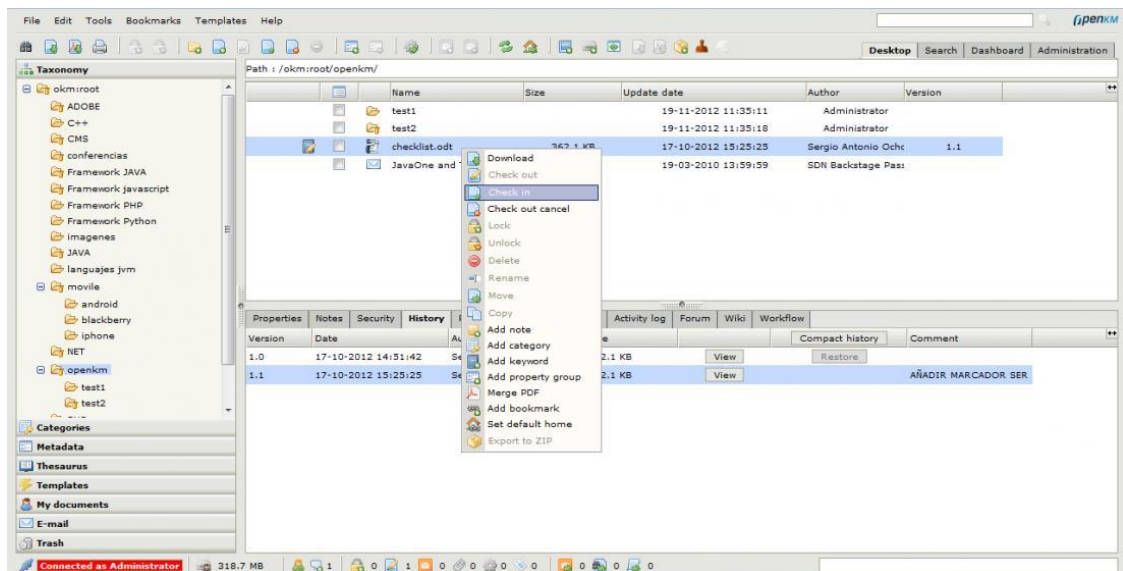


Figure 2.11 - OpenKM interface

2.7.4 HESK

HESK is an open source web based helpdesk and knowledge management system which is currently used by Lankem Ceylon PLC to manage their IT Helpdesk. Built using PHP and MySQL. Figure 2.12 illustrate the admin interface of the HESK

| <input type="checkbox"/> | Tracking ID | Updated | Name | Subject | Status | Last replier | <input type="checkbox"/> |
|--------------------------|------------------------------|-----------|--------------------------|---|-------------|--------------------------|--------------------------|
| <input type="checkbox"/> | RXD-1HH-JLUQ | 19 Jun 17 | Lakshitha | * Emailing SAP Reports | New | Lakshitha | <input type="checkbox"/> |
| <input type="checkbox"/> | LEQ-8XD-GXWL | 19 Jul 17 | Umayangani | * Computer Mouse not working porperly | New | Umayangani | <input type="checkbox"/> |
| <input type="checkbox"/> | AGZ-UQZ-AB7R | 10:46 | Mr.Seevali Rathnathilake | * Format lap top | New | Mr.Seevali Rathnathilake | <input type="checkbox"/> |
| <input type="checkbox"/> | GSU-9EN-L7LW | 10:46 | hashini | * Replace / change a mouse | New | hashini | <input type="checkbox"/> |
| <input type="checkbox"/> | X8L-3U7-Q9RD | 09 Aug 17 | Nilukshi | * No Access to Finance All folder | In Progress | Nilukshi | <input type="checkbox"/> |

Tagged Ticket
 * Assigned to me
 * Assigned to other staff

Set priority to: Low

>> Show tickets
 Status:
 New
 Resolved
 Waiting reply
 In Progress
 Replied
 On Hold

| [More options](#)

Figure 2.12 - HESK interface

2.8 Overview of Process Models

The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Implementation and Maintenance [10].

Evolutionary development this approach interleaves the activities of specification, development and validation. An initial system is rapidly developed from abstract specifications. This is then refined with customer input to produce a system that satisfies the customers' needs.

Component-based software engineering this approach is based on the existence of a significant number of reusable components. The system development process focuses on integrating these components into a system rather than developing them from scratch. (Ian Sommerville, 2007).

These are the more widely used process model in the software development industry today.

2.9 Process Model for the Proposed System

Iterative and Evolutionary process model was chosen to develop this system. This is where small portions of software are developed to uncover important issues early, before problems or faulty assumptions can lead to disaster. Although there are many process models, RUP (Rational Unified Process) model was chosen over other models considering the advantages it has over the other models.

The Rational Unified Process (RUP) is an iterative software development process (illustrated in figure 2.13) framework created by the Rational Software Corporation, a division of IBM since 2003. RUP is not a single concrete prescriptive process, but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs. RUP is a specific implementation of the Unified Process. It consists of four major phases [11].

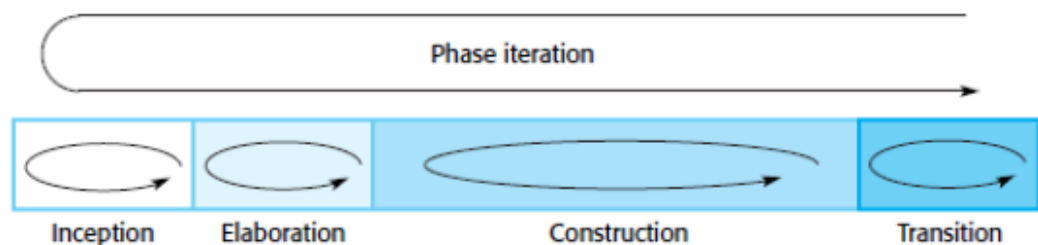


Figure 2.13 - Phases in the Rational Unified Process

1. Inception

The goal of the inception phase is to establish a business case for the system. You should identify all external entities (people and systems) that will interact with the system and

define these interactions. You then use this information to assess the contribution that the system makes to the business. If this contribution is minor, then the project may be cancelled after this phase.

2. Elaboration

The goals of the elaboration phase are to develop an understanding of the problem domain, establish an architectural framework for the system, develop the project plan and identify key project risks. On completion of this phase, you should have a requirements model for the system (UML use cases are specified), an architectural description and a development plan for the software.

3. Construction

The construction phase is essentially concerned with system design, programming and testing. Parts of the system are developed in parallel and integrated during this phase. On completion of this phase, you should have a working software system and associated documentation that is ready for delivery to users.

4. Transition

The final phase of the RUP is concerned with moving the system from the development community to the user community and making it work in a real environment. This is something that is ignored in most software process models but is, in fact, an expensive and sometimes problematic activity. On completion of this phase, you should have a documented software system that is working correctly in its operational environment.

[*Sommerville 2007*]

The following figure 2.14 shows the phases and iterations of RUP life cycle.

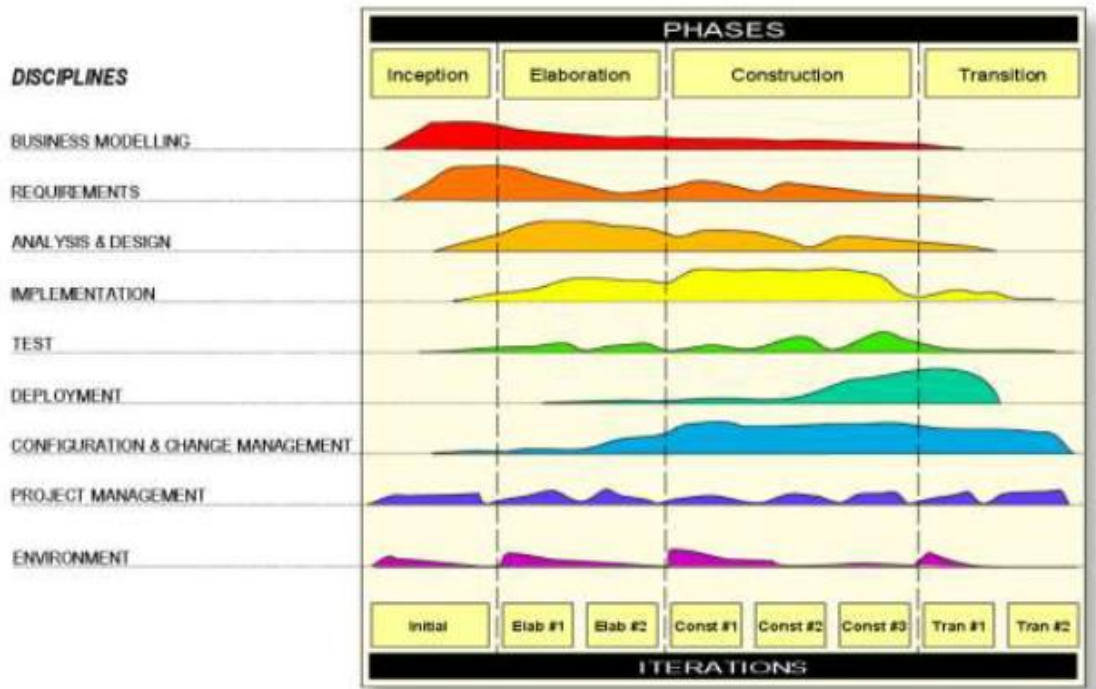


Figure 2.14 - Phases in the Rational Unified Process Cycle

Chapter 3 - Design

3.1 Introduction

Software design is an iterative process through which requirements are translated into a “blueprint” for constructing the software. Software design may refer to either "all the activities involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems [9]".

This chapter discusses about the few process models that are being used in the current software industry and the selected process model to develop the system. The object-oriented design techniques (UML Diagrams) were used throughout the design process. The database design process consists with ER diagram and some main user interfaces in the system.

3.2 Alternative Solutions to the System

Standalone System

Alternate solution is to develop a standalone application which can fulfill company requirements. But it is a very expensive solution. Also, the deployment, updating, maintenance processes are time consuming, as every client must be maintained individually. The standalone systems are limited to a physical location so have usability limitations.

Open source System

As an alternative, we can use freely available open source ISO Document Management System (Mostly available for ISO 9001 and 14001 only) in the internet and use it. Even though downloaded open source software may full fill most of the requirements. But when

it's comes to customization to suite our requirement it's very difficult to read and understand the code for modification. Also, it's very difficult to maintain and fix bugs on the open source systems.

Example Lankem Ceylon plc uses HESK Open Source helpdesk system but they couldn't customize to suite the incident reporting requirements since code is very complex.

3.3 Justification of Selected Solution

The system has been proposed to implement it as a web based solution for the following reasons.

- Web Based system there is no client installation necessary and maintenance is easy, but standalone GUI based system is very difficult to install and maintain.
- System would be a platform independence, since some of the employees are using different devices with different platforms (Mac PC, Windows, Linux, Android and etc)
- Client doesn't want to spend much on licensing, so open source approach is more suitable.
- Easy access to system from any location from a compatible web browser.
- Upgrading or adding new functionalities to the system is easy compare to standalone GUI based application.

3.4 Design Techniques

There are some designing techniques like Prototyping, Structured design, Rapid Application Development. Object Oriented design technique was selected out of them for the main design concepts like abstraction, inheritance, modularity, reusability, encapsulation and message passing. Unified Modeling Language 2.0v plays an important role in Object Orientation. UML allows us to build easy to use and easy to understand diagrams for programmers.

- Use-Case diagrams – shows what the system needs to do.

- Sequence diagrams – shows how the objects interact overtime.
- Activity diagrams – shows object states at a specific timeline.
- Class diagrams – shows the main objects and relationships between them.

PHP is the programming language which was used to develop the system, although it is not fully functional Object-oriented language, it does have extensive object-oriented features. Using this, the system can be structured to promote the reuse of program code, which is an important feature of Object orientation

[Somerville, 2007]

3.5 Use-Case Diagram for the Proposed System

Highlevel use-case diagram of the proposed system illustrated below in figure 3.1

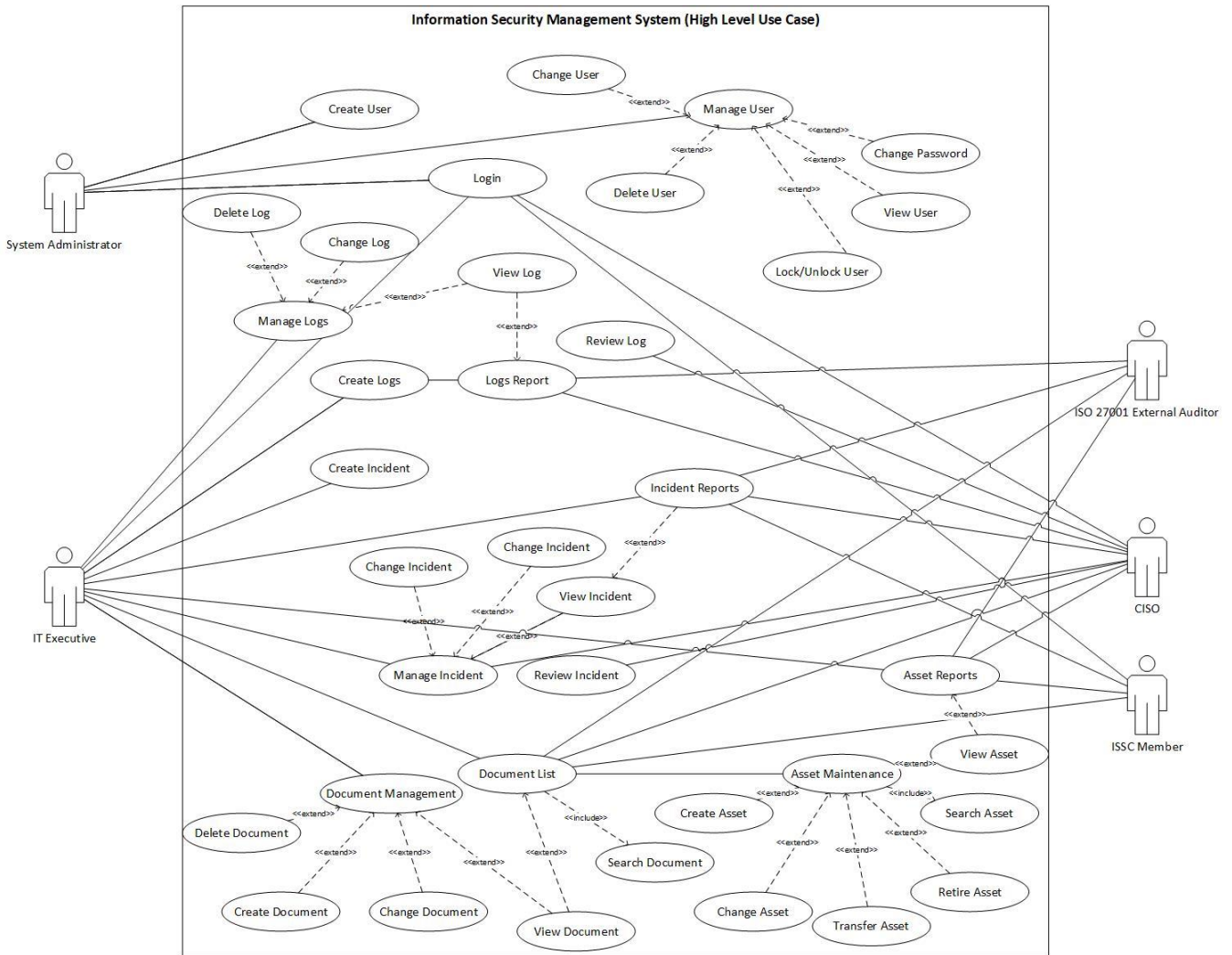


Figure 3.1 - High level use case diagram

User Management and Authentication Sub module

User management and Authentication submodule will handle the Creation, Change, Lock, Password reset, Login Process and authorization functionalities.

Figure 3.2 , Table 3.1 and Table 3.2 illustrate the use-case of the user management and authentication sub module.

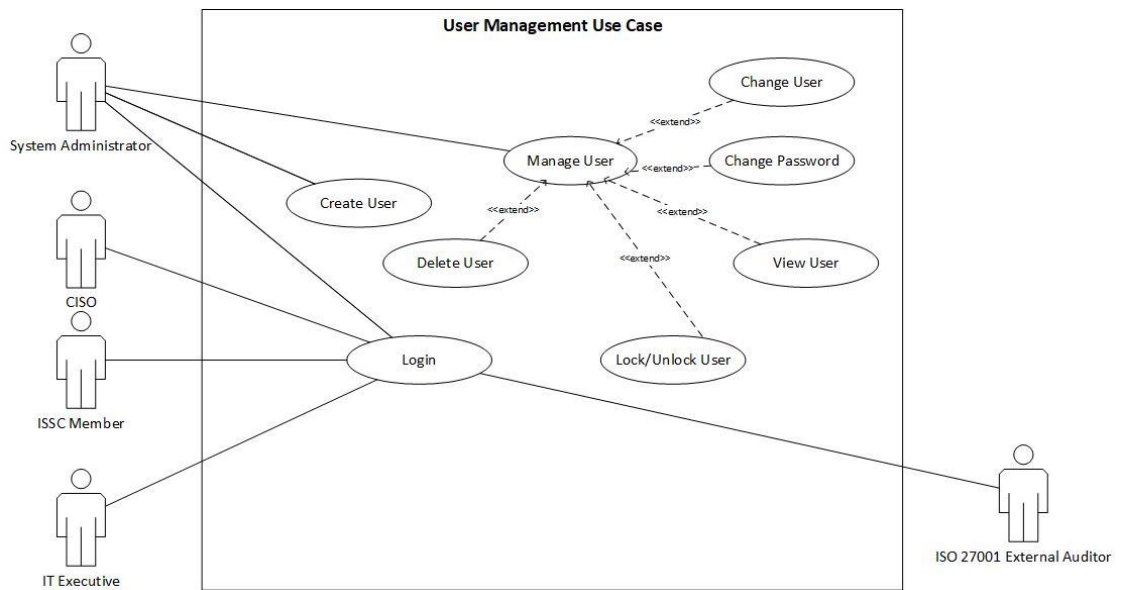


Figure 3.2 – User Management Use Case

| | |
|--|--|
| Use-Case | Login to the System |
| Actors | System Admin, IT Executive, CISO, Auditors and ISSC Members. |
| Overview | |
| Authorized users log in to the system | |
| Preconditions | |
| <ul style="list-style-type: none"> • User should have an account in the system. • Account should be in unlocked state, | |
| Flow of Events | |
| <ul style="list-style-type: none"> • The user enters user name and password. • If entry is invalid, system will throw an error message. • If the username and password is valid the system redirects the user to his/her appropriated home page where access is restricted according to the account privileges. | |
| Post Conditions | |
| Authorized users will be logged in the system and will be re directed to the home page. Where they will be able to access the menu items for which they are authorized. | |

Table 3.1 – Use Case Diagram for Login Process

| | |
|---|----------------------|
| Use-Case | User Management |
| Actors | System Administrator |
| Overview | |
| Create and Manage users to access the system. | |
| Preconditions | |
| <ul style="list-style-type: none"> • Person wanted to have access to the Information security management should forward their request. • Request must be approved by the Chief information security officer. | |
| Flow of Events | |
| <ul style="list-style-type: none"> • User request for the system access. • Request will be forwarded to CISO Approval. • If the request is approved by the CISO, System admin will create the User account and notify the user. • In case if user wanted to change to his user account, he or she should send the request to CISO, once approved System Admin will change the necessary user management data in the system. | |
| Post Conditions | |
| Only authorized personal can gain access to the system and restricted to allowed functionalities. | |

Table 3.2 – User Management use case diagram

Asset Management Sub Module Use-Case Diagram

Figure 3.3 and Table 3.3 illustrate the use-case of the Asset management sub module

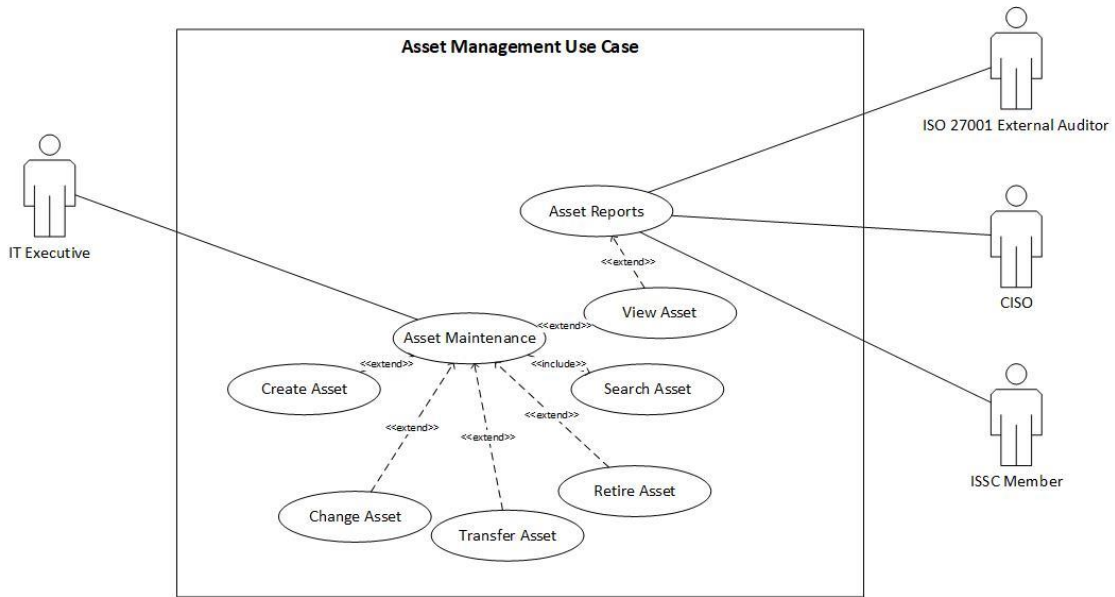


Figure 3.3 – Asset Management Use Case

| | |
|------------------------|---|
| Use-Case | Asset Management |
| Actors | IT Executive |
| Overview | Create and Manage Information Technology Assets in the system |
| Preconditions | <ul style="list-style-type: none"> Asset should be physically available in the system. Barcoded number of the asset should be available before creating the asset in the system. |
| Flow of Events | <ul style="list-style-type: none"> Once the Asset physically available and manual barcoded number is assigned Asset information will be fed in to the system. Any mistakes and changes to the entered asset can be changed. Transfer from one location to another, one cost center to another and change of owner can be done using the transfer option. When the asset reaches the end of life asset retirement can be performed. |
| Post Conditions | Effectively manage the IT asset Inventory |

Table 3.3 – Asset Maintenance Use Case

Log Management Sub Module Use-Case Diagram

Figure 3.4 and Table 3.4 illustrate the use case of the log management sub module.

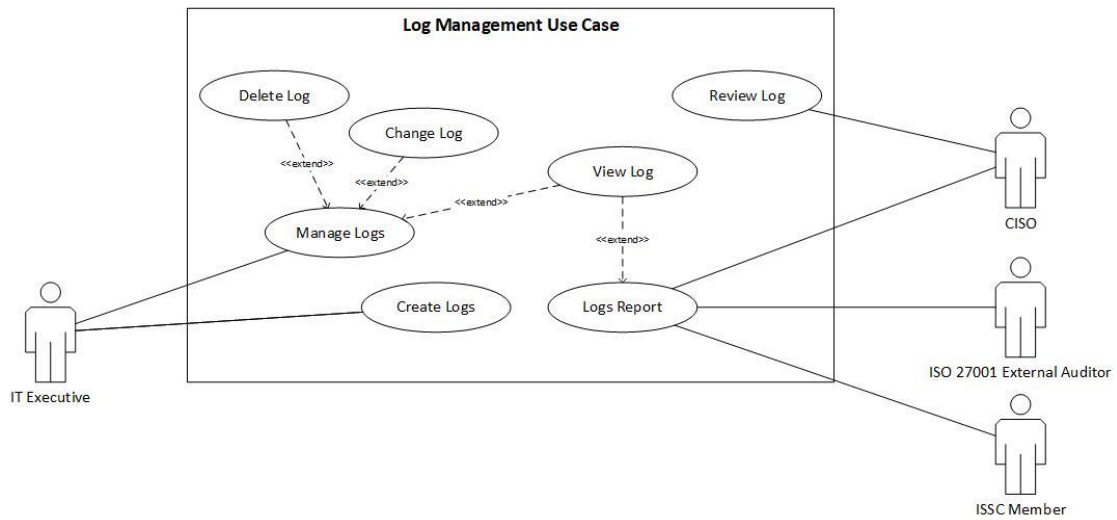


Figure 3.4 – Log Management Use Case

| | |
|------------------------|--|
| Use-Case | Log Management |
| Actors | IT Executive |
| Overview | Create and Manage System Monitoring and Job Monitoring Log. |
| Preconditions | <ul style="list-style-type: none"> Monitoring system should be checked and gather the required information from the systems before creating the log entry in the ISMS system. |
| Flow of Events | <ul style="list-style-type: none"> Once the IT executive complete the Daily monitoring activates and gather required information such as CPU Utilization, Job start time, Job End time and etc. Gathered information’s will be entered to the system. If any mistakes in the entered data using the change option data can be changed. Only changes possible is before the review |
| Post Conditions | Effectively manage the daily monitoring logs and status. |

Table 3.4 – Log Management Use Case

Incident Management Sub Module Use-Case Diagram

Figure 3.5 and Table 3.5 illustrate the use case of the incident management sub module.

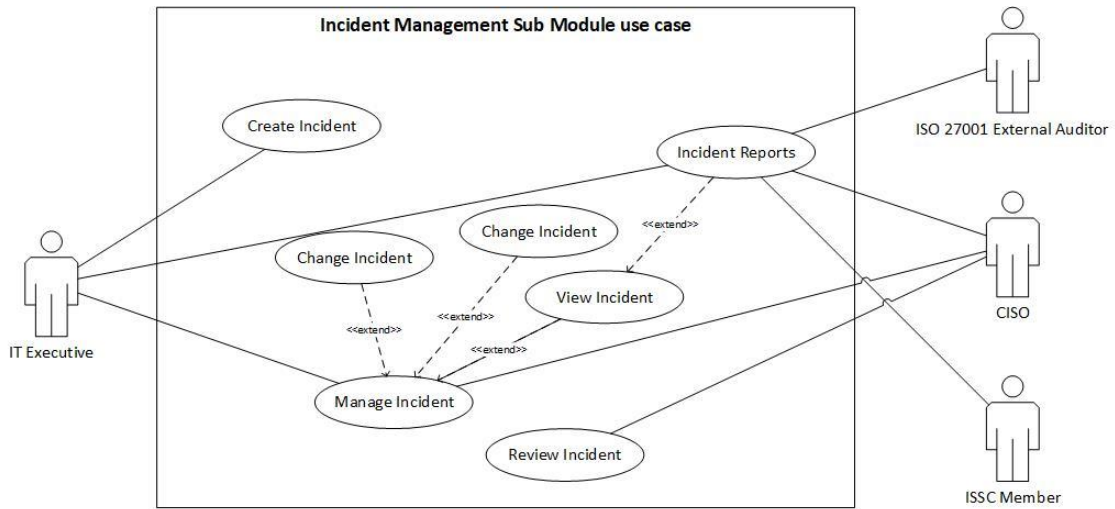


Figure 3.5 – Incident Management Use Case

| | |
|---|---------------------|
| Use-Case | Incident Management |
| Actors | IT Executive, CISO |
| Overview | |
| Create and Manage Information Security Related incidents. | |
| Preconditions | |
| <ul style="list-style-type: none"> • Employees should inform any information security related incidents to IT Executive. | |
| Flow of Events | |
| <ul style="list-style-type: none"> • Once the IT executive receive a complaint from the employees He/She will create the incident in the system. • Any supporting documents can be uploaded. • Changes to the already created incident can be made using the change. • CISO will review the incident and update the action taken to mitigate the re occurrence. | |
| Post Conditions | |
| Effectively managing the incidents for future analysis when improving the information security. | |

Table 3.5 – Incident Management Use Case

3.6 Class diagram for the Proposed System

Figure 3.6 illustrate the class diagram of the proposed system.

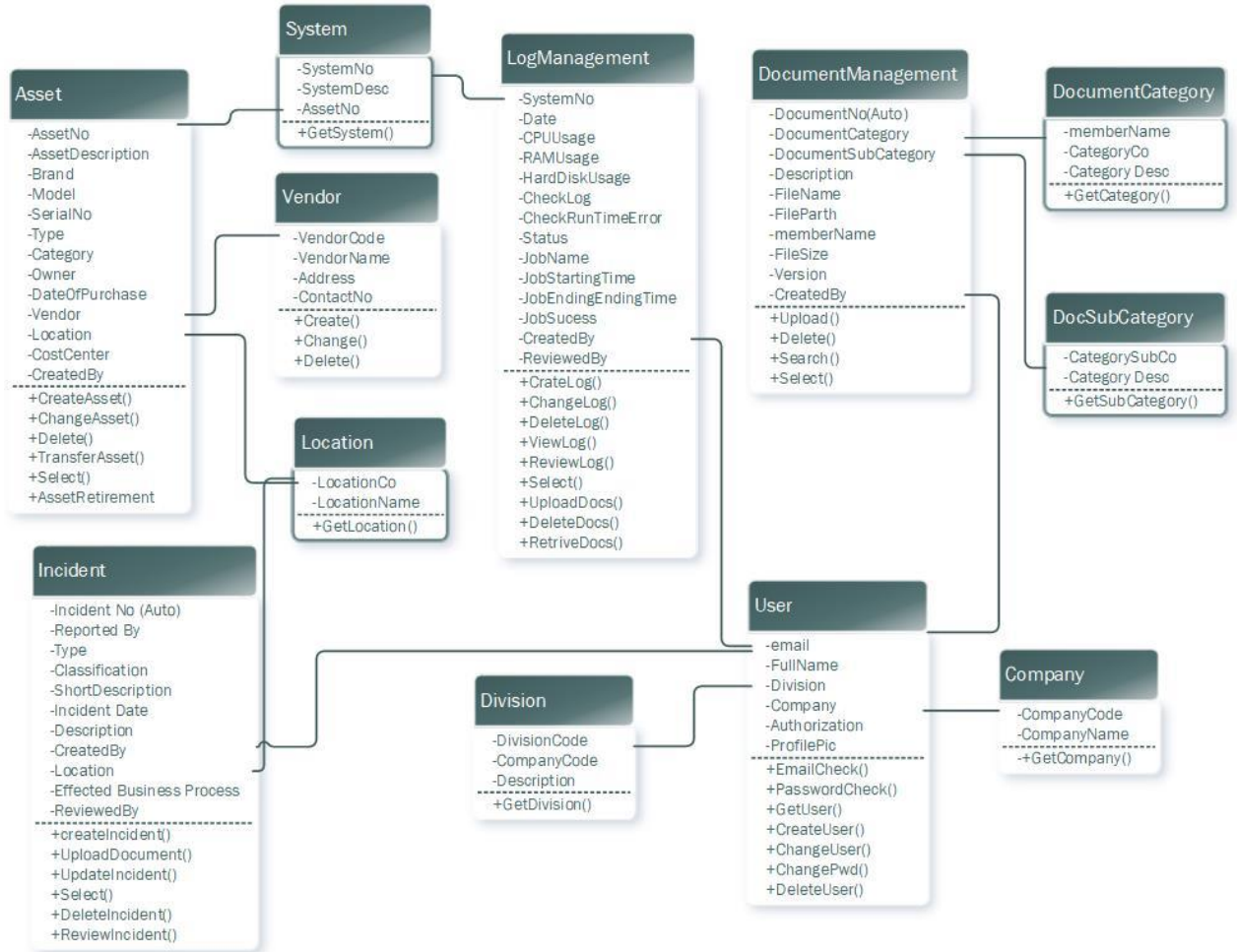


Figure 3.6 – Class Diagram for Proposed System

3.7 Database Model for the Proposed System

A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized, and manipulated so relational model is selected for database modeling. The relational model for database management is a database model based on first-order predicate logic. All data is represented in terms of tuples, grouped into relations. A database organized in terms of the relational model is a relational database. Figure 3.7 illustrate the database diagram of the proposed information security management system.

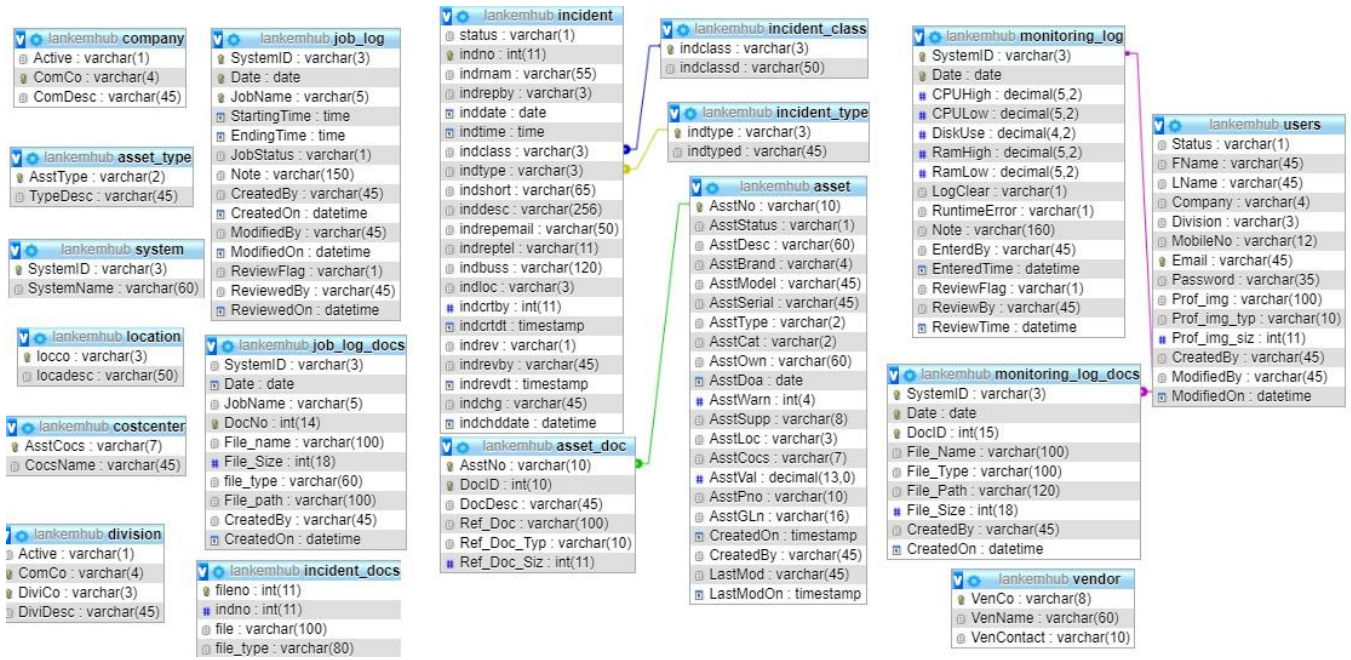


Figure 3.7 – Database diagram of the Proposed System

3.8 User Interface Design

The user interface design was carried out to create a user friendly clean and clear operating environment for the users to work with the system efficiently with the easy to use UI Screens. Entire system UI is developed using Bootstrap 3. Bootstrap is the most popular HTML, CSS and JavaScript framework for developing responsive mobile-first web sites. To reduce the Screen refresh and smooth seamless interaction JQUERY is used for validations. JQUERY is a Java Script library designed to simply the Client-Side scripting.

Some main user interface design rules were followed while designing the system. As shown below.

- Select most suitable color combination for user convenience and to feel comfortable.
- Responsive and Mobile-first.
- Provide clear and consistent navigation along with easy access.
- Choose a font and the text-size which is clear and readable.
- Display meaningful error messages and proper instructions when the user encounters any errors.

Login Screen

Following figure 3.8 shows the Login screen.

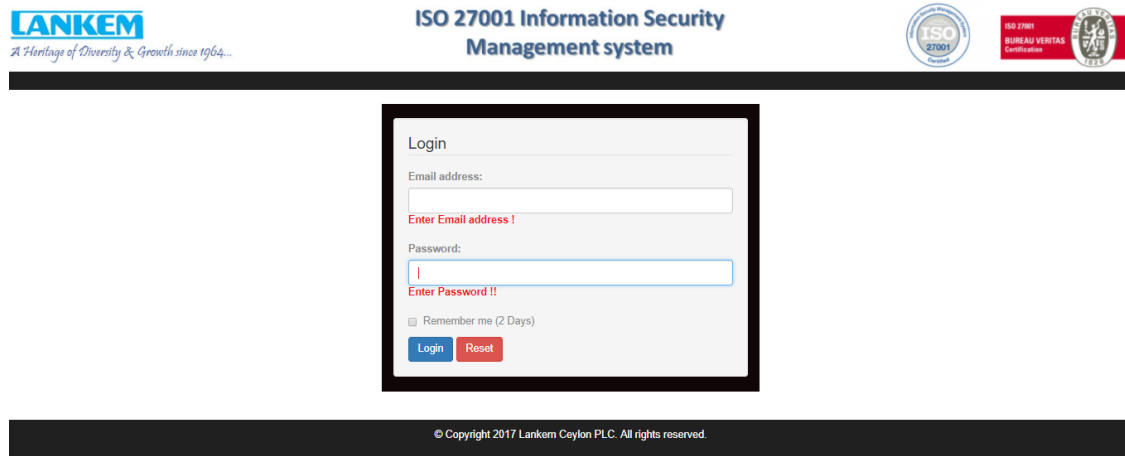


Figure 3.8 – Login Screen of the Information Security Management System

Home Page

Following figure 3.9 shows the Home page of the system (After User login successfully in to the System)



Figure 3.9 – Home Screen of the Information Security Management System

User Management

Following figure 3.10 shows the User Management Screen where System Administrator Can Create (figure 3.11 illustrate Create User interface), Change, Lock/Unlock, Search and Delete Users(figure 3.12). One Screen include all the user management options for easy access without going through menu path.

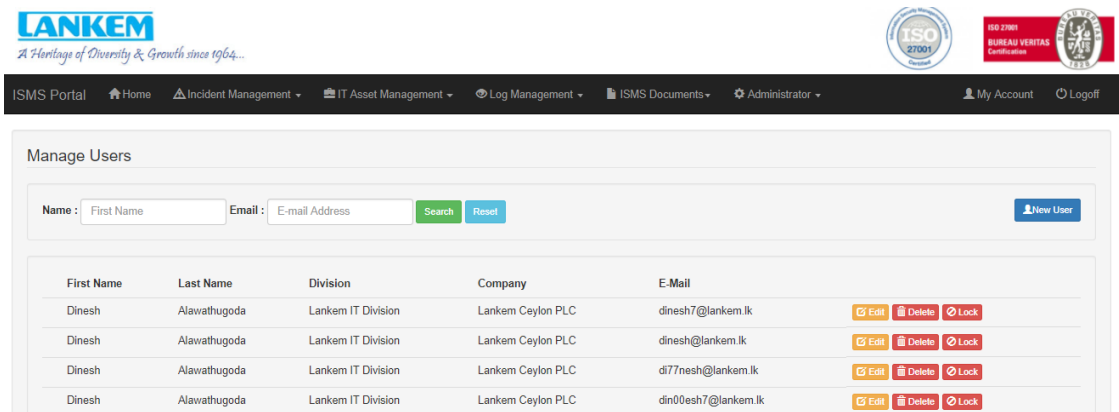


Figure 3.10 – User Administration of the Information Security Management System

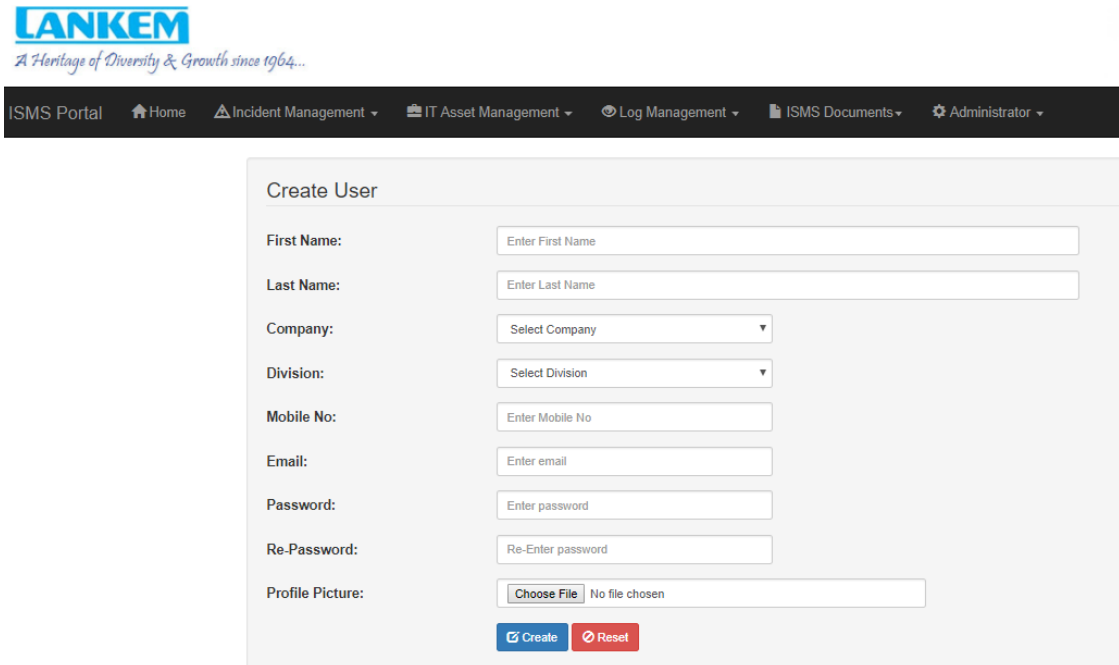


Figure 3.11 – New User Creation

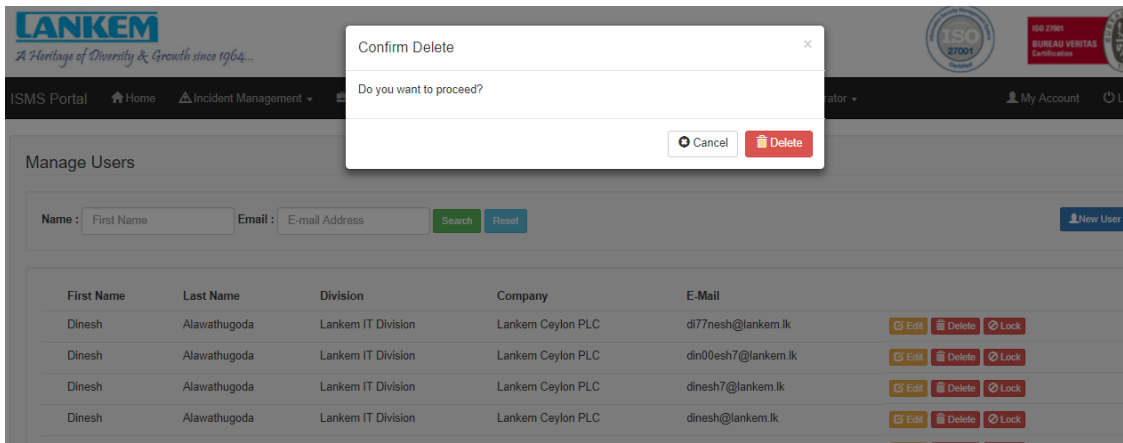


Figure 3.12 - Delete user confirmation dialog screen

Incident Creation

Following figure 3.13 shows the incident reporting screen.

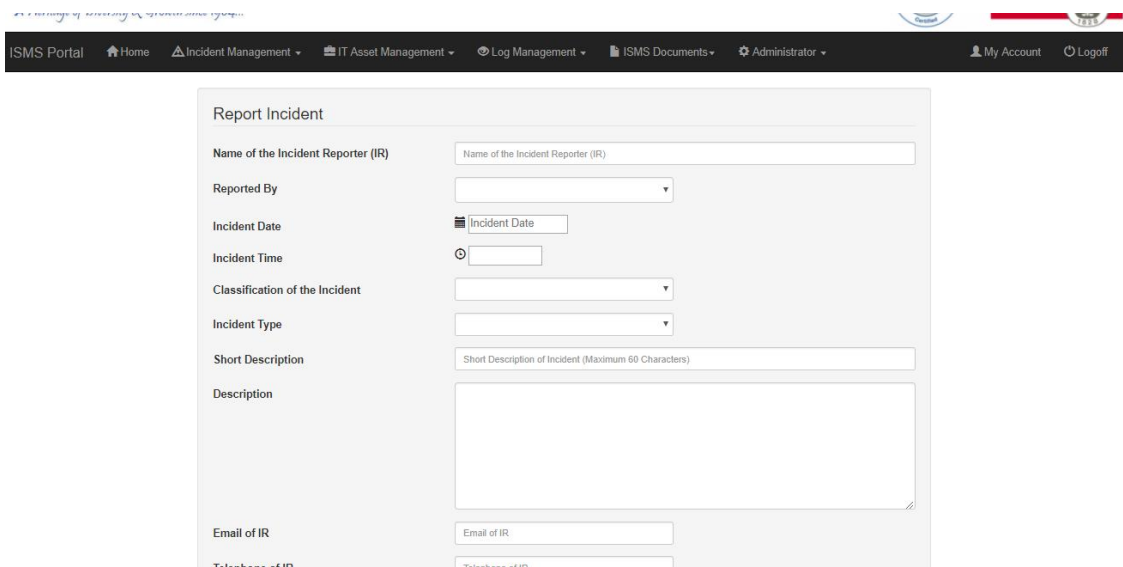


Figure 3.13 – Incident Reporting Screen

Incident Management

Following figure 3.14 shows the incident management screen easy access screen. This screen enables the user to change, Delete, Review, Search and View the Incidents.

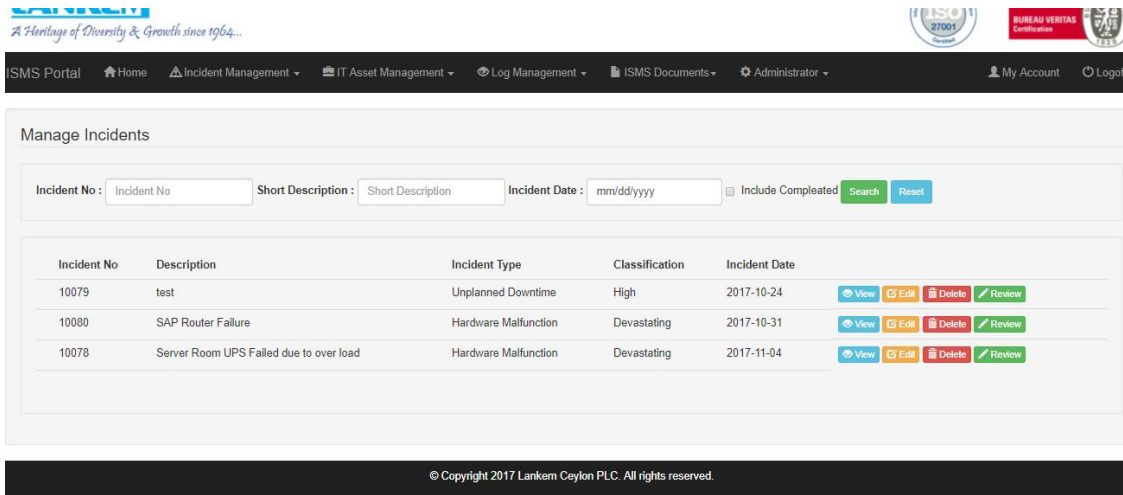


Figure 3.14 – Incident Management Screen

Asset Creation

Following figure 3.15 shows the asset creation form.

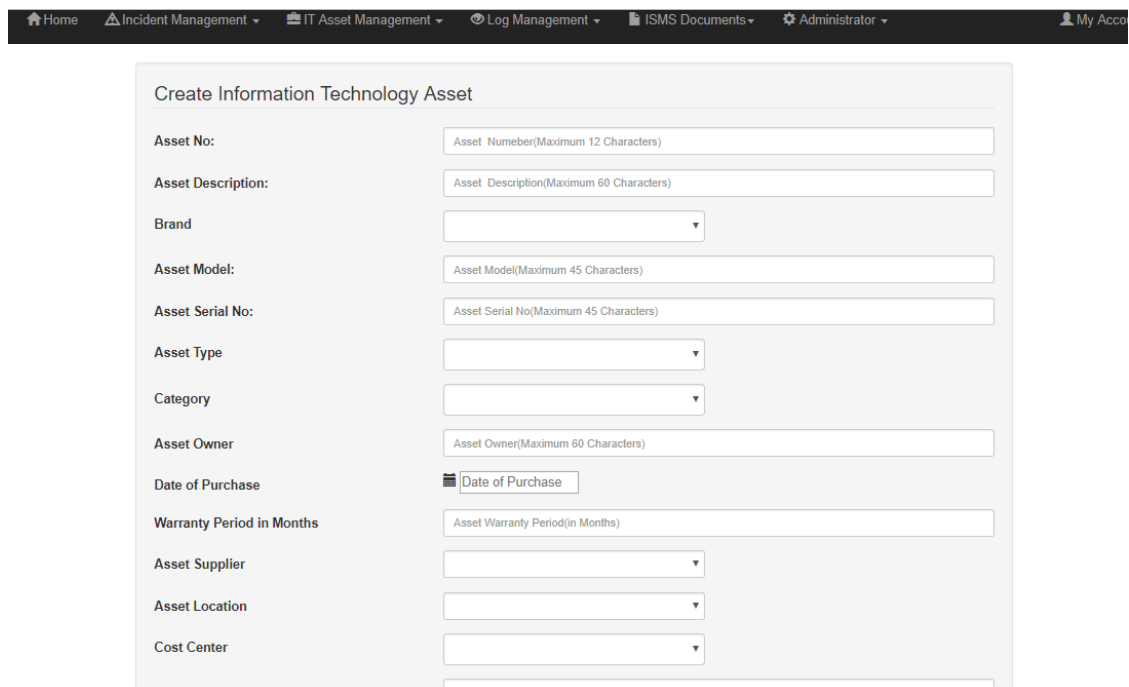


Figure 3.15 – Asset Creation Screen

Asset Management

Following figure 3.16 shows the asset management which includes Changing, Deleting, Asset Retiring, Asset Transfer, Asset search and View Functionalities.

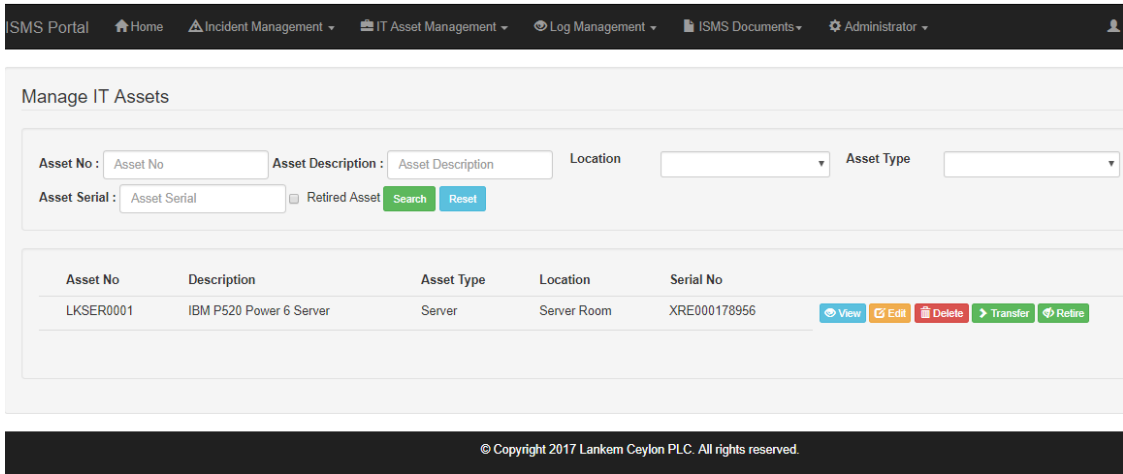


Figure 3.16 – Asset Management

System Monitoring Log Entry Screen

Following figure 3.17 shows the log creation.

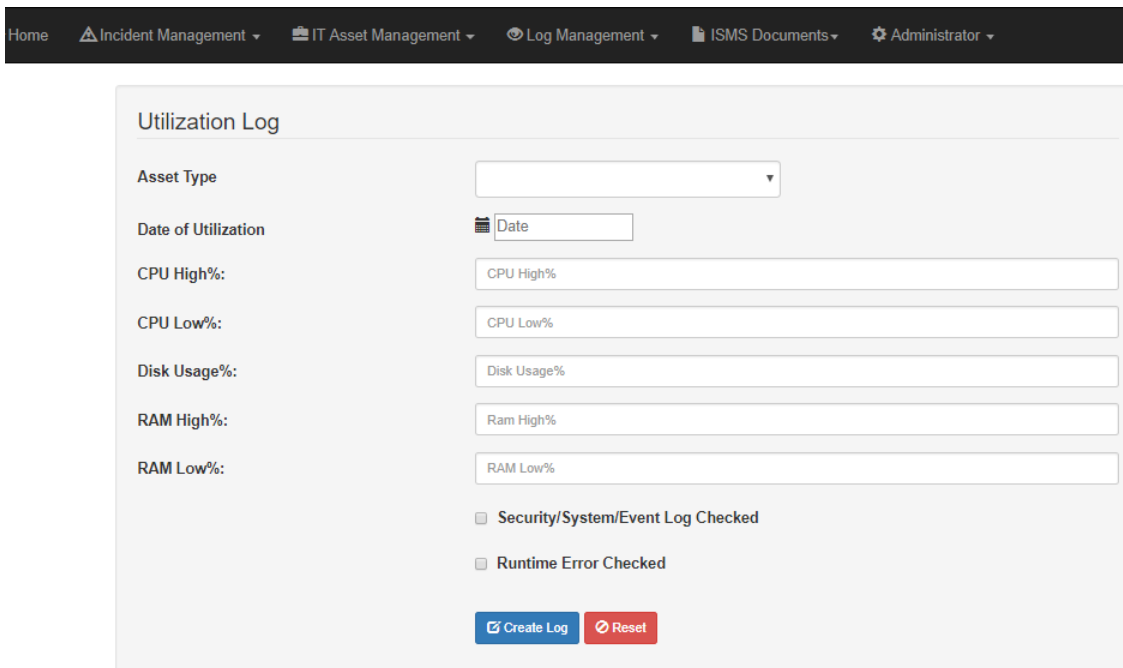


Figure 3.17 – System Monitoring Log Entry Screen

ISMS Document upload screen

Figure 3.18 is the upload screen of the document management system.

Home Incident Management IT Asset Management Log Management ISMS Documents Administrator

Upload Document

Document Category Policy

Document Sub Category Acceptable sage Policy

Description of the Document Document Description

Document Date 2017-10-31

Special Note

Document Choose File No file chosen

Create Reset

Figure 3.18 – ISMS Document upload screen

Asset Register Report Selection

Following figure 3.18 is the asset register report selection screen.

iversity & Growth since 1964... ISO 27001 Certified

Home Incident Management IT Asset Management Log Management ISMS Documents Administrator

Asset Register Selection

Asset Type

Category

Brand

Date of Purchase From Date To Date

Asset Supplier

Asset Location

Cost Center

Execute Reset

© Copyright 2017 Lankem Ceylon PLC. All rights reserved.

Figure 3.19 – Asset Register Report Selection

3.9 Proposed Design Architecture

Model–view–controller (MVC) is a software architecture pattern which separates the representation of information from the user's interaction with it. The model consists of application data, business rules, logic, and functions. A view can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as a bar chart for management and a tabular view for accountants. The controller mediates input, converting it to commands for the model or view the central ideas behind MVC are code reusability and separation of concerns [12]

MVC is an Object Oriented (OO) design pattern. A Model-View-Controller uses class to organize Business Logic (where data is stored, who is authorized to manipulate it, and how the data is manipulated) contained in “Models”, Presentation Logic (how the data from the Models should be rendered) in “Views” and has an overall flow for the application within a “Controller”

Below figure shows the interaction of the MVC components.

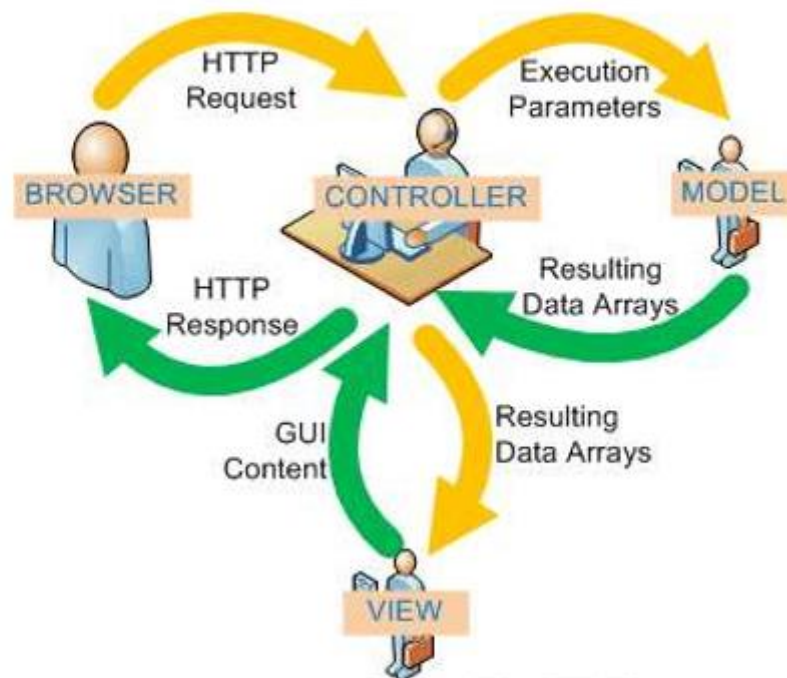


Figure 3.20 – MVC Architecture

The MVC pattern addresses three problems in architecting a solution:

- Maintaining your data in some kind of persistent storage
- Maintaining the logic controlling the flow of the application, what to display to the user, and what actions the user is allowed to perform on the data in your application.
- Presenting information to users of your application

The Model

Models are the portion of the MVC application that implements the “Business Logic”. Business logic is any PHP logic relating to how information is stored in your application. For models that interact with a database table, a single table should ideally be associated with only one model class.

The View

Views are the portion of the MVC application that presents output to the user. The most common output for PHP web applications would be HTML, but views are not restricted to this. Your MVC application might output XML, WML, plain text, images, email or some other content. PHP has a variety of capabilities to use in views, ranging from just being designed to send data back to the browser, to implementing template solutions, to XSLT. Views generally create instances of models, and use methods of these models to get data to present to the user. Views contain all of the PHP code required to transform the raw data from your models into the format you want to present in your application.

The Controller

The controller is the heart of the MVC application. This component is the object that should be aware of the user’s HTTP request (alternatively, the controller might be aware of the command line arguments or last input in a CLI). From this information, the controller should determine what view to display, or what other application related action should take place. Controllers are generally the core component of any MVC project you may adopt (primarily because models and views must be customized to your application by their very nature). Various projects have different strategies for determining how the views and models interact. How these objects interact is referred to as system coupling. If the nature of these relationships is defined using a configuration file (often XML based), this is said to be “loose coupling” [13]. This is the opposite of hard coding application flow into the system

Chapter 4 – Implementation

4.1. Introduction

The implementation stage of software development is the process of converting a system specification into an executable system. It always involves processes of software design and programming. [Sommerville 2007] A suitable programming language and appropriate were selected in the process of implementation and coding. The codes were arranged in understandable format with comments will ensure the possibility for future modifications and maintainability.

4.2. Implementation Environment

There are many considerations when finalizing the implementation environment. Main consideration was to use free and open source development software and components to minimize the cost and eliminate the requirement of licensing. Also, Client strictly adhere to copyright compliance no unlicensed software can't be used. The Following table 4.1 and 4.2 explain the Hardware and Software components and tools were used in the both development and production implementation environments.

| Development Environment | |
|---|--|
| Hardware | Software |
| <ul style="list-style-type: none"> • Inter Core i5 2.30 GHz Processor • 4 GB RAM • 500 GB Hard Disk • Laptop / Desktop PC | <ul style="list-style-type: none"> • Microsoft Windows 8 or Above • XAMPP 3.2.2 or Above <ul style="list-style-type: none"> ○ PHP 7.0.21 or Above ○ MariaDB 10.1.25 ○ Apache 2.4.26 ○ phpMyAdmin 4.7.0 • Web Browser (Chrome) • Code Editor (Adobe Dreamweaver) |

Table 4.1 – Development environment hardware and software requirements

| Production Environment | |
|---|--|
| Hardware | Software |
| <ul style="list-style-type: none"> • Inter Corei5 2.30 GHz Processor • 4 GB RAM • 500 GB Hard Disk • Entry Level Server | <ul style="list-style-type: none"> • Microsoft Windows Server 2016 R2 • XAMPP 3.2.2 or Above <ul style="list-style-type: none"> ○ PHP 7.0.21 or Above ○ MariaDB 10.1.25 ○ Apache 2.4.26 ○ phpMyAdmin 4.7.0 • Backup Software |

Table 4.2 – Production Environment hardware and Software requirements

System was developed on XAMPP 3.2.2 on Microsoft Windows 10 operating system and tested on XAMPP 3.2.2 on Windows 2016 R2 operating system. Even though system is not tested in Linux environment system will work on XAMPP 3.2.2 or Above on any Linux variants.

The System was tested under the following resolutions

- 1024 x 600 (pixels) - Netbook and Tablet devices
- 1024 x 768 (Pixels) - 4:3 Monitors
- 1280 x 768 (pixels) - Widescreen Monitors
- 1366 x 768 (pixels) - Widescreen Laptops
- 5.5 Inches Mobile Screen
- 7” Inches Tablet Screen

Also developed system use Bootstrap for the UI design its responsive and mobile-first. So, system is compatible with any standard screen sizes (5.5 Inches, 7 Inches, 8 Inches, 10.1 Inches etc.)

Development Tools

- Adobe Dreamweaver CC 2017 for coding.
- phpMyAdmin 4.7.0 for Database creation and Modeling.
- Adobe Photoshop CC 2017 for Image editing.

Development Technologies

- PHP 7 for Server-Side Scripting
- Bootstrap was used to build the UI's.
- JQUERY 2.1.3 framework and JQUERY validation plug-in was used for client side and server-side validation.
- MariaDB MySQLi was used for Database development and Query.

4.3. Reused Modules/Components

- JQUER Validation Plug-in from <https://jqueryvalidation.org/> is used for client- side Validation and server-side remote validations.
- JQUERY Date and Time picker was used in the forms to pick date and time.

Other than the above two components no other components were used as its. Some of other components were Open-source modules but customized to suite the development.

4.4. Background of the used Software's

Adobe Dreamweaver

Adobe Dreamweaver is a web design and development application that provides a visual WYSIWYG editor (colloquially referred to as the Design view) and a code editor with standard features such as syntax highlighting, code completion, and code collapsing as well as more sophisticated features such as real-time syntax checking and code introspection for generating code hints to assist the user in writing code. The Design view facilitates rapid

layout design and code generation as it allows users to quickly create and manipulate the layout of HTML elements. Dreamweaver features an

integrated browser for previewing developed web pages in the program's own preview pane in addition to allowing content to be open in locally installed web browsers. It provides transfer and synchronization features, the ability to find and replace lines of text or code by search terms or regular expressions across the entire site, and a templating feature that allows single-source update of shared code and layout across entire sites without server-side includes or scripting. The behaviors panel also enables use of basic JavaScript without any coding knowledge, and integration with Adobe's Spry Ajax framework offers easy access to dynamically-generated content and interfaces. [14]

phpMyAdmin

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement [15].

Adobe Photoshop

Adobe Photoshop is a raster graphics editor developed and published by Adobe Systems. Available for both Windows and Mac, Adobe Photoshop is an extremely powerful application that's used by many professional photographers and designers. You can use Photoshop for almost any kind of **image editing**, such as touching up photos, creating high-quality graphics, and much, much more [16]

4.5. Network Implementation

Figure 4.1 illustrate the network implementation diagram of the developed system.

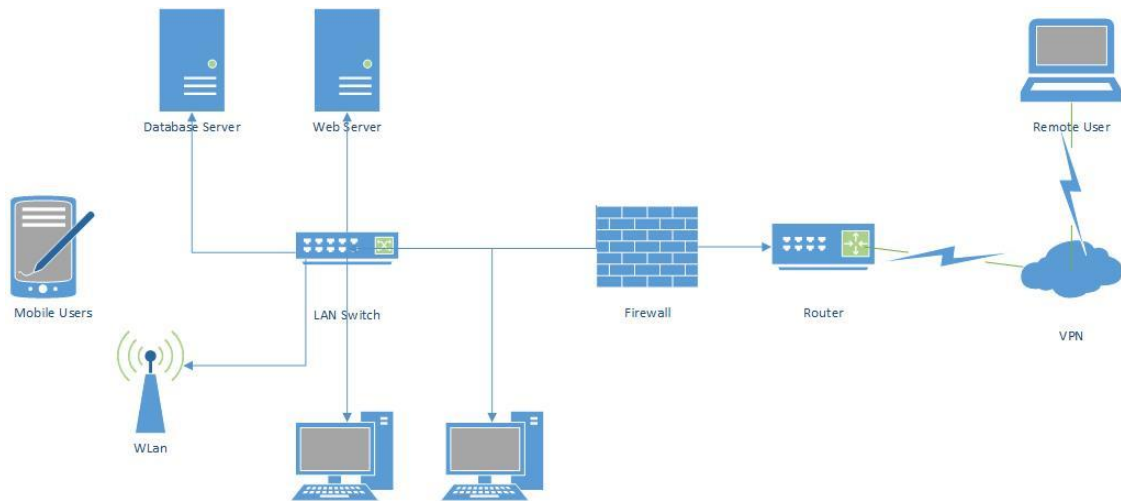


Figure 4.1 – Network diagram for ISMS system implementation

Since this is a web-based system, Lankem IT Team could be installed in a web server which employs any operating system (Windows or Linux). Recommended to use dedicated web server and a database server for better performance. However, the same results can be achieved from a single server if the network traffic to the system and transactions are low in volume. System can be accessed from outside using Remote VPN connections.

4.6. Use of Design Patterns

Even though MVC design patterns were used in development and coding of the system. **It's not fully fledged MVC development** since no frameworks were used in this development.

Model-View- Controller

MVC is a fundamental design pattern often used by applications that need the ability to maintain multiple views of the same data. The MVC pattern hinges on a clear separation of objects into one of three categories – Models for maintaining data, Views for displaying all or a portion of the data and Controllers for handling events that affect the model or views.

Interaction of MVC Architecture with PHP is shown by the following figure 4.2.

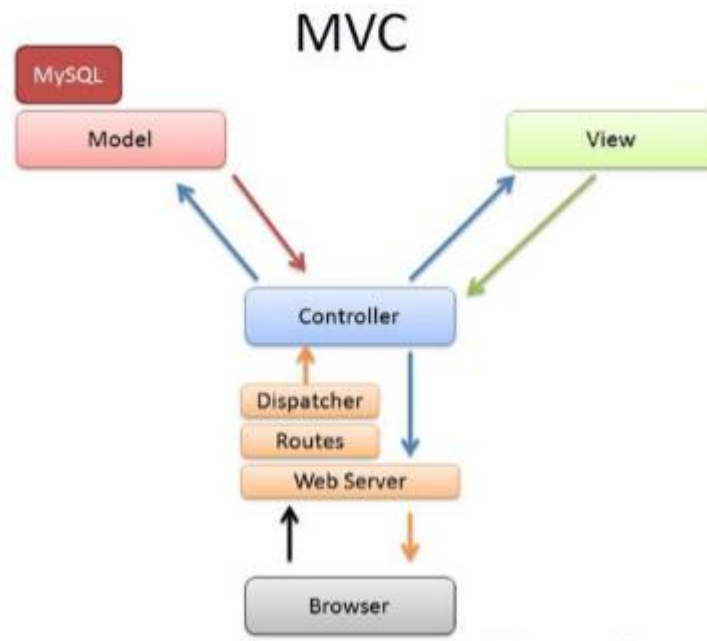


Figure 4.2 MVC Architecture with PHP

4.7. Major Code Sections

The System code is divided to three major segments as it is implemented in MVC design pattern.

As described in previous sections,

- Model: Business logic & processing
- View: User interface (UI)
- Controller: Navigation & input

Only important part of the code section is explained here

Code Segment for Database Configuration

include/db_config.php (figure 4.3)

This php file hold the Database Server, User, Password and Database information required for the system to initiate a database connection.

```

1 <?php
2  define('DB_SERVER', 'localhost');
3  define('DB_USERNAME', 'root');
4  define('DB_PASSWORD', '');
5  define('DB_DATABASE', 'lankemhub');
6  ?>

```

Figure 4.3 – Database connection Parameters code

Important Code Segment for Overall Layout

include/header.php (View)

This php file contain the header layout and User Menu. Figure 4.4 is the screen short of the code section.

```

35
36 <nav class="navbar navbar-inverse">
37   <div class="container-fluid">
38     <div class="navbar-header">
39       <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar"> <span
40         class="icon-bar"></span> <span class="icon-bar"></span> <span class="icon-bar"></span> </button>
41       <a class="navbar-brand" href="#">
42         <p id="brand"> ISMS Portal </p>
43       </a> </div>
44     <div class="collapse navbar-collapse" id="myNavbar">
45       <ul class="nav navbar-nav">
46         <li><a href="index.php"> <span class="glyphicon glyphicon-home"></span> Home</a></li>
47
48         <li class="dropdown"> <a class="dropdown-toggle" data-toggle="dropdown" href="#"> <span
49           class="glyphicon glyphicon-warning-sign"></span> Incident Management <span class="caret"></span>
50         </a>
51         <ul class="dropdown-menu multi-level">
52           <li><a href="CreateIncident.php">Report Incident</a></li>
53           <li><a href="manage_incident.php">Manage Incident</a></li>
54           <li><a href="Selectionincident.php">Incident Log Book</a></li>
55         </ul>
56       </li>
57
58       <li class="dropdown"> <a class="dropdown-toggle" data-toggle="dropdown" href="#"> <span
59         class="glyphicon glyphicon-briefcase"></span> IT Asset Management <span class="caret"></span></a>

```

Figure 4.4 – Header layout and User menu code

include/pagination.php (Pagination Controller)

This file contains the pagination controller code which is used in where ever paginations is required. Figure 4.5 shows code section of the pagination.php file.

```

1 <?php
2 function perpage($count, $per_page, $href) {
3     $output = '';
4     $paging_id = "link_perpage_box";
5     if(!isset($_POST["page"])) $_POST["page"] = 1;
6     if($per_page != 0)
7         $pages = ceil($count/$per_page);
8     if($pages>1) {
9
10        if((($_POST["page"]-3)>0) {
11            if($_POST["page"] == 1)
12                $output = $output . '<span id=1>1</span>';
13            else
14                $output = $output . '<input type="submit" name="page" value="1" />';
15        }
16
17        if((($_POST["page"]-3)>1) {
18            $output = $output . '...';
19        }
20
21        for($i=(($_POST["page"]-2); $i<=(($_POST["page"]+2); $i++))
22        {
23            if($i<1) continue;
24            if($i>$pages) break;
25
26            if($_POST["page"] == $i)
27                $output = $output . '<span id=' . $i . ' class="current-page" >' . $i . '</span>';
28            else
29                $output = $output . '<input type="submit" name="page" value="' . $i . '" />';
30        }

```

Figure 4.5 – Pagination Code Segment

Important Code Segment for User Authentication Module

main.php (View)

this php file contain the UI part of the user login screen. Whenever user not login system will redirect to this login page. Figure 4.6 shows the code section of the main.php source file.

```

75 <form role="form" id="myform" method="post" action="login.php">
76 <fieldset>
77
78 <!-- Form Name -->
79 <legend>Login</legend>
80
81
82 <div class="form-group">
83 <label for="email" id="lemail">Email address:</label>
84 <input type="email" name="email" class="form-control" id="email" value="<?php
if(isset($_COOKIE['email'])) echo $_COOKIE['email']; ?>" required>
85 </div>
86 <div class="form-group">
87 <label for="pwd" id="lpwd">Password:</label>
88 <input type="password" name="pwd" class="form-control" id="pwd" value="<?php
if(isset($_COOKIE['password'])) echo $_COOKIE['password']; ?>" required>
89 </div>
90
91 <div class="checkbox">
92 <label id="lrem">
93 <input type="checkbox" id="remember_me" name="remember_me" <?php
if(isset($_COOKIE['email'])) {echo "checked='checked'"; } ?>
94 Remember me (2 Days)</label>
95 </div>
96 <button type="submit" class="btn btn-primary" id="btnSubmit" name="btnSubmit">Login</button>
97 <input class="btn btn-danger" type="reset" value="Reset">

```

Figure 4.6 Main login page code segment

Login.php (Login Controller)

This file contains the login controller code which control the login process. Figure 4.7 shows the code segment of the login.php.

```

1 <?php
2 session_start();
3 include_once 'class/class.user.php';
4 $user = new User();
5
6 if ( isset( $_POST['btnSubmit'] ) )
7 {
8 $login = $user->check_login($_POST['email'], $_POST['pwd']);
9 if ($login)
10 {
11 // Login Success
12 if(isset($_POST['remember_me'])){ // if user check the remember me checkbox
13     $twoDays = 60 * 60 * 24 * 2 + time();
14     setcookie('email',$_POST['email'] , $twoDays);
15     setcookie('password',$_POST['pwd'] , $twoDays);
16 } else { // if user not check the remember me checkbox
17     $twoDaysBack = time() - 60 * 60 * 24 * 2;
18     setcookie('email', '', $twoDaysBack);
19     setcookie('password', '', $twoDaysBack);
20 }
21 header("location:index.php");
22 }
23 else
24 {

```

Figure 4.7 – Login controller Code segment

Class.user.php (Model)

This file contains the required models for the Login Process and User Management. Figure 4.8 shows code section of the class.user.php .

```

1 <?php
2 include (dirname(__FILE__)."/../include/db_config.php");
3
4 class User
5 {
6 //----- Database connect -----
7 public $db;
8
9     public function __construct(){
10         $this->db = new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_DATABASE);
11
12         if(mysqli_connect_errno()) {
13             echo "Error: Could not connect to database.";
14             exit;
15         }
16     }
17 //----- Return User Details -----
18
19     public function user_detail($email)
20     {
21
22         $sql = "SELECT * from users WHERE Email= '$email'";
23
24         $result = mysqli_query($this->db, $sql);
25
26         if ($result)
27         {
28             return mysqli_fetch_array($result);

```

Figure 4.8 – Authentication system model code segment

Important Code Segment for Functional Modules

(Incident Management, Asset Management, Log Management and Document Management)

CreateIncident.php (View)

This file contains the UI part of the Incident creation system. Figure 4.9 shows the code segment of the createincident.php .


```

60 <fieldset>
61
62 <!-- Form Name -->
63 <legend>Report Incident</legend>
64
65
66 <div class="form-group">
67   <label class="control-label col-sm-4" for="irname">Name of the Incident Reporter (IR)</label>
68   <div class="col-sm-8">
69     <input type="text" class="form-control input-sm" id="irname" name="irname"
70       placeholder="Name of the Incident Reporter (IR)">
71   </div>
72 </div>
73
74
75 <div class="form-group">
76   <label class="col-sm-4 control-label" for="repyby">Reported By</label>
77   <div class="col-sm-4">
78     <select id="repyby" name="repyby" class="form-control">
79       <option value="" selected></option>
80       <option value="01">Employee/Internal</option>
81       <option value="02">Client</option>
82       <option value="03">Vendor</option>
83     </select>
84   </div>
85 </div>
86 </div>
87

```

Figure 4.9 – Incident Creation UI Code Segment

create_incident.php (Controller)

this file contains the code for the incident creation. Figure 4.10 shows the Create incident controller code segment.

```

52 $inddesc = nl2br($_POST['textfeed']);
53 $indrepemail = $_POST['email'];
54 $indreptel = $_POST['tele'];
55 $indbuss = $_POST['beffect'];
56 $indloc = $_POST['indloc'];
57
58
59
60 $sql = "insert into incident (status,indrnam,indrepyby,inddate,indtime,indclass,indtype,indshort,inddesc,
indrepemail,indreptel,indbuss,indloc,indertby) values('A','$indrnam','$indrepyby', '$inddate', '$indtime',
'$indclass', '$indtype', '$indshort', '$inddesc', '$indrepemail', '$indreptel', '$indbuss', '$indloc',
'$user_name')";
61
62 $return = $incident->IndQuery($sql);
63
64 if($return > 0)
65 {
66
67
68 if(isset($_FILES['inddoc'])){
69   $errors= array();
70   foreach($_FILES['inddoc']['tmp_name'] as $key => $tmp_name ){
71     $file_name = rand(100,100000)."-".$_FILES['inddoc']['name'][$key];
72     $file_size =$_FILES['inddoc']['size'][$key];
73     $file_tmp =$_FILES['inddoc']['tmp_name'][$key];
74     $file_type=$_FILES['inddoc']['type'][$key];
75     if($file_size > 3145728){
76       $errors[]='File size must be less than 3 MB';
77

```

Figure 4.10 – Incident Creation controller code segment.

delete_incident.php (Controller)

this file contains the code segment of the delete incident controller. Figure 4.11 shows the code segment of the delete incident controller.

```

1 ▼ <?php
2   require_once("class/class.incident.php");
3   $incident = new incident();
4   require_once("class/class.user.php");
5   $user = new User();
6   $role = $_SESSION['role'];
7   if (!$user->get_session())
8 ▼ {
9     header("location:main.php");
10  }
11  if ($role == 'E1' || $role == 'I1')
12 ▼ {
13      header("location:deny.php");
14  }
15
16  $result = $incident->delete_incident($_GET['id']);
17
18      if($result)
19 ▼      {
20          $incident->delete_incident_doc($_GET['id']);
21          header("Location: " . $_SERVER["HTTP_REFERER"]);
22      }
23
24 ▼ else {
25
26      echo "You Cannot Delete !!";
27  }
28
29

```

Figure 4.11 – Delete incident controller code segment

manage_asset.php

this php file contain the view and controller code segments of the combined asset management system. Figure 4.12 shows the mange asset controller segment of the code.

```

85     $orderby = " ORDER BY AsstNo";
86     if($include01 == "")
87     {
88     if (empty($queryCondition)){
89         $queryCondition .= " where AsstStatus = 'A'";
90     }
91     else {
92         $queryCondition .= " and AsstStatus = 'A'";
93     }
94     }
95
96     $sql = "SELECT * FROM asset INNER JOIN asset_type ON asset.AsstType = asset_type.AsstType INNER JOIN
location ON asset.AsstLoc = location.locco" . $queryCondition;
97
98     $href = 'manage_asset.php';
99
100    $perPage = 10;
101    $page = 1;
102
103
104    if(isset($_POST['page']))
105    {
106        $page = $_POST['page'];
107    }
108    $start = ($page-1)*$perPage;
109    if($start < 0) $start = 0;
110
111    $query = $sql . $orderby . " limit " . $start . "," . $perPage;
112
113    $result = $user->runQuery($query);
114
115    if(!empty($result))
116    {

```

Figure 4.12 – Asset Management Code segment

CreateDocs.php

This code segment contains the view of the Document creation UI. Figure 4.13 illustrate the code segment of the document upload view.

```

>88 </fieldset>
59
60 <!-- Form Name -->
61 <legend>Upload Document</legend>
62
63 <div class="form-group">
64     <label class="col-sm-4 control-label" for="DocCat">Document Category</label>
65     <div class="col-sm-4">
66         <select id="DocCat" name="DocCat" class="form-control">
67             <option value="" selected></option>
68         <?php $document->create_category_dropdown(); ?>
69
70     </select>
71     </div>
72 </div>
73
74 <div class="form-group">
75     <label class="col-sm-4 control-label" for="DocSubCat">Document Sub Category</label>
76     <div class="col-sm-4">
77         <select id="DocSubCat" name="DocSubCat" class="form-control">
78             <option value="" selected></option>
79         <?php $document->create_subcategory_dropdown(); ?>
80
81     </select>
82 </div>

```

Figure 4.13 – Code Segment of Document Upload

CreateUtilizationLog.php (View)

This php file contain the view UI code of the Utilization log entering screen, Figure 4.14 illustrate the utilization log view code segment.

```

▼ <div class="form-group">
  <label class="col-sm-4 control-label" for="System">System
  </label>
  <div class="col-sm-4">
  <select id="System" name="System" id="System" class="form-control">
    <option value="" selected></option>
  <?php $utilization->create_system_dropdown($utilization->db)??>
  </select>
  </div>
</div>

▼ <div class="form-group">
  <label class="control-label col-sm-4" for="LogDate">Date of Utilization</label>
  <div class="input-append date col-sm-3" id="datepicker" data-date="2017/10/31" data-date-
  format="yyyy-mm-dd">
    <span class="add-on"><span class="glyphicon-calendar glyphicon"></span></span>
    <input size="14" type="text" name="LogDate" id="LogDate" placeholder="Date" readonly />
  </div>
  <script type="text/javascript">
  &#36;('#datepicker').datepicker({
    autoclose: true,
  });
  </script>
</div>

▼ <div class="form-group">
  <label class="control-label col-sm-4" for="CPUHigh">CPU High%:</label>

```

Figure 4.14 – Code Segment of Log Creation View

Class.incident.php (Model)

This file contains the model required for the Incident management sub module. Figure 4.15 illustrate the segment of the incident class.

```

4 class incident
5 {
6 //----- Database connect -----
7 public $db;
8
9 public function __construct(){
10     $this->db = new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_DATABASE);
11
12     if(mysqli_connect_errno()) {
13         echo "Error: Could not connect to database.";
14         exit;
15     }
16 }
17
18 //----- Create Incident Type Drop Down -----
19 public function create_incidenttype_dropdown($conn)
20 {
21     $result = mysqli_query($conn, "SELECT * from incident_type");
22
23     while($row = mysqli_fetch_assoc($result))
24     {
25         echo "<option value= '$row[indtype]'">$row[indtyped]</option>";
26     }
27 }
28
29 //----- Create classification Drop Down -----
30 public function create_incidentclass_dropdown($conn)
31 {

```

Figure 4.15 – Incident management Sub Module model code segment

Chapter 5 – Evaluation

5.1. Introduction

Evaluation is the process of determining the final achievement of the work carried out through system development life cycle. Even though system is developed well, it is not a guarantee that system will work properly. To ensure this guarantee a testing must be carried out. Even if a certain system is error free it cannot be considered a very good solution since it is essential to test both validation and verification if it is to be a complete testing.

5.2. Requirement for Testing

Software Verification and Validation mechanism should be maintained thoroughly throughout the system development life cycle which check and analyze the process that ensures the trustfulness of that software to its specification and meeting the needs of the customers who are paying for the software.

- Validation – set of activities that ensure that the software that has been built matches with customer's requirements. (Are we building the right product?)
- Verification – set of activities that ensure the correct implementation of specified functions by the Software. (Are we building the product right?)

5.3. Testing Strategies

The following figure 5.1 shows the software testing levels & sequence.

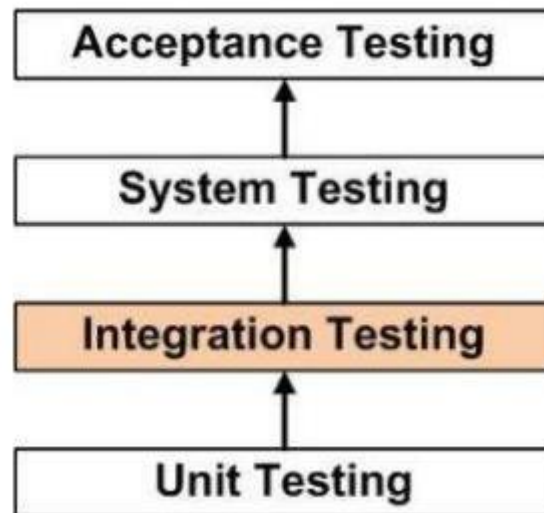


Figure 5.1 – Software Testing Levels

5.3.1. Unit Testing

Unit testing can be done on the smallest unit of software design, the software component or module, to discover the defects in programs. Important control paths are tested to uncover the error within the boundary of the module using the component level design description as a guide. Unit testing focuses on internal processing logic and data structures and it can be done on multiple components simultaneously. There are two common approaches to Unit Testing.

- **White Box Testing** – the program is tested using the test data selected from specification just to ensure that the output is correct. As the logic of the program is not tested by white – box testing only the functional errors of the program will be detected.
- **Black Box Testing** – checks the control structure of the program and it is possible to detect control flow defects. The test data will be selected based on the program logic.

5.3.2. Integration Testing

After the unit testing those modules can be integrated into sub-systems and test the integrated system. After a successful integration testing, the modules of the system should interact as designed and the system should behave as defined in the functional specification. There are several strategies which are used for system integration testing.

- **Top Down Testing** – tests the higher levels of the system before testing its lower level detailed components. After top level components are implemented and tested the lower level components will be implemented and tested until it reaches the lowest level.
- **Bottom Up Testing** – as the reverse of top down testing, the lower level components are tested and then working up the modules until the final components is tested in bottom up testing.
- **Interface Testing** – takes place when modules or sub systems are integrated to create larger systems. It detects faults which may have been introduced into the system because of interface errors.
- **Stress Testing** – designs to ensure that the system can process its intended load. During the stress testing the system is loaded with unbearable load and the load is steadily increased until the system is failed.

5.3.3. Validation Testing

At the end of the system testing, the validation testing begins when the full package of software is available. The purpose of this testing is to ensure that the ultimate software satisfies the functional and performance requirements of the user. There are two approaches which are used for validation testing.

- **Alpha Testing** – the software is tested in a developer-controlled environment by the end-users.
- **Beta Testing** – the software is tested at end-user sites which are not controlled by the developers. All errors recorded by the user will be reported to the developer. These errors are corrected and ironed out the remaining problems with the product before it puts on the general release.

5.4. Testing Procedure

Unit Testing was carried out from the beginning of the development stage when the sub modules are developed. The tested modules are integrated to make sub Systems and Integration Testing is performed. Then the first part of the Validation Testing is carried out in developer-controlled environment to check whether the system satisfies the functional and performance requirements of user. Finally, acceptance testing was carried by the client.

5.5. Test Plan

Test Schedule

Although reviewing and code inspection were done in the designing and implementation phases respectively, a period of three weeks was separately allocated for the testing.

Hardware and Software requirement for the Testing

| Test Environment | |
|--|---|
| Hardware | Software |
| <ul style="list-style-type: none"> • Inter Core i5 2.30 GHz Processor • 4 GB RAM • 1 GB Hard Disk • Desktop PC | <ul style="list-style-type: none"> • Microsoft Windows 8 or Above • XAMPP 3.2.2 or Above <ul style="list-style-type: none"> ○ PHP 7.0.21 or Above ○ MariaDB 10.1.25 ○ Apache 2.4.26 ○ phpMyAdmin 4.7.0 • Web Browser (Chrome) |

Table 5.1 – Test Environment Hardware and Software requirement

5.6. Test Data

Sample test data were given by client from there manual/excel documents. Two members from the Lankem IT team allocated to test the system.

Some of the Test data used shown Below in figure 5.2.

| Asset No. | Asset | Asset Type | Asset Description | Asset Owner | Asset Custodian | Confidentiality Rating (1-3) | Integrity Rating (1-3) | Availability Rating (1-3) | Asset Value | Asset Classification |
|-----------|--|-------------|---|-------------|--------------------------------|------------------------------|------------------------|---------------------------|-------------|----------------------|
| | PC use by Manager IT | Workstation | For Day to day operations | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| | Laptop use by manager IT | Workstation | To store critical documents related to IT department | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| | PCs use by IT staff (11) | Workstation | Day today IT operations | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| LCCPU0151 | LDAP win server | Server | LDAP active directory server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| LCCPU0362 | Anti Virus Server-HP Pro | Server | Anti- Virus Server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| LCCPU0145 | Call Billing server-Dell | Server | Call Billing server – PABX | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| LCCPU0192 | AS400 - I Series | Server | AS400 Application server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| LCCPU0416 | IBM Blade Server | Server | 3 blades (BI Server / HR System server / Solmon Server) | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| LCCPU0319 | LDAP server-HP Pro | Server | LDAP Server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| LCCPU0318 | LDAP Backup server-HP ProLiant MC110G6 | Server | Backup LDAP Server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |
| LCCPU0190 | Mail Server-Dell Vostro 2205 | Server | E Mail Server | GM-IT | Associate SAP Consultant-Basis | 3 | 3 | 3 | 3 | High |

Figure 5.2 – Asset Master Test Data.

5.7. Test Cases

Test Case is used to represent a plan of the inputs to the testing, expected output and priority of the test event. Test cases for this system were defined giving due care to the operational behavior of the system. The test cases were recorded inclusive of information listed below.

- **Module** - The tested module was mentioned above this section.
- **Number** - A uniquely identifiable number was given to each test case for the future purposes.
- **Test Case** - Clear description was given to each test case considering the functionality of each scenario and the actual test data for the relevant functionality.
- **Expected Output** - The requisite output which was mentioned in the system specification.
- **Priority** – Priority for the test case (High, Medium, Low)

Table 5.2, 5.3, 5.4, 5.5 , 5.6 illustrate the test cases for respective modules.

User Authorization Module

| No | Test Case | Expected Result | Priority |
|----|--|---|----------|
| 1 | Enter Invalid User Name and Password | Display error message and prevent user from login. | High |
| 2 | Enter Valid user name and wrong password | Display error “Invalid password”. | High |
| 3 | Login as IT Executive | Only can access the relevant functional Options. | High |
| 4 | Login as CISO | Should be able access all the functional options other than admin functions | High |
| 5 | Login as System Admin | Should be able to use all the options in the system. | High |

| | | | |
|---|---|---|------|
| 6 | Login as ISSC member | Should be able to see reports and documents. No access to creation, edit or delete options. | High |
| 7 | Logout from the system using logout option. | Current session expired, and user redirected to login page | High |

Table 5.2 – User management module test Cases.

Incident Module

| No | Test Case | Expected Result | Priority |
|----|--|---|----------|
| 1 | Incident reporting creation with blank “Incident reporter” field. | Display error message and prevent user creating incident entry. | High |
| 2 | Create incident without selecting reported by. | Display error “Select Reported by”. | High |
| 3 | Create incident without entering incident date. | Display error “Select incident reported date” | High |
| 4 | Create incident without entering incident time | Display error “Select Incident Time” | High |
| 5 | Create incident without selecting the Classification | Display Error “Select Incident Class” | High |
| 6 | Create incident without selecting the incident type. | Display Error “Select Incident Type” | High |
| 7 | Create incident without entering Short Description | Display Error “Enter short description of the incident” | High |
| 7 | Create incident without entering description. | Display Error “Enter Description of the incident” | High |
| 8 | Create incident without entering Incident reporters email address. | Display Error “Enter the email address” | High |
| 9 | Create incident without entering effected business process | Display error “Enter the area or business effected” | High |

| | | | |
|----|--|---|--------|
| 10 | Create incident without selection location of the incident | Display error “Select the location” | High |
| 11 | Incident creation attached file more than 3 MB per file | Display error “File size should be less than 3 MB per file” | High |
| 12 | Incident creation attach file other than XLS, DOC ,PDF or JPEG | Display error “File should be JPG/JPEG/Doc/Xls/PDF File !! | High |
| 13 | Reviewed Incident should not be able to Change, Delete or review again. | System will disable the option for already reviewed incident documents. | High |
| 14 | IT Executive should be able to change or delete the incident before review | System display option to change or delete incident | Medium |

Table 5.3 – Incident Management Module Test cases

Asset Management Sub Module

| No | Test Case | Expected Result | Priority |
|----|---|---|----------|
| 1 | Create Asset with blank Asset no. | Display Error “Enter an Asset No” | High |
| 2 | Create asset with asset number already in the system | Display error “Asset No Already exists in the System” | High |
| 3 | Create asset without entering asset description. | Display error “Enter an description” | High |
| 4 | Create asset without selecting the Brand of the asset | Display error “Please select a brand”. | High |
| 5 | Create asset without entering the asset model | Display Error “Select enter the model” | High |
| 6 | Create asset without entering the serial no. | Display Error “Enter the serial no” | High |

| | | | |
|----|--|---|--------|
| 7 | Create asset without selecting the asset type | Display Error “Select the Asset Type” | High |
| 7 | Create asset without selecting the asset category | Display Error “select the asset category” | High |
| 8 | Create asset without entering Owner of the asset. | Display Error “Please enter the owners name” | High |
| 9 | Create asset without selecting Date of purchase | Display error “Please select the asset procurement date” | High |
| 10 | Create asset without selecting supplier | Display error “Select the supplier” | High |
| 11 | Create asset without selecting the location | Display error “Select the cost center” | High |
| 12 | Create asset without entering the asset purchase value | Display error “enter asset procurement value” | High |
| 14 | Users should not be able to change, delete, or transfer asset which are already retired. | System will disable the options for already retired assets. | High |
| 15 | IT Executive should be able to change, delete, transfer the assets before retirement. | System display options to change, delete and Transfer | Medium |

Table 5.4 – Asset Management Sub Module Test cases

Log Management Sub Module

| No | Test Case | Expected Result | Priority |
|----|---|--|----------|
| 1 | Create Monitoring log selecting System and Date | Display Error “Select a System” and “Select the Log Date” | High |
| 2 | Create Monitoring log for the combination of system and date for which record is already being created. | Display error “Entry for the Selected System Already exists in the System” | High |

| | | | |
|---|---|--|------|
| 3 | Create Job Log without selecting the system. | Display error “Please Select the system” | High |
| 4 | Create job log without selecting the date. | Display error “Please select the job monitoring date”. | High |
| 5 | Create job log without selecting the Job | Display Error “Please Select the Job name” | High |
| 6 | Create job log for a system, date and job combination already exist in the system | Display Error “Record already exist for the combination of System, Date and Job Combination” | High |
| 7 | Change or Delete logs which are already reviewed by CISO | System Will disable the Change and delete options | High |

Table 5.5 – Log management sub module test cases

Document Management Sub Module

| No | Test Case | Expected Result | Priority |
|----|--|---|----------|
| 1 | Upload document without selecting the document category. | Display Error “Select a Document Category” | High |
| 2 | Upload a document without selecting the document sub Category | Display error “Please select a document sub Category” | High |
| 3 | Upload a document without entering the description of the document | Display error “Please enter the meaning full description of the document” | High |
| 4 | Upload a document without Document date | Display error “Please select the document date”. | High |
| 5 | Upload documents other than file extensions .XLS, XLSX, DOC, DOCX, PDF | Display Error “You can only upload documents with following extensions” | High |

Table 5.6 – Document Management Module Tests cases.

5.8. System Evaluation

The system was tested in various test cases and all the test cases were successfully completed. To finalize the evaluation process and to get the user feedback in depth following Software Acceptance Questionnaire was distributed among main system users including all user levels. The Questionnaire is used the “Likert scale” five level feedback points to get users experience and suggestions.

Levels of the Five-point Likert Scale –

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

User Acceptance Testing

Following figure 5.3 shows the received user feedbacks form with user suggestions. It was included 12 questions to cover all the useful objectives of the system.

Objectives

- Accessibility
- Usability
- Clarity of tasks
- Learn ability
- Information availability
- Efficiency & Effectiveness
- Acceptability & Overall Satisfaction

Software Acceptance Questionnaire

Information Security Management System for Lankem Ceylon PLC

Please Mark (X) on your acceptance level

(Your feedback will be treated anonymously and used to improve the Lankem Information Security Management System)

| | Strongly Agree | Agree | Neutral | Disagree | Strongly Disagree |
|--|----------------|-------|---------|----------|-------------------|
| I find the system easy to login and navigate through the menus. | ✓ | | | | |
| Learning to operate the system is easy for me | ✓ | | | | |
| My interaction with the system is easy and smooth, user interface is easy to use | | | ✓ | | |
| It would be easy for me to become skill full in using the system quickly | ✓ | | | | |
| It was easy to report an incident in the system, Search for the existing incident, change the existing incident and delete incident | | | ✓ | | |
| It was easy for me to create an asset in the system, Search for the existing asset, change asset, transfer asset and Retire asset in the system. | | | ✓ | | |
| It was easy to create and maintain the Monitoring logs, Backup logs and batch job logs in the system. | ✓ | | | | |
| It was easy to upload and manage documents in the document management system. | ✓ | | | | |
| Error messages are clear and easy to understand. | ✓ | | | | |
| Using the system, I would be able to improve the productivity and information availability. | ✓ | | | | |
| Reports available in the system satisfy my requirements | | | ✓ | | |
| Over all, I am satisfied with the system | | | ✓ | | |

Suggestions and Improvements

Need Suggested to have more analytical reports.

Figure 5.3 – Questionnaire Sample

Based on the received feedback overall 75% of the users from 6 survey participants happy with the overall system functionality. But some of the participant suggested to add more functionality in automating IT departments paper based process.

Chapter 6 – Conclusion

6.1. Introduction

This is the final chapter of the dissertation text and it will conclude the dissertation with a critical evaluation of the final result of the system and suggestions for future enhancements and upgrades. This chapter also describes the problems encountered with the path to the final result and lessons learnt during the project work.

6.2. Project Critical Assessment

Based on the feedback from the Lankem Management, IT Division employees and end users Information security management system developed meets the requirement and expectation of the Client. Critical functionality of the developed system Incident management, Document management, Log Management and Asset Management have the necessary information and functions required to comply with the ISO 27001 reporting requirements. System was reviewed by external ISO consultant and he concluded that reports from the system comply with the ISO27001 reporting requirement.

Therefore, its fair to say that developed information security management system achieved its objective set during the project kickoff. Even though some of the functionalities were dropped during the scope preparation time due to time constrains system have all the provisions for adding the functionalities in the future.

6.3. Problems encountered

- Analyzing and extracting useful information from the gathered requirements in designing the system was challenging and time-consuming task.
- Time was the main problem as time required for some of the stages was more than as scheduled and expected previously.

- Lack of knowledge in programming with PHP and MySQL and client-side scripting languages required more time than expected on developing some of the modules.
- User requirement changing and new requirement generating was another major problem.
- It was very difficult to meet and discuss relevant users also a considerable problem.

6.4. Future Improvements

Since the Scope of the project is very big. For the project system was scaled down to manage the development, Testing and Documentation time. Source code and training also provided to internal software developers to develop and enhanced the system to meet all the ISMS related Requirements.

Below are the some of the improvements which are in the pipeline after the initial launch.

- Email Notification to Incident reporter about the action taken to rectify the reported incident.
- Enhance the incident management system to include multi-level escalation functionality.
- Allow end users (Non-IT Users) to access the information such as acceptable usage policy, download request forms, directly reporting incident in the system.
- Enhance the incident reporting system to include non-information security related incidents (General IT helpdesk Issue)
- Improve the Authorization mechanism to fully fledge role based authorization.
- Add administrator functionality to mage the portal configuration.

References

- [1]"Lankem Ceylon PLC", Lankem.lk, 2017. [Online]. Available: <http://www.lankem.lk>. [Accessed: 16- Feb- 2017].
- [2]"Systems analyst", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Systems_analyst. [Accessed: 16- Feb- 2017].
- [3]"Requirements analysis", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Requirements_analysis. [Accessed: 16- Feb- 2017].
- [4]"Functional requirement", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Functional_requirement. [Accessed: 16- Feb- 2017].
- [5]"Non-functional requirement", *En.wikipedia.org*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Non-functional_requirement. [Accessed: 16- Feb- 2017].
- [6]"isoTracker Quality Management System & Document Control System", Isotracker.com, 2017. [Online]. Available: <https://www.isotracker.com/>. [Accessed: 16- Feb- 2017].
- [7]"Free Document Management Software - Open source document management for PHP", Free Document Management Software, 2017. [Online]. Available: <http://www.opendocman.com/>. [Accessed: 16- Feb- 2017].
- [8]"Document Management System Software | OpenKM", OpenKM, 2017. [Online]. Available: <https://www.openkm.com/>. [Accessed: 16- Feb- 2017].
- [9]"Software design", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Software_design. [Accessed: 16- Feb- 2017].
- [10]"Waterfall model", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Waterfall_model. [Accessed: 15- Feb- 2017].
- [11]"Rational Unified Process", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Rational_Unified_Process. [Accessed: 16- Feb- 2017].
- [12]"Model–view–controller", En.wikipedia.org, 2017. [Online]. Available: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. [Accessed: 16- Feb- 2017].
- [13]"php[architect] | The site for PHP professionals, Magazine ...". [Online]. Available: <https://www.phparch.com/>. [Accessed: 16- Sep- 2017].
- [14]"Adobe Dreamweaver", En.wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Adobe_Dreamweaver. [Accessed: 15- Sep- 2017].
- [15]p. contributors, "phpMyAdmin", phpMyAdmin, 2017. [Online]. Available: <https://www.phpmyadmin.net/>. [Accessed: 20- Oct- 2017].

[16]"Adobe Photoshop", En.wikipedia.org, 2017. [Online]. Available:
https://en.wikipedia.org/wiki/Adobe_Photoshop. [Accessed: 20- Oct- 2017].

APENDIX A – System Documentations

This documentation provides guidelines for preparing the installation environment and setting up the Information Security Management System.

To install the system, the hardware system and software's chosen for installation should meet the following prerequisites of Hardware and Software illustrated in the table A.1.

| Production Environment | |
|---|--|
| Hardware | Software |
| <ul style="list-style-type: none"> • Inter Corei5 2.30 GHz Processor • 4 GB RAM • 500 GB Hard Disk • Entry Level Server | <ul style="list-style-type: none"> • Microsoft Windows Server 2016 R2 • XAMPP 3.2.2 or Above <ul style="list-style-type: none"> ○ PHP 7.0.21 or Above ○ MariaDB 10.1.25 ○ Apache 2.4.26 ○ phpMyAdmin 4.7.0 • Backup Software |

Table A.1 – Production Environment hardware and Software requirements

Information Security Management System Setup

1. Install XAMPP package with its default settings.

Note- Use the Default installation path for easy management

2. Copy the ISMS folder given in the supplementary CD and paste it inside the htdocs folder in the following paths, Windows Environment with XAMPP installed - the default path would be - C:\xampp\htdocs and Linux Environment with LAMPP installed – the default path would be – /opt/lampp/htdocs

Database Setup

1. Open phpMyAdmin by typing the following URL in the browser's address bar
http://localhost/phpmyadmin/
2. Login by giving the username and password.
3. Create a blank database named "lankemhub".
4. Click the Import tab (shown in figure A.1) and browse through the supplementary CD's database folder (The path would be ../Database/lankemhub.sql) and select lankemhub.sql file.
5. Click the go button to import the SQL into the newly created ISMS database.

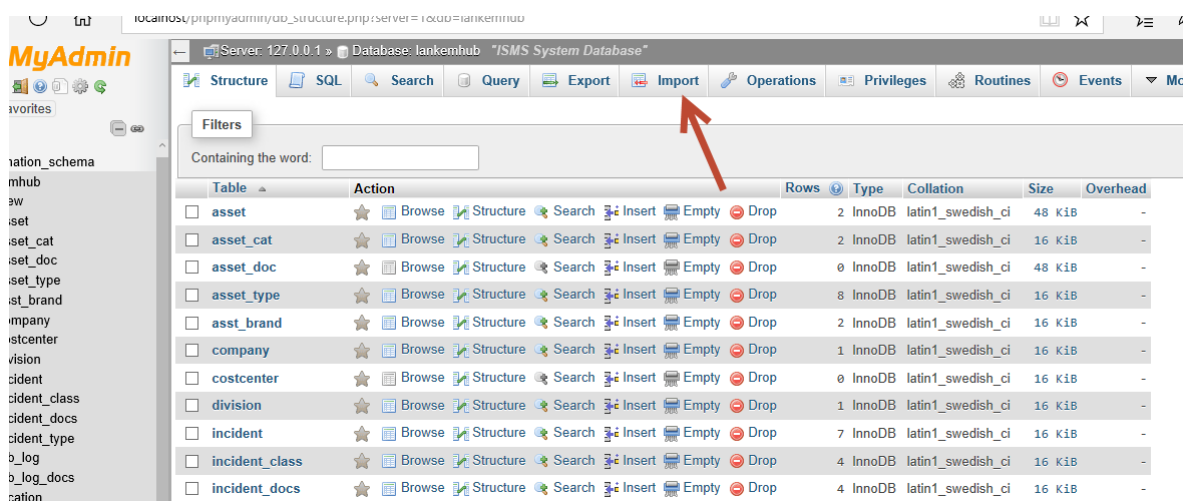


Figure A.1 – Importing lankemhub.sql in to database

Other Configurations

1. Create a folder to store the daily backups in any partition of the Server or Workstation or at a network location.
2. Configure the Database username and password in the include/db_config.php file (illustrated in figure A.2) in this following path, assuming XAMPP package was installed in windows environment

C:\xampp\htdocs\isms\include\helper\db_config.php

```
1 <?php
2 define('DB_SERVER', 'localhost');
3 define('DB_USERNAME', 'root');
4 define('DB_PASSWORD', '');
5 define('DB_DATABASE', 'lankemhub');
6 ?>
```

Figure A.2 – Database parameter configuration

System Usage

Once the isms folder is located in htdocs folder and the database is imported, and the configurations are done;

You can open preferred web browser and type the following URL in the address bar:
<http://localhost/isms/index.php> or <http://127.0.0.1/isms/index.php>

If you could the rename the Server name to isms you will be able to open the system by typing the following URL in the address bar

<http://isms/index.php>

And Login to gain access, by providing correct username and password.

APENDIX B – Design Documentation

Use-Case Diagram for the Proposed System

Figure B.1 illustrate the higher-level use-case of the proposed system.

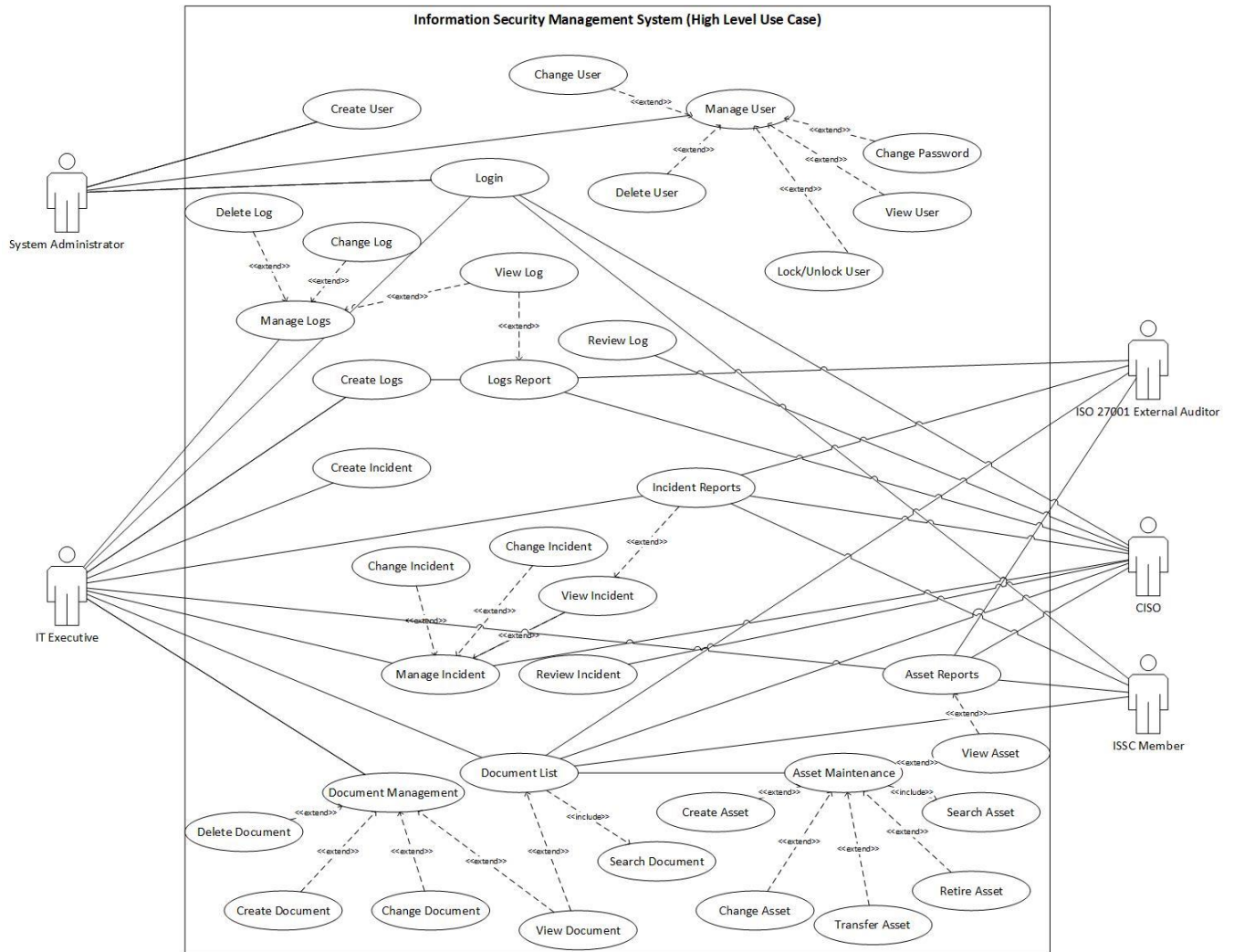


Figure B.1 - High level use case diagram

User Management and Authentication Sub module

User management and Authentication submodule will handle the Creation, Change, Lock, Password reset, Login Process and authorization functionalities.

Figure B.2 and Table B.1 illustrate the use-cases of the Module user management and Authentication.

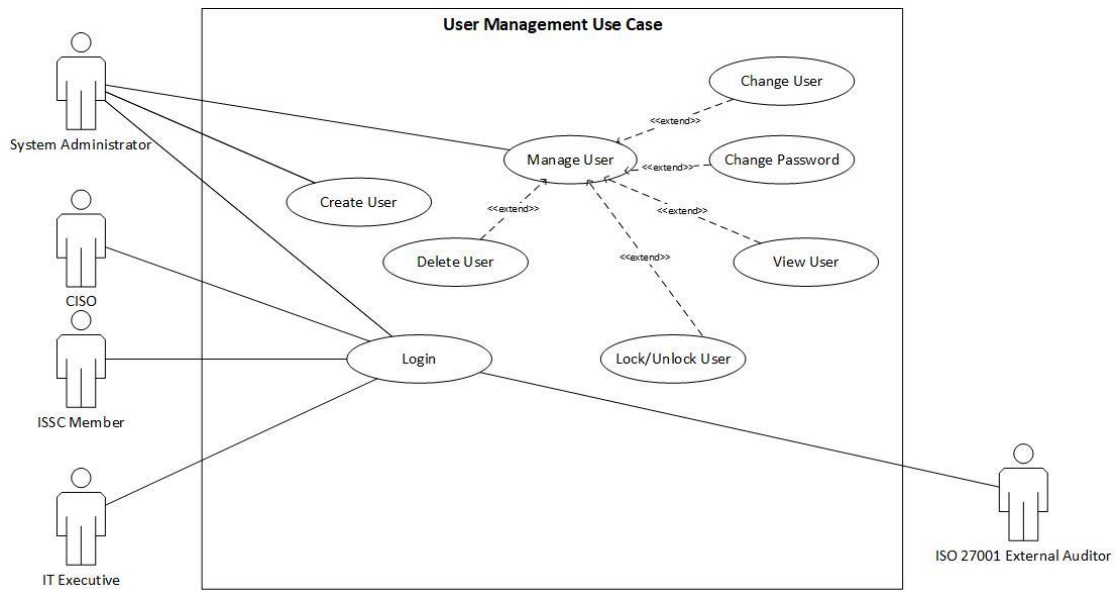


Figure B.2 – User Management Use Case

| | |
|------------------------|--|
| Use-Case | Login to the System |
| Actors | System Admin, IT Executive, CISO, Auditors and ISSC Members. |
| Overview | Authorized users log in to the system |
| Preconditions | <ul style="list-style-type: none"> • User should have an account in the system. • Account should be in unlocked state, |
| Flow of Events | <ul style="list-style-type: none"> • The user enters user name and password. • If entry is invalid, system will throw an error message. • If the username and password is valid the system redirects the user to his/her appropriated home page where access is restricted according to the account privileges. |
| Post Conditions | Authorized users will be logged in the system and will be re directed to the home page. Where they will be able to access the menu items for which they are authorized. |

Table B.1 – Use Case Diagram for Login Process

| | |
|---|----------------------|
| Use-Case | User Management |
| Actors | System Administrator |
| Overview | |
| Create and Manage users to access the system. | |
| Preconditions | |
| <ul style="list-style-type: none"> • Person wanted to have access to the Information security management should forward their request. • Request must be approved by the Chief information security officer. | |
| Flow of Events | |
| <ul style="list-style-type: none"> • User request for the system access. • Request will be forwarded to CISO Approval. • If the request is approved by the CISO, System admin will create the User account and notify the user. • In case if user wanted to change to his user account, he or she should send the request to CISO, once approved System Admin will change the necessary user management data in the system. | |
| Post Conditions | |
| Only authorized personal can gain access to the system and restricted to allowed functionalities. | |

Table B.2 – User Management use case

Asset Management Sub Module Use-Case Diagram

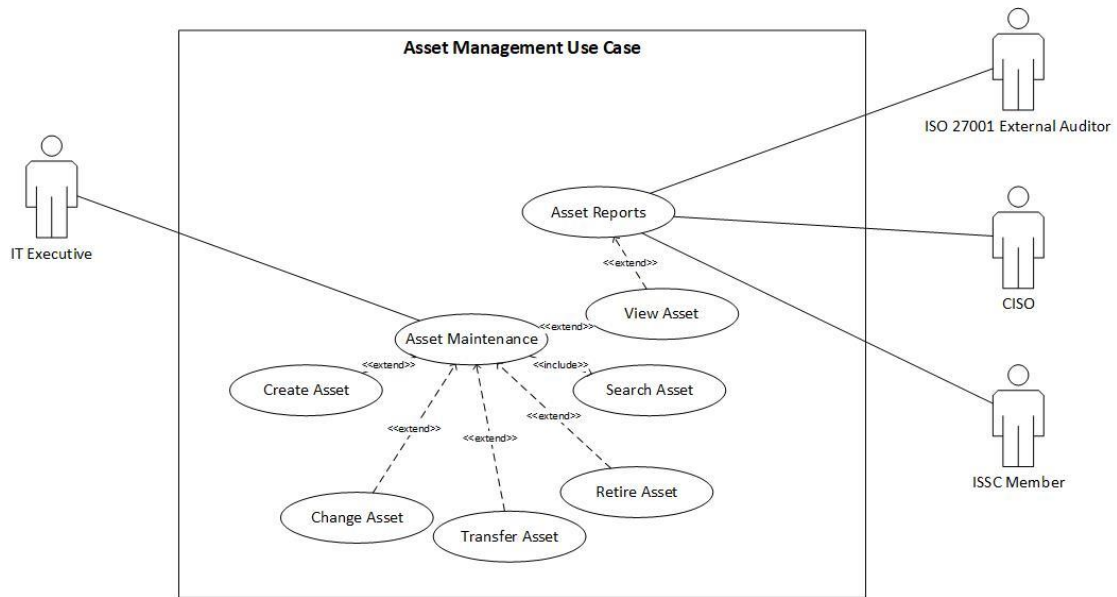


Figure B.3 – Asset Management Use Case

| | |
|------------------------|---|
| Use-Case | Asset Management |
| Actors | IT Executive |
| Overview | |
| | Create and Manage Information Technology Assets in the system |
| Preconditions | |
| | <ul style="list-style-type: none"> Asset should be physically available in the system. Asset should be pasted with a Unique barcoded preprinted sticker. Barcoded number of the asset should be available before creating the asset in the system. |
| Flow of Events | |
| | <ul style="list-style-type: none"> Once the Asset physically available and manual barcoded number is assigned Asset information will be fed in to the system. Any mistakes and changes to the entered asset can be changed. Transfer from one location to another, one cost center to another and change of owner can be done using the transfer option. When the asset reaches the end of life asset retirement can be performed. |
| Post Conditions | |
| | Effectively manage the IT asset Inventory |

Table B.3 – Asset Maintenance Use Case

Log Management Sub Module Use-Case Diagram

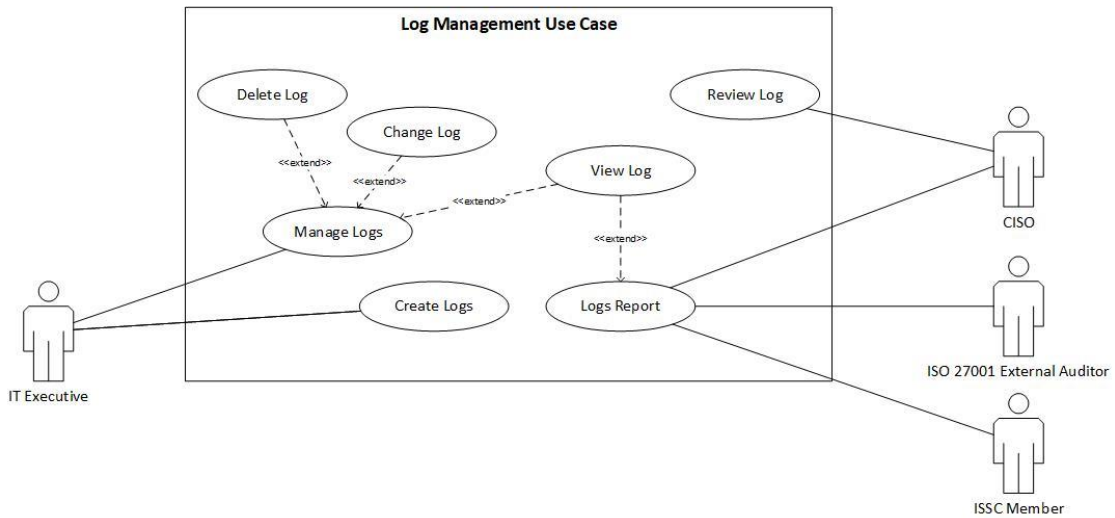


Figure B.4 – Log Management Use Case

| | |
|--|----------------|
| Use-Case | Log Management |
| Actors | IT Executive |
| Overview | |
| Create and Manage System Monitoring and Job Monitoring Log. | |
| Preconditions | |
| <ul style="list-style-type: none"> Monitoring system should be checked and gather the required information from the systems before creating the log entry in the ISMS system. | |
| Flow of Events | |
| <ul style="list-style-type: none"> Once the IT executive complete the Daily monitoring activates and gather required information such as CPU Utilization, Job start time, Job End time and etc. Gathered information's will be entered to the system. If any mistakes in the entered data using the change option data can be changed. Only changes possible is before the review | |
| Post Conditions | |
| Effectively manage the daily monitoring logs and status. | |

Table B.4 – Log Management Use Case

Incident Management Sub Module Use-Case Diagram

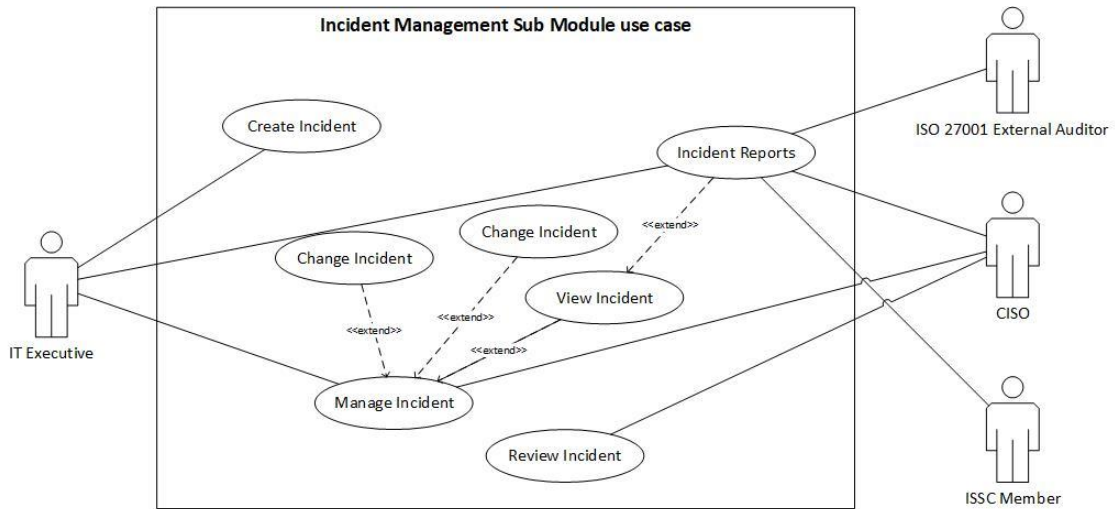


Figure B.5 – Incident Management Use Case

| | |
|------------------------|---|
| Use-Case | Incident Management |
| Actors | IT Executive, CISO |
| Overview | Create and Manage Information Security Related incidents. |
| Preconditions | <ul style="list-style-type: none"> Employees should inform any information security related incidents to IT Executive. |
| Flow of Events | <ul style="list-style-type: none"> Once the IT executive receive a complaint from the employees He/She will create the incident in the system. Any supporting documents can be uploaded. Changes to the already created incident can be made using the change. CISO will review the incident and update the action taken to mitigate the re occurrence. |
| Post Conditions | Effectively managing the incidents for future analysis when improving the information security. |

Table B.5 – Incident Management Use Case

Class diagram for the Proposed System

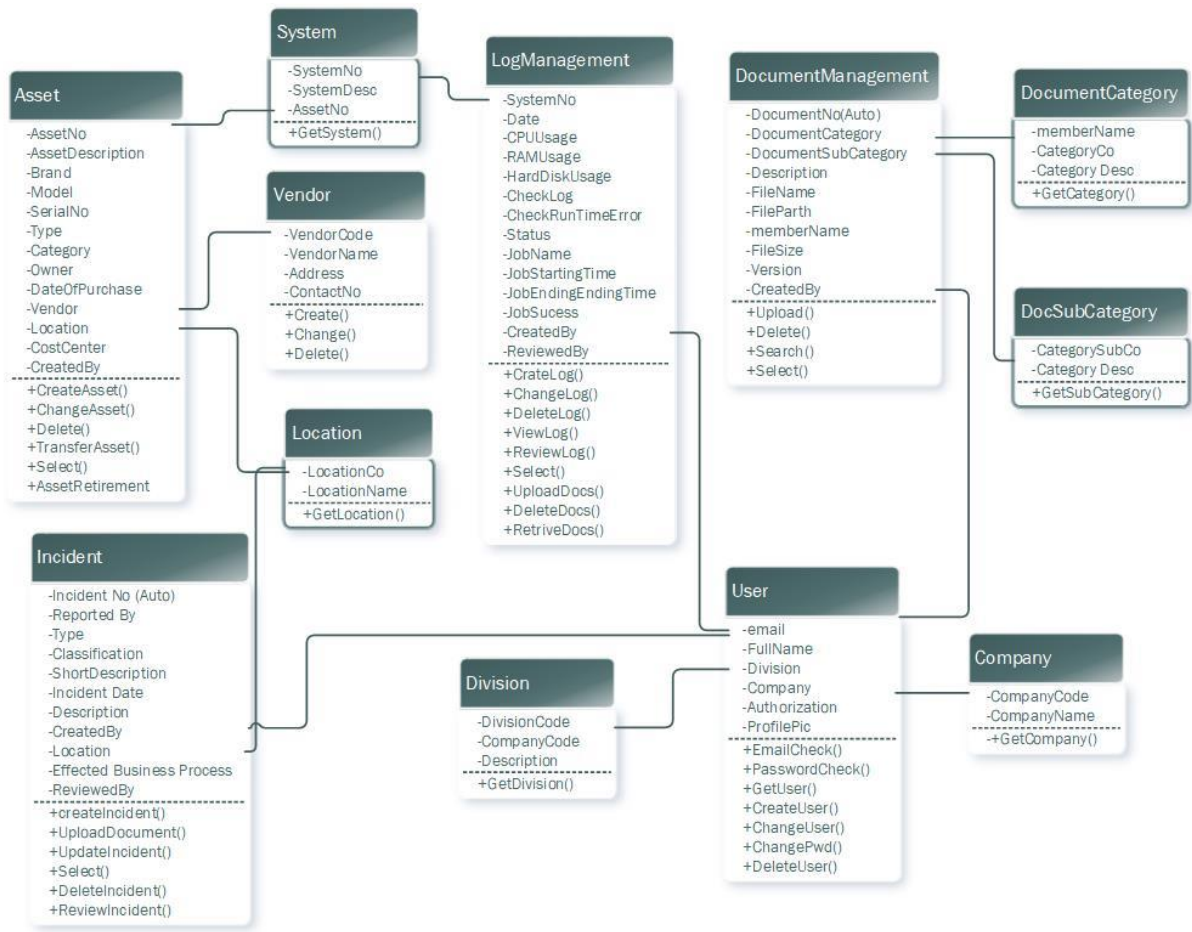


Figure B.6 – Class Diagram for Proposed System

Database Diagram

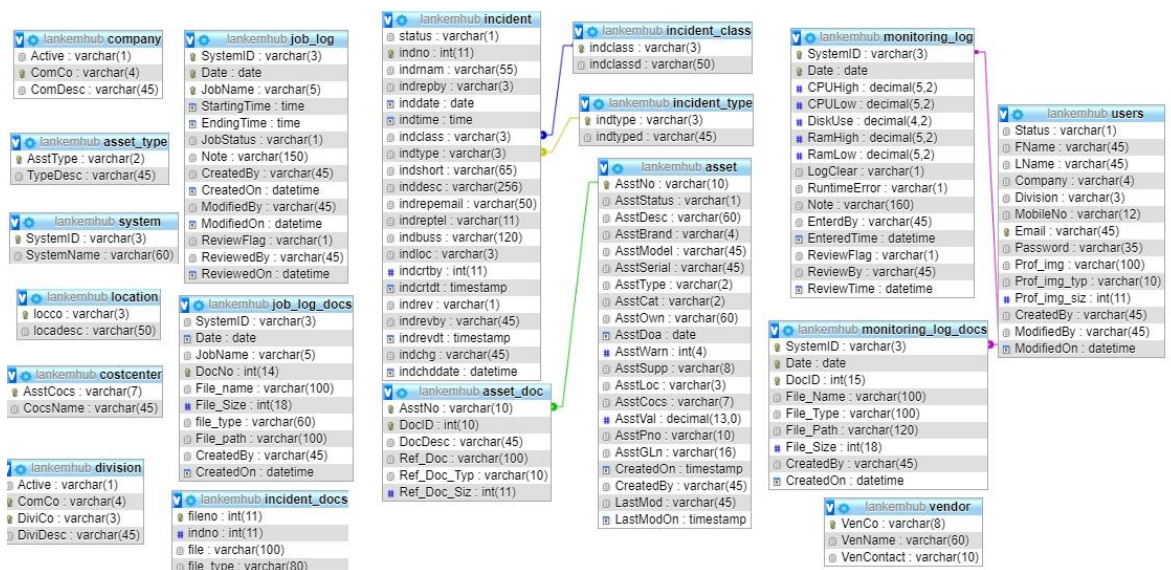


Figure B.7 - Database Diagram

Activity Diagram from user Creation

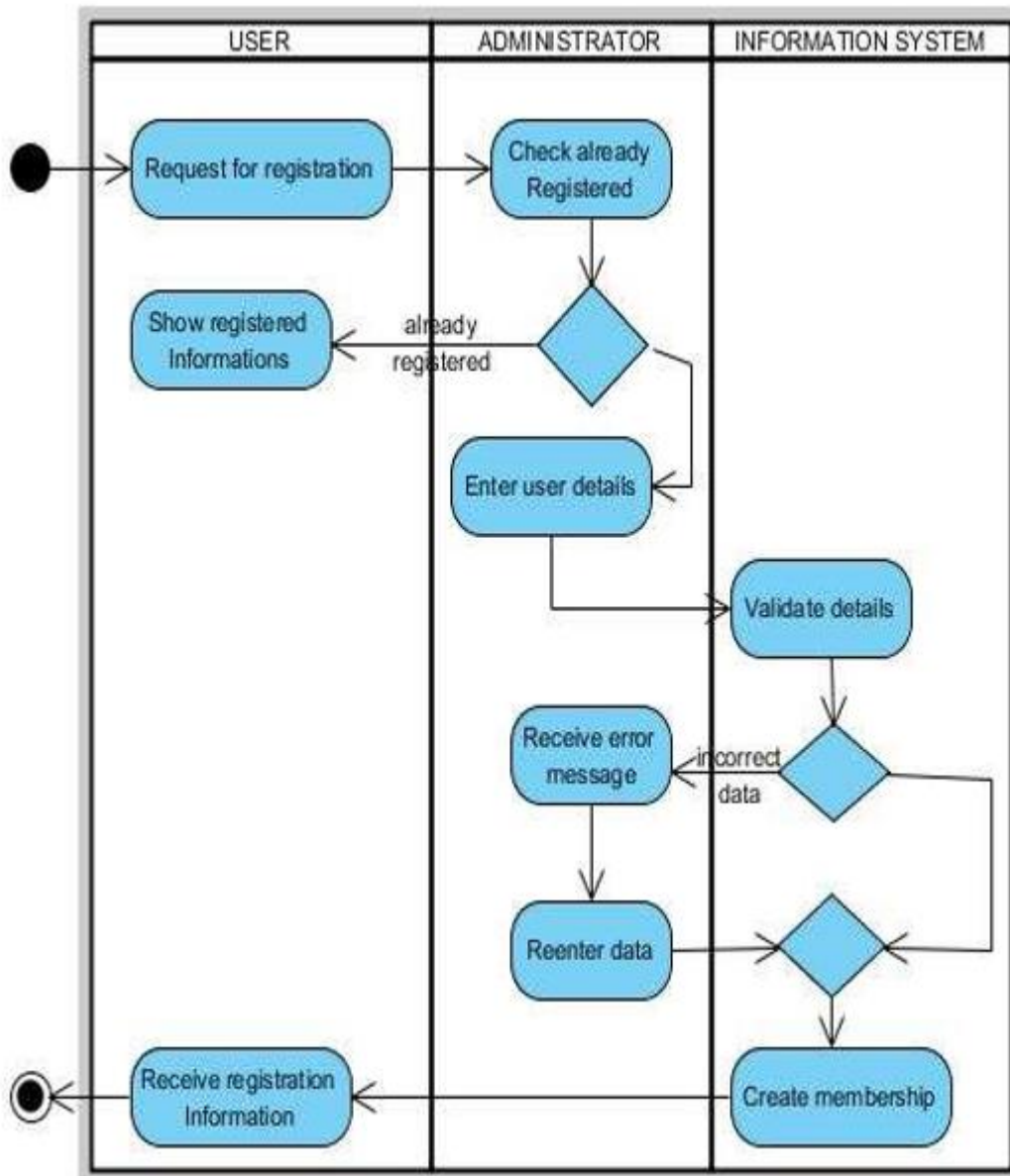


Figure B.8 – Activity Diagram of user Creation

APENDIX C – User Manual

Login to the system

1. Open your web Browser and type <http://192.168.190.20/isms> .
2. Home page of the system will be loaded if you are not logged in already system will redirect to the login page.
3. Enter the User name and Password provided by the Administrator as shown in the Figure C.1 . Note: User name will be your Email address.



Figure C.1 – Login Screen

4. If the Login is successful system will redirect to the Home page of the system.

Logout of the System

1. In the home page Right hand side of the corner “Logoff” button is available as shown in the Figure C.2, once you click the logoff button system will log you out of the system.

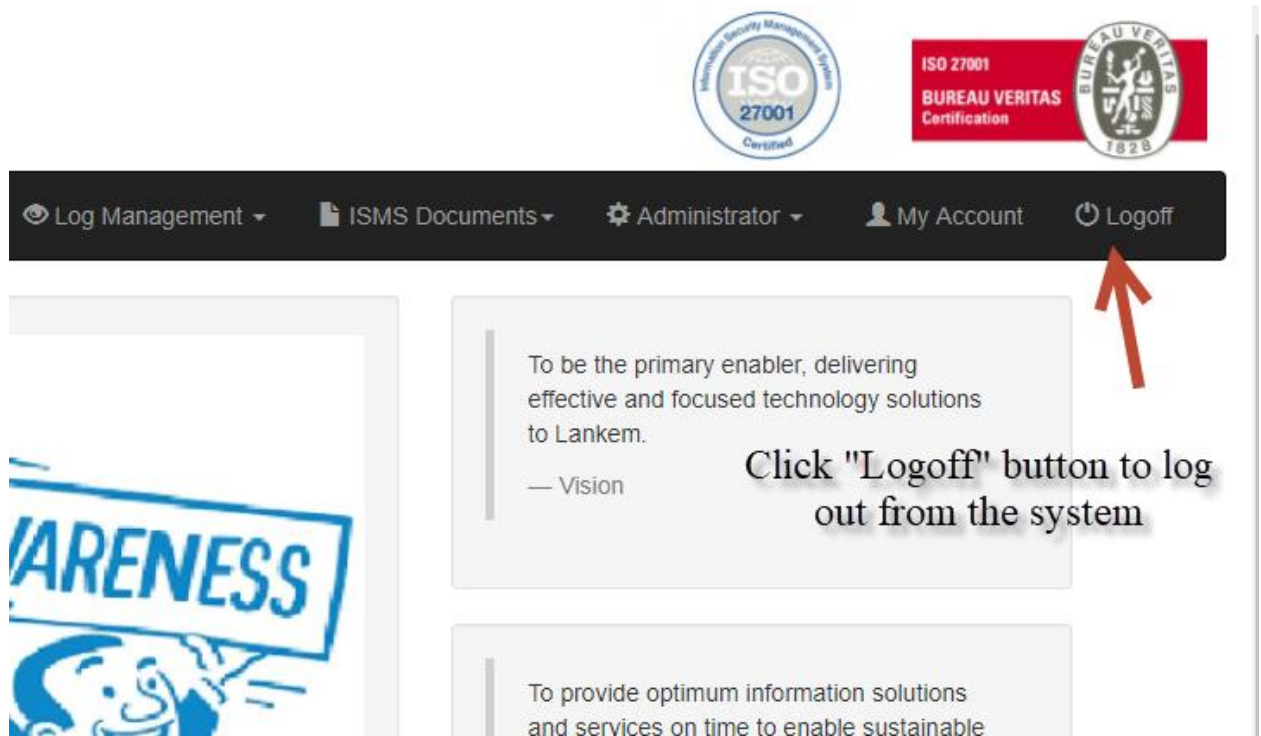


Figure C.2 – Logoff menu

Incident Management: Report an Incident

1. From the main menu select “Incident Management” then from the sub menu select “Report Incident” as shown in the Figure C.3.

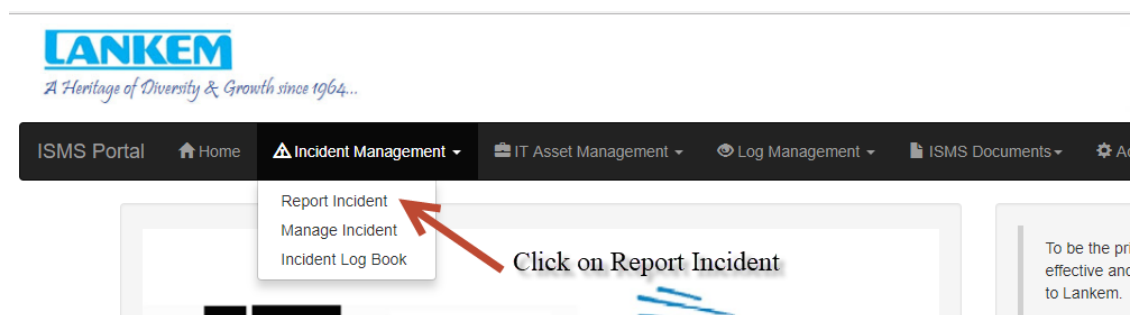


Figure C.3 – Report Incident menu path

2. Enter the Required Details as shown in the figure C.4. Note: all the fields are mandatory other than the attachments.

Report Incident

Name of the Incident Reporter (IR)

Reported By

Incident Date

Incident Time

Classification of the Incident

Incident Type

Short Description

Description

Email of IR

Telephone of IR

Business Unit/Process Effected

Physical Location of the Incident

Attachments

Figure C.4 – Incident Creation Screen

- **Name of the Incident Reporter** – Enter the Name of the person who reported the incident.
- **Reported By** – Select from the dropdown list type of the reporter.
- **Incident Date** – Select the Date of the Incident occurred.
- **Incident Time** – Select the Time of the incident
- **Classification of the incident** – Select the incident classification from the drop-down.
- **Incident Type** – Select the Incident type from the drop down.
- **Short Description** – Enter the short description of the incident
- **Description** – Enter the detail description of the Incident.

- **Email IR-** Enter the Incident reporters email address.
- **Business Unit/Process effected** – Enter the effected business unit or the process effected by the security incident.
- **Physical Location of the incident** – Select the physical location of the incident from the dropdown.
- **Attachments** – if there are any documents to attached select the documents. Multiple attachments also possible.

3. Click submit to save the Incident. System will return you a Incident Number.

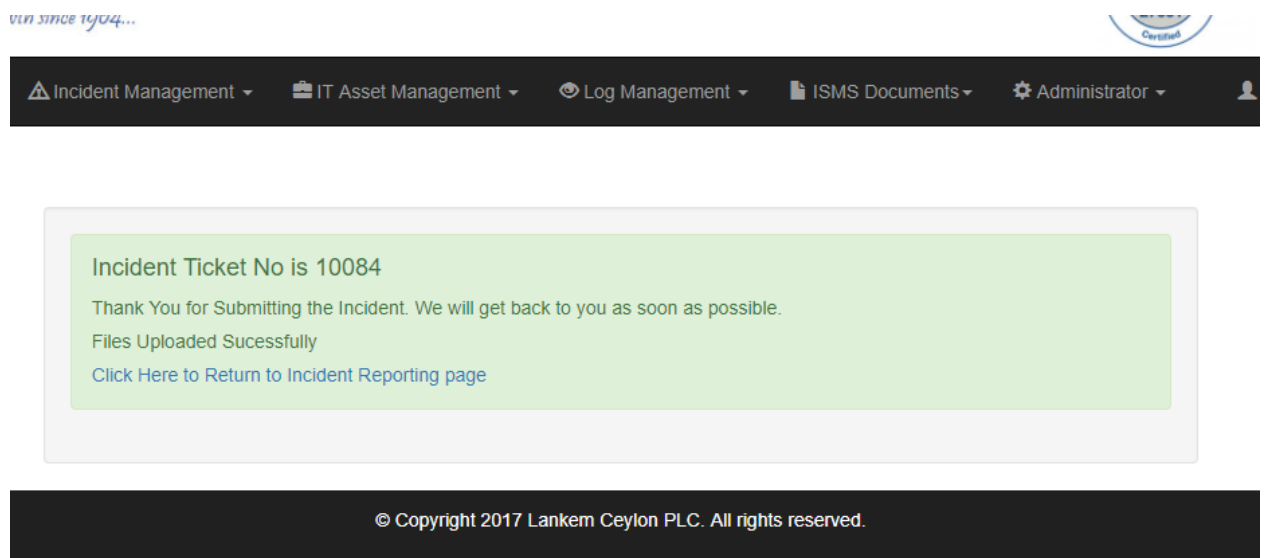


Figure C.5 – Incident Ticket Number

Incident Management: Manage Incident

1. From the main menu select “Incident Management” then from the sub menu select “Manage Incident” as shown in the figure C.6.

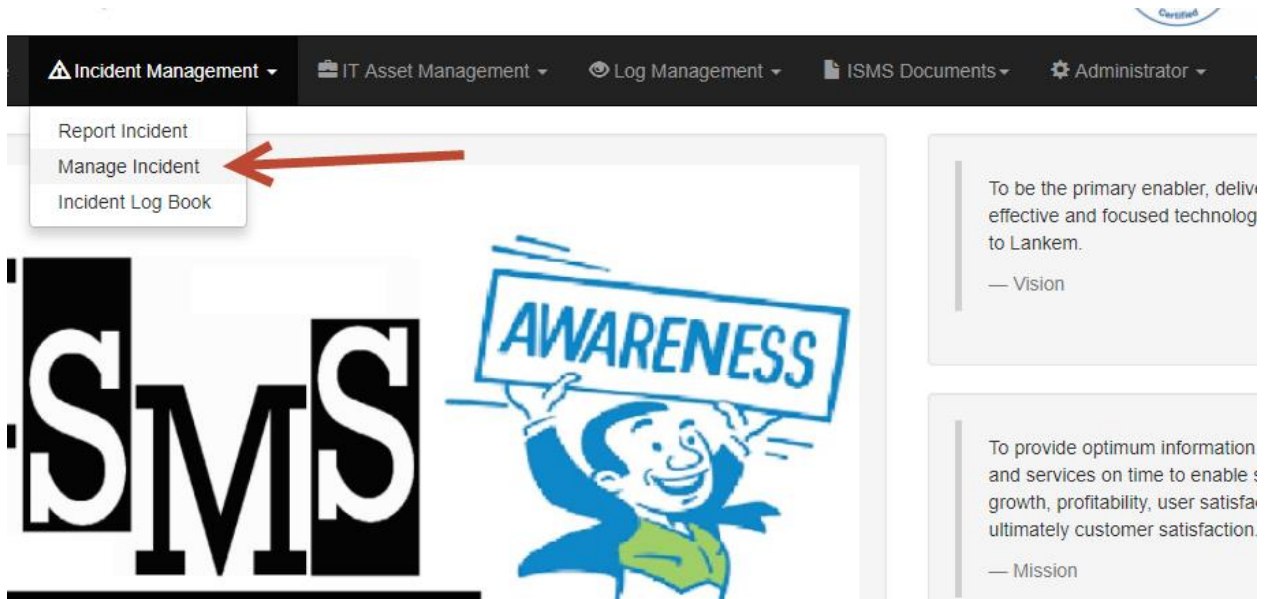


Figure C.6 - Menu Incident Manage

- Once the Page is loaded you will only see the documents which are pending for CISO review. If you want to see the already reviewed Incident you must select “Include reviewed” and click “search” as shown in the figure C.7

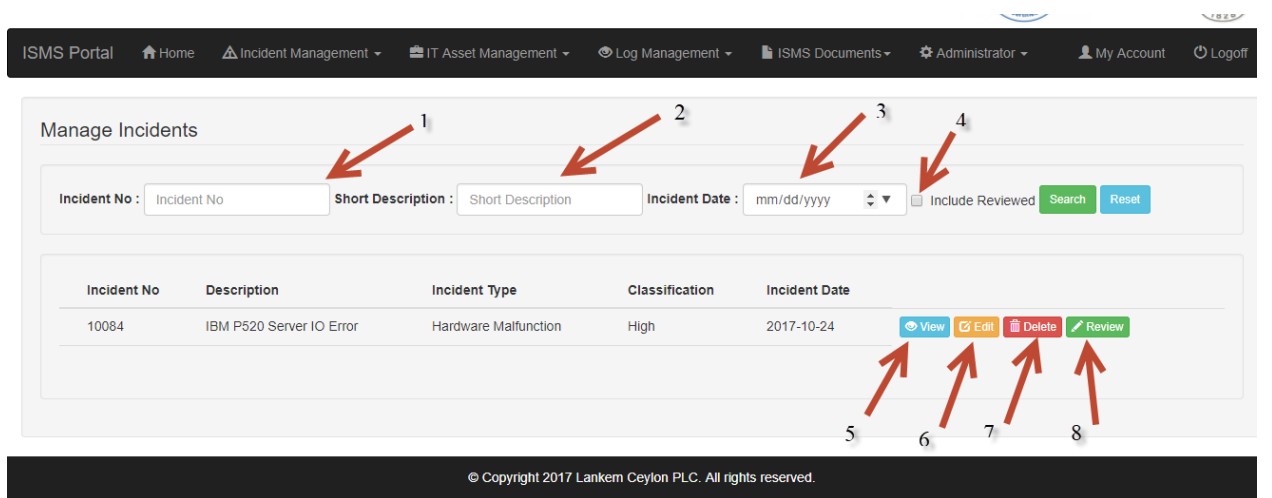


Figure C.7 – Manage Incidents

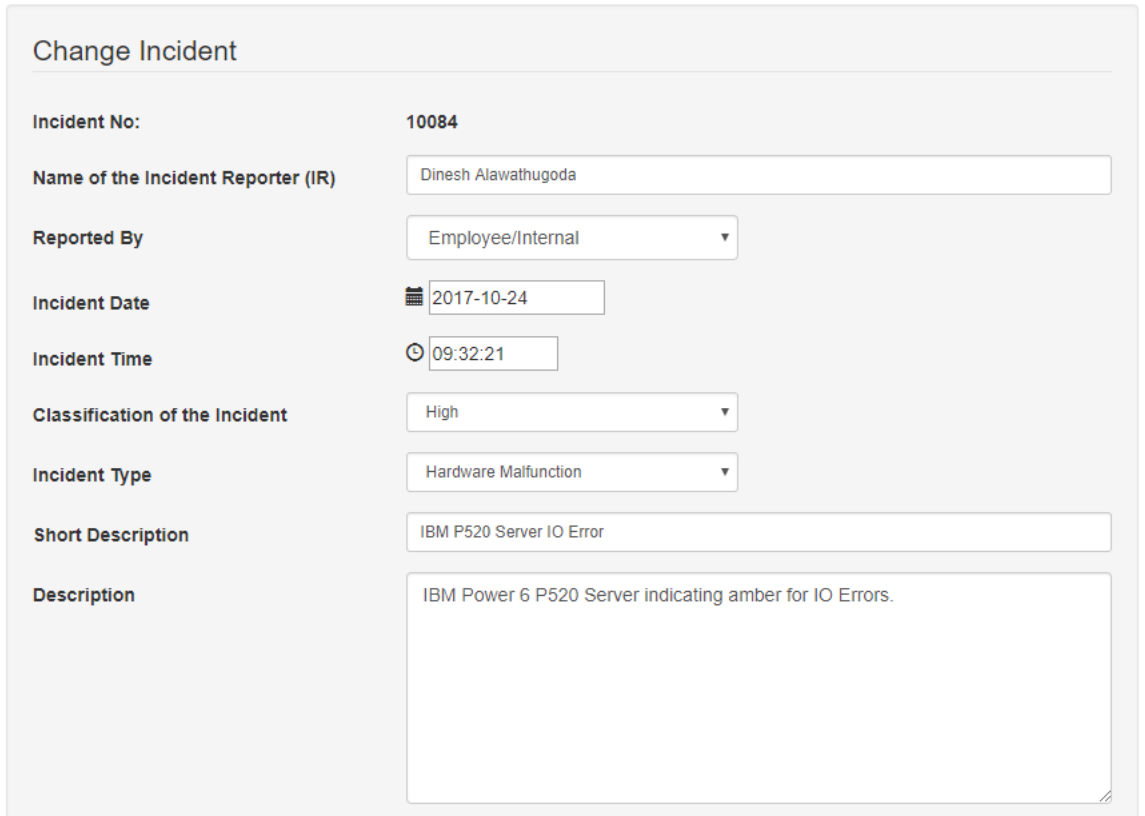
Using options 1,2,3,4 you can search for Incidents using any combinations of search criteria.

If you want to view the incident, click “View” button in front of the incident in the list (Option 5).

| View Incident | |
|---|---|
| Incident No: | 10084 |
| Name of the Incident Reporter (IR) | Dinesh Alawathugoda |
| Reported By | Employee/Internal |
| Incident Date | 2017-10-24 |
| Incident Time | 09:32:21 |
| Classification of the Incident | High |
| Incident Type | Hardware Malfunction |
| Short Description | IBM P520 Server IO Error |
| Description | IBM Power 6 P520 Server indicating amber for IO Errors. |
| Email of IR | dinesh@lankem.lk |
| Telephone of IR | 0772702502 |

Figure C.8 – View Incidents

If you like to change the incident before the CISO review. You can click on the “Edit” button in front of the incident as shown in the figure C.9 . (Option 6)



Change Incident

Incident No: 10084

Name of the Incident Reporter (IR): Dinesh Alawathugoda

Reported By: Employee/Internal

Incident Date: 2017-10-24

Incident Time: 09:32:21

Classification of the Incident: High

Incident Type: Hardware Malfunction

Short Description: IBM P520 Server IO Error

Description: IBM Power 6 P520 Server indicating amber for IO Errors.

Figure C.9 – Change Incident

If you want to delete the mistakenly entered Incident you can delete using the “Delete” Button in front of the incident. system will prompt you for the confirmation before deleting. Note: deletion is recoverable.

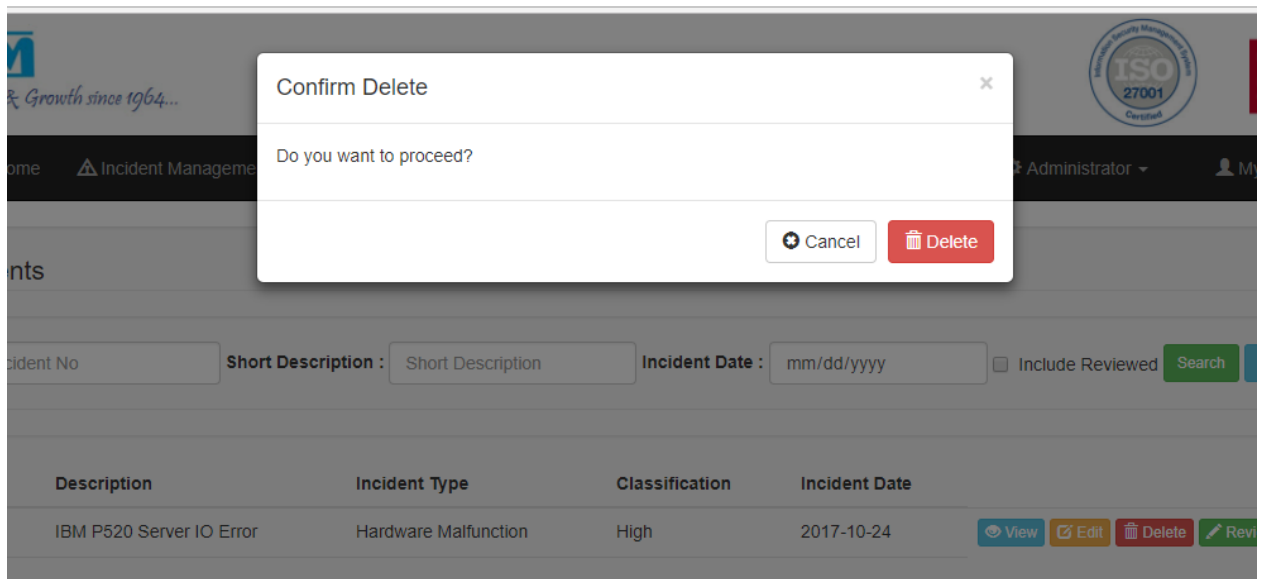


Figure C.10 – Confirm Delete Incident

Incident Review

Only CISO have the authorization to review the incident. Click the “Review” button in front of the incident to review as shown in the figure C.11.

Review Incident

| | |
|------------------------------------|--|
| Incident No: | 10084 |
| Name of the Incident Reporter (IR) | Dinesh Alawathugoda |
| Incident Date | 2017-10-24 |
| Incident Time | 09:32:21 |
| Short Description | IBM P520 Server IO Error |
| Description | IBM Power 6 P520 Server indicating amber for IO Errors. |
| Review Comments | <div style="border: 1px solid #ccc; height: 100px;"></div> |

Figure C.11 – Review incident

Incident Management: Incident Log Book

1. From the main menu select “Incident Management” then from the sub menu select “Incident Log Book” as shown in the figure C.12. Incident log book is a printable report of the reported incidents and review comments.

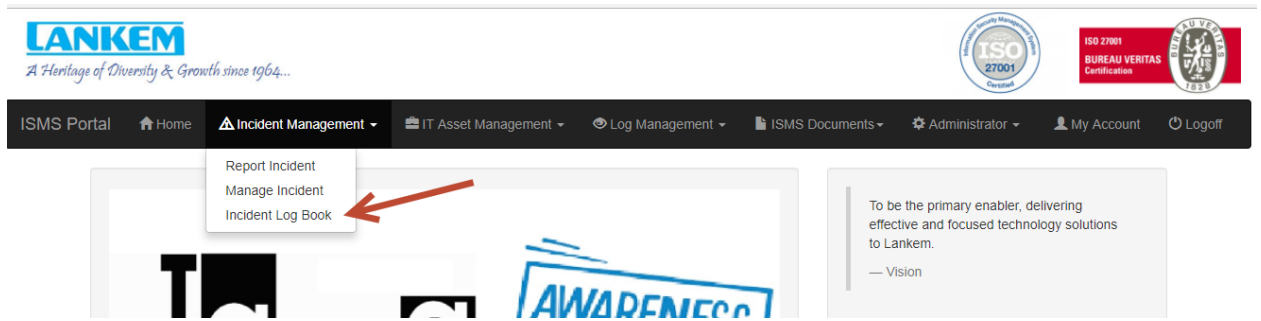


Figure C.12 - Incident Log Book

IT Asset Management: Create IT Asset

From the main menu select “IT Asset Management” then from the sub menu select “Create IT Asset”.

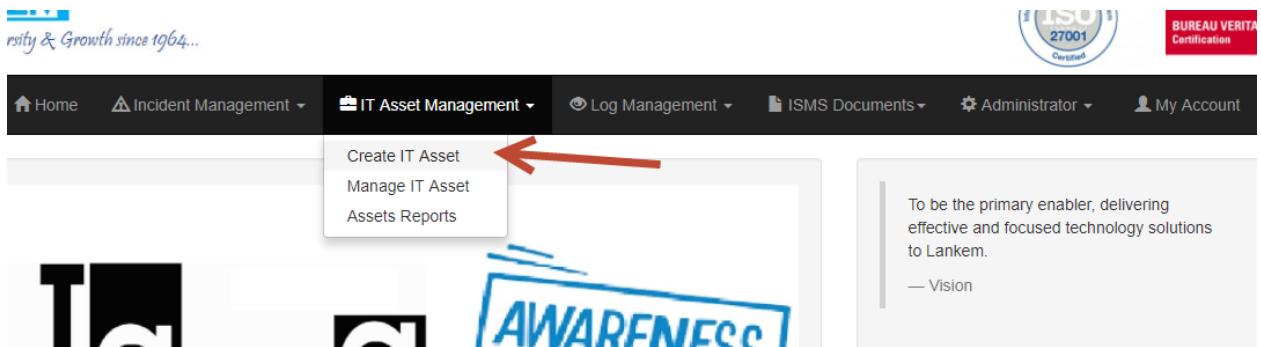


Figure C.13 – Create IT Asset Menu

Then Enter the required details and Click “Create Asset” Button as shown in the figure C.13.

Create Information Technology Asset

| | |
|----------------------------------|---|
| Asset No: | <input type="text" value="Asset Numeber(Maximum 12 Characters)"/> |
| Asset Description: | <input type="text" value="Asset Description(Maximum 60 Characters)"/> |
| Brand | <input type="text" value=""/> |
| Asset Model: | <input type="text" value="Asset Model(Maximum 45 Characters)"/> |
| Asset Serial No: | <input type="text" value="Asset Serial No(Maximum 45 Characters)"/> |
| Asset Type | <input type="text" value=""/> |
| Category | <input type="text" value=""/> |
| Asset Owner | <input type="text" value="Asset Owner(Maximum 60 Characters)"/> |
| Date of Purchase | <input type="text" value="Date of Purchase"/> |
| Warranty Period in Months | <input type="text" value="Asset Warranty Period(in Months)"/> |
| Asset Supplier | <input type="text" value=""/> |
| Asset Location | <input type="text" value=""/> |
| Cost Center | <input type="text" value=""/> |
| Asset Value | <input type="text" value="Asset Value in Rs."/> |

Figure C.14 – Create IT Asset Screen

- **Asset No** – Enter the Asset number pasted in the asset. Manual barcoded stickers.
- **Asset Description** – Enter the Description of the Asset. Example: HP Laptop 4520S
- **Brand** – Select the brand from the dropdown list.
- **Asset Model** – Enter the Asset Model
- **Asset Serial No** – Enter the Serial no of the asset.
- **Asset Type** – Select Asset Type from the Drop-down list
- **Category**- Select the category from the drop-down list
- **Asset Owner** – Enter the name of the asset owner.

- **Date of purchase** – Enter the date of purchase
- **Warranty periods in months** – enter the warranty period in months
- **Asset Supplier** – Select the supplier from the drop-down list.
- **Asset Location** – Select the location from the list
- **Cost Center** – Select the cost center from the list
- **Asset Value** – enter the asset value.

Finally click “Create Asset” button to create the asset in the system.

IT Asset Management: Manage IT Asset

1. From the main menu select “IT Asset Management” then from the sub menu select “Manage IT Asset” as shown in the figure C.15.

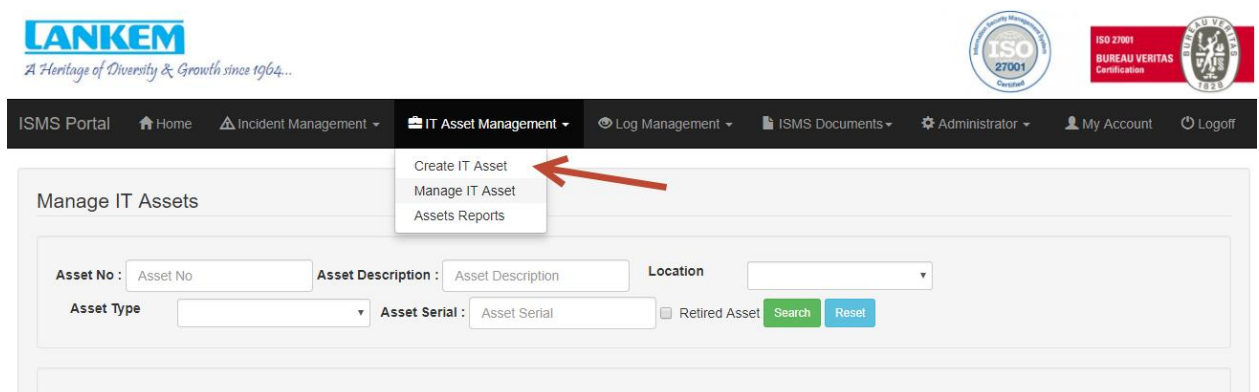


Figure C.15 – Manage IT Assets

2. From the “Manage IT Asset” user can perform following actions as shown in the figure C.16.
 - a. View – View the details of the Asset
 - b. Edit – Make Changes to Asset.
 - c. Delete – Delete an asset.

- d. Transfer – Transfer from one location to another, Transfer from one owner to another and one cost center to another cost center.
- e. Retire – Dispose the assets.

Entire life cycle of the asset can be managed with only this option.

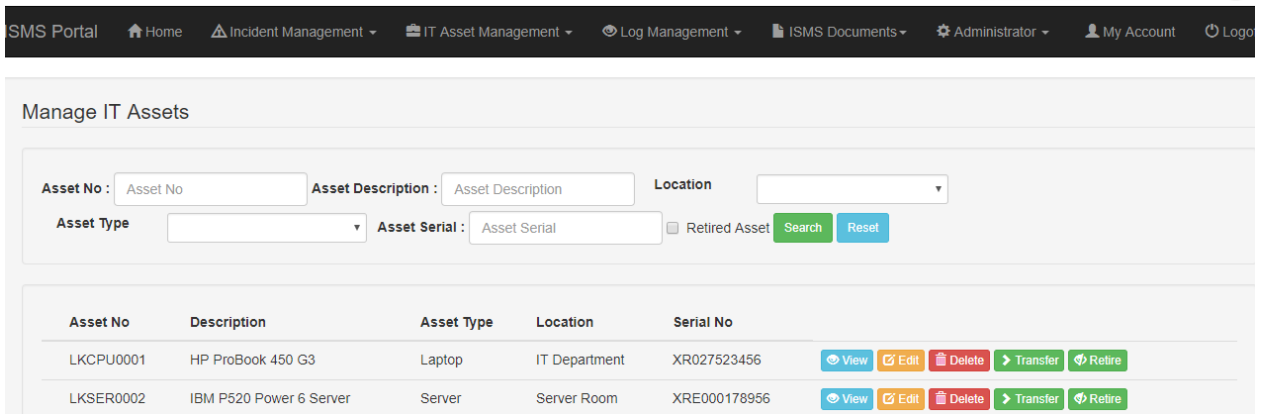
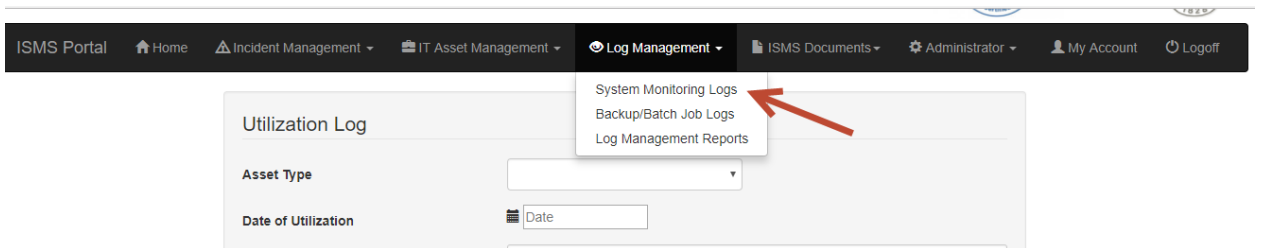


Figure C.16 – Mange IT Asset

Log Management: System Monitoring Logs

1. Select “Log management” from the main menu then click the “utilization log” from the sub menu to enter the daily System monitoring activities of the critical servers and other devices.

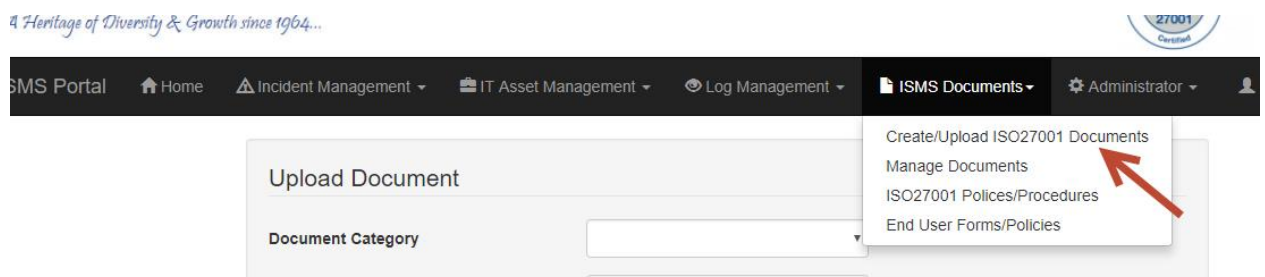


2. User then can enter as shown in the C.17 CPU utilization of the system, Disk utilization, Ram Utilization etc. and save the log. This must be carried out for all the critical systems.

Figure C.17 – Log management

ISMS Documents: Create/Upload ISO27001 Documents

1. From the main menu select “ISMS Documents” then from the sub menu select “Create/Upload ISO27001 Documents”.



2. Then Select the Document Category and Document sub Category from drop down list, Enter the description and select the document and click upload to upload the document as shown in the figure C.18.

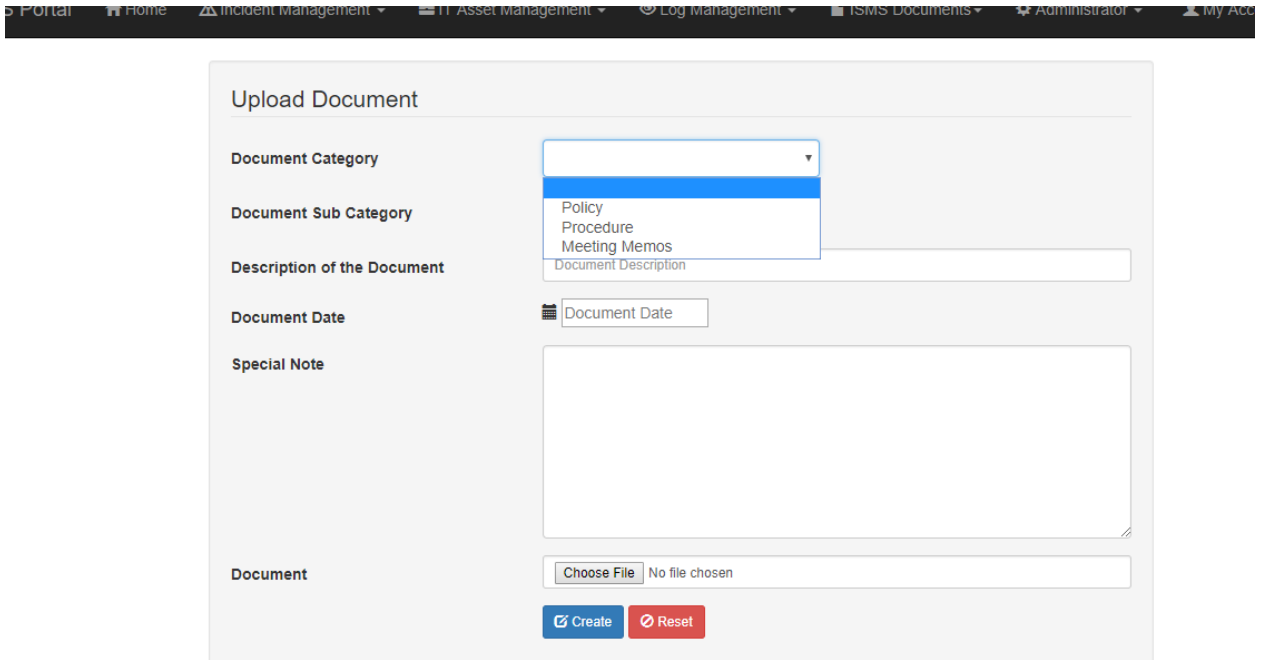


Figure C.18 – Upload Documents

Administrator: User Management

1. From the main menu select “Administrator” then from the sub menu select “Manage Users”.

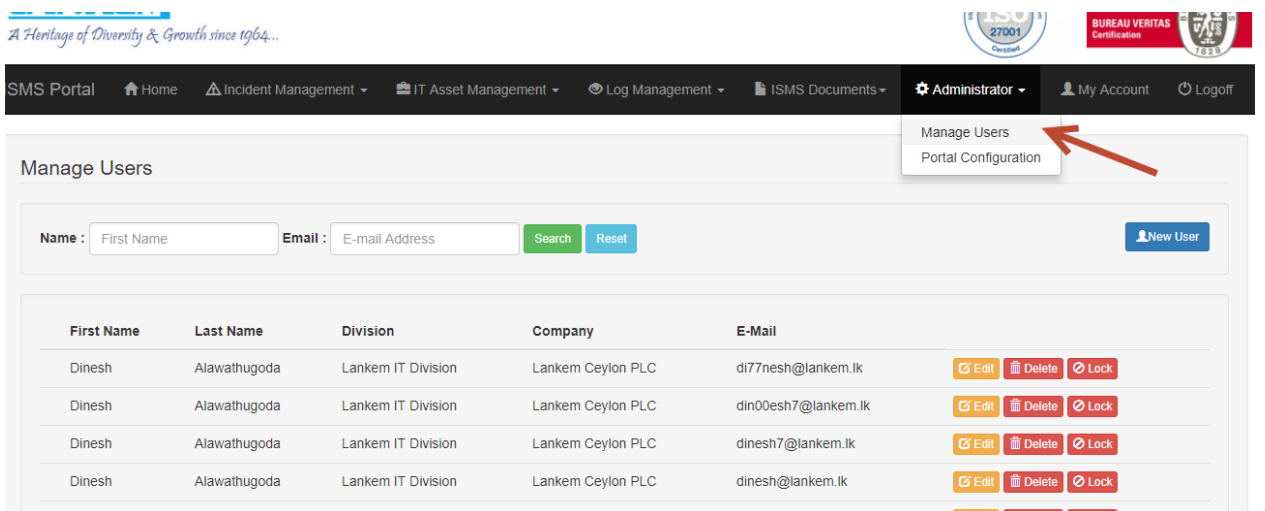


Figure C.19 – User Manage menu path

2. Following activities can be carried out in the user management option.

- Create a New user
- Search for a user
- Edit User Information
- Delete a user
- Lock user from accessing the system.

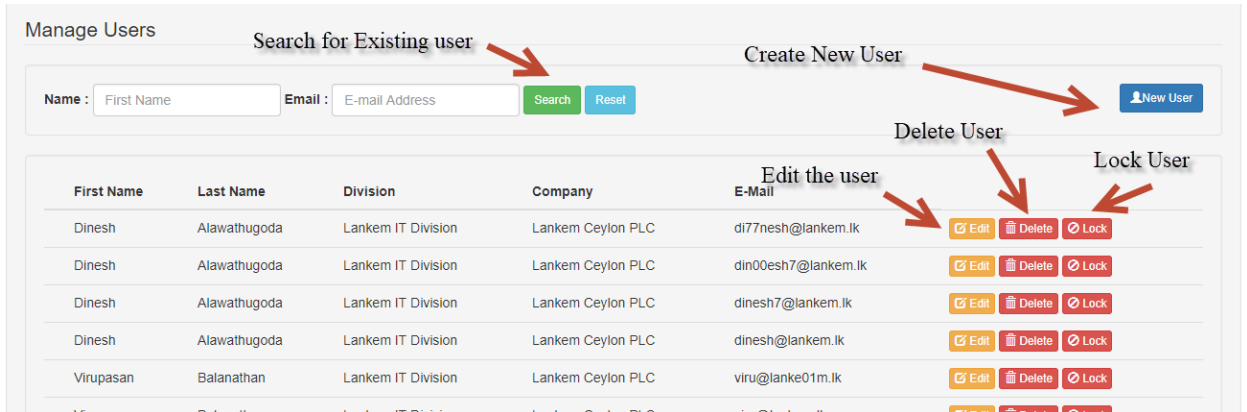


Figure C.20 – User Manage screen

APENDIX D – Testing Result

User Authorization Module

Table D.1 illustrate the test results

| No | Test Case | Expected Result | Result |
|----|---|---|--------|
| 1 | Enter Invalid User Name and Password | Display error message and prevent user from login. | PASS |
| 2 | Enter Valid user name and wrong password | Display error “Invalid password”. | PASS |
| 3 | Login as IT Executive | Only can access the relevant functional Options. | PASS |
| 4 | Login as CISO | Should be able access all the functional options other than admin functions | PASS |
| 5 | Login as System Admin | Should be able to use all the options in the system. | PASS |
| 6 | Login as ISSC member | Should be able to see reports and documents. No access to creation, edit or delete options. | PASS |
| 7 | Logout from the system using logout option. | Current session expired, and user redirected to login page | PASS |

Table D.1 – User management module test Results

Testing Evidence

The screenshot shows a login form with the following elements:

- Title:** Login
- Email address:** A text input field with a red error message below it: "Enter Email address !".
- Password:** A text input field with a red error message below it: "Enter Password !!".
- Remember me (2 Days):** A checkbox that is currently unchecked.
- Buttons:** A blue "Login" button and a red "Reset" button.

Figure D.1 – Login Screen Testing Result

Incident Module

| No | Test Case | Expected Result | Result |
|----|---|---|--------|
| 1 | Incident reporting creation with blank "Incident reporter" field. | Display error message and prevent user creating incident entry. | PASS |
| 2 | Create incident without selecting reported by. | Display error "Select Reported by". | PASS |
| 3 | Create incident without entering incident date. | Display error "Select incident reported date" | PASS |
| 4 | Create incident without entering incident time | Display error "Select Incident Time" | PASS |
| 5 | Create incident without selecting the Classification | Display Error "Select Incident Class" | PASS |

| | | | |
|----|--|---|------|
| 6 | Create incident without selecting the incident type. | Display Error “Select Incident Type” | PASS |
| 7 | Create incident without entering Short Description | Display Error “Enter short description of the incident” | PASS |
| 7 | Create incident without entering description. | Display Error “Enter Description of the incident” | PASS |
| 8 | Create incident without entering Incident reporters email address. | Display Error “Enter the email address” | PASS |
| 9 | Create incident without entering effected business process | Display error “Enter the area or business effected” | PASS |
| 10 | Create incident without selection location of the incident | Display error “Select the location” | PASS |
| 11 | Incident creation attached file more than 3 MB per file | Display error “File size should be less than 3 MB per file” | PASS |
| 12 | Incident creation attach file other than XLS, DOC ,PDF or JPEG | Display error “File should be JPG/JPEG/Doc/Xls/PDF File !! | PASS |
| 13 | Reviewed Incident should not be able to Change, Delete or review again. | System will disable the option for already reviewed incident documents. | PASS |
| 14 | IT Executive should be able to change or delete the incident before review | System display option to change or delete incident | PASS |

Table D.2 – Incident Management Module Test Result

Test Evidence

The screenshot shows a web form for incident management. The fields and their associated error messages are as follows:

- Name of the Incident Reporter (IR):** A text input field with the placeholder "Name of the Incident Reporter (IR)". Below it is a red error message: "Enter Incident Reporter Name !!".
- Reported By:** A dropdown menu with a red error message: "Select Reported By !!".
- Incident Date:** A date picker with the placeholder "Incident Date". Below it is a red error message: "Select Incident Reported Date !!".
- Incident Time:** A time picker with a red error message: "Select Incident Time !!".
- Classification of the Incident:** A dropdown menu with a red error message: "Select Incident Class !!".
- Incident Type:** A dropdown menu with a red error message: "Select Incident Type !!".
- Short Description:** A text input field with the placeholder "Short Description of Incident (Maximum 60 Characters)". Below it is a red error message: "Enter Short Description of the Incident !!".
- Description:** A large text area with a red error message: "Enter Description of the Incident !!".
- Email of IR:** A text input field with the placeholder "Email of IR".

Figure D.2 – Incident management Test Evidence

Asset Management Sub Module

| No | Test Case | Expected Result | Result |
|----|---|---|--------|
| 1 | Create Asset with blank Asset no. | Display Error “Enter an Asset No” | PASS |
| 2 | Create asset with asset number already in the system | Display error “Asset No Already exists in the System” | PASS |
| 3 | Create asset without entering asset description. | Display error “Enter a description” | PASS |
| 4 | Create asset without selecting the Brand of the asset | Display error “Please select a brand”. | PASS |

| | | | |
|----|--|---|------|
| 5 | Create asset without entering the asset model | Display Error “Select enter the model” | PASS |
| 6 | Create asset without entering the serial no. | Display Error “Enter the serial no” | PASS |
| 7 | Create asset without selecting the asset type | Display Error “Select the Asset Type” | PASS |
| 7 | Create asset without selecting the asset category | Display Error “select the asset category” | PASS |
| 8 | Create asset without entering Owner of the asset. | Display Error “Please enter the owners name” | PASS |
| 9 | Create asset without selecting Date of purchase | Display error “Please select the asset procurement date” | PASS |
| 10 | Create asset without selecting supplier | Display error “Select the supplier” | PASS |
| 11 | Create asset without selecting the location | Display error “Select the cost center” | PASS |
| 12 | Create asset without entering the asset purchase value | Display error “enter asset procurement value” | PASS |
| 14 | Users should not be able to change, delete, or transfer asset which are already retired. | System will disable the options for already retired assets. | PASS |
| 15 | IT Executive should be able to change, delete, transfer the assets before retirement. | System display options to change, delete and Transfer | PASS |

Table D.3 – Asset Management Sub Module Test Result

Test Evidence

Create Information Technology Asset

Asset No:
Enter an Asset Number

Asset Description:
Enter an Description

Brand:
Please Select a Brand !!

Asset Model:
Please Enter the Model !!

Asset Serial No:
Enter the Serial no

Asset Type:
Select the Asset Type

Category:
Select the asset Category

Asset Owner:
Please enter the Owners Name

Date of Purchase:
Please select the Asset Procurement Date

Warranty Period in Months:

Figure D.3 - Asset Module Test Evidence

Log Management Sub Module

| No | Test Case | Expected Result | Result |
|----|---|--|--------|
| 1 | Create Monitoring log selecting System and Date | Display Error “Select a System” and “Select the Log Date” | PASS |
| 2 | Create Monitoring log for the combination of system and date for which record is already being created. | Display error “Entry for the Selected System Already exists in the System” | PASS |
| 3 | Create Job Log without selecting the system. | Display error “Please Select the system” | PASS |

| | | | |
|---|---|--|------|
| 4 | Create job log without selecting the date. | Display error “Please select the job monitoring date”. | PASS |
| 5 | Create job log without selecting the Job | Display Error “Please Select the Job name” | PASS |
| 6 | Create job log for a system, date and job combination already exist in the system | Display Error “Record already exist for the combination of System, Date and Job Combination” | PASS |
| 7 | Change or Delete logs which are already reviewed by CISO | System Will disable the Change and delete options | PASS |

Table D.4 – Log management sub module test results

Document Management Sub Module

| No | Test Case | Expected Result | Priority |
|----|--|---|----------|
| 1 | Upload document without selecting the document category. | Display Error “Select a Document Category” | PASS |
| 2 | Upload a document without selecting the document sub Category | Display error “Please select a document sub Category” | PASS |
| 3 | Upload a document without entering the description of the document | Display error “Please enter the meaning full description of the document” | PASS |
| 4 | Upload a document without Document date | Display error “Please select the document date”. | PASS |
| 5 | Upload documents other than file extensions .XLS, XLSX, DOC, DOCX, PDF | Display Error “You can only upload documents with following extensions” | PASS |

Table D.5 – Document Management Module Tests Results.

APENDIX E – Code Listing

Major module codes only listed here for full source code available in the installation CD.

db_config.php (Database Connection Configuration)

```
<?php
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_DATABASE', 'lankemhub');
?>
```

class.user.php (User Management Class)

```
<?php
include (dirname(__FILE__)."/../include/db_config.php");
class User
{
//----- Database connect -----
public $db;
    public function __construct(){
        $this->db = new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD,
DB_DATABASE);
        if(mysqli_connect_errno()) {
            echo "Error: Could not connect to database.";
            exit;
        }
    }
//----- Return User Details -----
public function user_detail($email)
{
    $sql = "SELECT * from users WHERE Email= '$email'";
    $result = mysqli_query($this->db,$sql);
    if ($result)
```

```

{ return mysqli_fetch_array($result);
} }
//----- Get User Roles -----
public function check_user_role($email)
{
$sql = "SELECT * from users WHERE Email= '$email' ";
$result = mysqli_query($this->db,$sql);
if ($result)
{
return $result["Role"];
}
}
//-----Check User and Password-----
public function check_user_password($email, $pwd)
{
$pwd = md5($pwd);
$sql = "SELECT Email from users WHERE Email= '$email' and password = '$pwd'";
$check = $this->db->query($sql) ;
$no_row = $check->num_rows;
//-----Check Login email -----
public function check_login_email($email)
{
$sql = "SELECT Email from users WHERE Email= '$email' and Status = 'A'";
$check = $this->db->query($sql) ;
$no_row = $check->num_rows;
if ($no_row == 1)
{
echo "true";
}
//----- Login process-----
public function check_login($username, $password)
{
$password = md5($password);
$sql1 = "SELECT Email, Role from users WHERE Email= '$username' and password =
$password' and Status = 'A'";

```

```

$result = mysqli_query($this->db,$sql1);
$user_data = mysqli_fetch_array($result);
$no_rows = $result->num_rows;
if ($no_rows == 1)
{
$_SESSION['login'] = true;
$_SESSION['username'] = $user_data['Email'];
$_SESSION['role'] = $user_data['Role'];
return "true";
}else{
return "false";
}}
//----- Update Query -----
public function UpdateQuery($query)
{
    $result = mysqli_query($this->db, $query);
    if($result)
    {return true;
    }
    else
    {return false;
    }
}
//----- Delete User ----->
public function delete_user($email)
{
$sql = "delete from users WHERE Email= '$email'";
$result = mysqli_query($this->db, $sql);
    if($result)
    { return true;}
    Else {
return false; }
}

```



```
// ----- Get Profile Picture For Delete-----
public function get_profile_pic_del($email)
{
$sql2="SELECT * FROM users where Email = '$email'";
$result_set=mysqli_query($this->db, $sql2);
while($row=mysqli_fetch_array($result_set))
{
return $row['Prof_img'];
}
?>
```

class.incident.php (Incident Management Class)

```
<?php
include (dirname(__FILE__)."../include/db_config.php");
class incident
{
//----- Database connect -----
public $db;
public function __construct(){
$this->db = new mysqli(DB_SERVER, DB_USERNAME, DB_PASSWORD,
DB_DATABASE);
if(mysqli_connect_errno()) {
echo "Error: Could not connect to database.";
exit;
}
}
//----- Create Incident Type Drop Down -----
public function create_incidenttype_dropdown($conn)
{
$result = mysqli_query($conn, "SELECT * from incident_type");

while($row = mysqli_fetch_assoc($result))
{
```

```

    echo "<option value= '$row[indtype]'>$row[indtyped]</option>";
  }
}
//----- Create classification Drop Down -----
public function create_incidentclass_dropdown($conn)
{ $result = mysqli_query($conn, "SELECT * from incident_class");
while($row = mysqli_fetch_assoc($result))
  {
    echo "<option value= '$row[indclass]'>$row[indclassd]</option>";
  } }
//----- Create Drop Down For Edit Incident Type-----
public function edit_incidenttype_dropdown($conn)
{
$result = mysqli_query($conn, "SELECT * from incident_type");
if($result)
{
return $result;
}}
//----- Create Drop Down For Edit Incident Class-----
public function edit_incidentclass_dropdown($conn)
{
$result = mysqli_query($conn, "SELECT * from incident_class");
if($result)
{
return $result; }
}
//----- Return User Details -----
public function incident_detail($indno)
{
$sql = "SELECT * from incident WHERE indno= '$indno'";
$result = mysqli_query($this->db,$sql);

if ($result)
{ return mysqli_fetch_array($result); }}

```

```

//-----Run Query -----
public function runQuery($query)
{ $result = mysqli_query($this->db, $query);
  while($row=mysqli_fetch_assoc($result))
    { $resultset[] = $row;
    }
  if(!empty($resultset))
    return $resultset;}

//----- Exeicute Query -----
public function IndQuery($query)
{ if($result = mysqli_query($this->db, $query))
  { return $this->db->insert_id;
  }
  else
  { return '0';
  } }

//----- Return No of Row for Pagination -----
public function numRows($query)
{ $result = mysqli_query($this->db,$query);
  $rowcount = $result->num_rows;
  return $rowcount;
}

//----- Delete Incident ----->
public function delete_incident($indno)
{ $sql = "delete from incident WHERE indno= '$indno' and indrev <> 'Y'";
  $result = mysqli_query($this->db, $sql);
  if($result)
  { return true;
  }else
  {return false;
  }
}
}

```

```
//----- Delete Incident Documents ----->
public function delete_incident_doc($indno)
{   $sql2="SELECT * FROM incident_docs where indno = '$indno'";
$result_set=mysqli_query($this->db, $sql2);
while($row=mysqli_fetch_array($result_set))
{ $fileloc = "incident_docs/".$row['file'];
    if ($fileloc != "")
        {
            unlink($fileloc);
        }
}
$sql = "delete from incident_docs WHERE indno= '$indno'";
$result = mysqli_query($this->db, $sql);
    if($result)
    {
        return true;
    } else {
        return false;
    }
}

//-----Delete Particular Document -----
public function delete_doc($fileno)
{   $sql2="SELECT * FROM incident_docs where fileno = '$fileno'";
$result_set=mysqli_query($this->db, $sql2);
while($row=mysqli_fetch_array($result_set))
{ $fileloc = "incident_docs/".$row['file'];
    if ($fileloc != "")
        {unlink($fileloc); } }
$sql = "delete from incident_docs WHERE fileno= '$fileno'";
$result = mysqli_query($this->db, $sql);

    if($result){
        return true;} else { return false;}
}
```

```
//----- Return Incident Review Status -----
public function get_review_status($indno)
{ $sql3="SELECT * FROM incident where indno = '$indno'";
$result_set=mysqli_query($this->db, $sql3);
    if($result_set){
        return $result_set;}
}
//-----Destruct-----
public function __destruct(){
    mysqli_close($this->db);
}
?>
```

class.utilization.php (Utilization Management Class)

```
<?php
include (dirname(__FILE__)."../include/db_config.php");
class utilization
{
//----- Database connect -----
public $db;
    public function __construct(){
        $this->db = new mysqli(DB_SERVER, DB_USERNAME,
DB_PASSWORD, DB_DATABASE);
        if(mysqli_connect_errno()) {
echo "Error: Could not connect to database.";
            exit; }}
//----- Create System Drop Down -----
public function create_system_dropdown($conn)
{ $result = mysqli_query($conn, "SELECT * from system");

while($row = mysqli_fetch_assoc($result))
    { echo "<option value= '$row[SystemID]'">$row[SystemName]</option>"; } }
```

```

//-----Check Utilization Log-----
public function check_entry($system,$date)
{ $sql = "SELECT * from utilization_log WHERE SystemID= '$system' and `Date` =
'$date'"; $result = mysqli_query($this->db,$sql);
$no_rows = $result->num_rows;
if ($no_rows == 1)
{ echo "true";
}else
{echo "false";
}}
//-----Run Query -----
public function runQuery($query)
{ $result = mysqli_query($this->db, $query);
    while($row=mysqli_fetch_assoc($result))
    { $resultset[] = $row;
    } if(!empty($resultset))
    return $resultset; }
//----- Exeicute Query -----
public function IndQuery($query)
{ if($result = mysqli_query($this->db, $query))
    { return true;
    }else
    { return false;
    } }
//----- Update Query -----
public function UpdateQuery($query)
{ $result = mysqli_query($this->db, $query);
    if($result)
    { return true;
    }else
    {return false;
    } }
//----- Return No of Row for Pagination -----
public function numRows($query)

```

```

{ $result = mysqli_query($this->db,$query);
    $rowcount = $result->num_rows;
    return $rowcount; }

//----- Delete Utilization Log ----->
public function delete_log($system,$date)
{
$sql = "delete from utilization_log WHERE SystemID= '$system' and Date = '$date'";
    $result = mysqli_query($this->db, $sql);
    if($result)
    { return true;
    }else
    { return false;
    }}

//-----Destruct-----
public function __destruct(){
    mysqli_close($this->db);
    }} ?>

```

class.asset.php (Asset Management Class)

```

<?php
include (dirname(__FILE__)."/../include/db_config.php");

class asset
{
//----- Database connect -----
public $db;

    public function __construct(){
        $this->db = new mysqli(DB_SERVER, DB_USERNAME,
DB_PASSWORD, DB_DATABASE);

        if(mysqli_connect_errno() {
            echo "Error:
Could not connect to database.";

```

```

        exit;
    }
}

//----- Print Asset Transfer Log -----
public function print_transfer_log($AsstNo)
{
$query = "SELECT * from asset_transfer INNER JOIN location AS new ON
asset_transfer.AsstLoc = new.locco INNER JOIN location AS old ON
asset_transfer.OAsstLoc = old.locco where AsstNo = '$AsstNo'";
?>
<div class="table-responsive">
    <table class="table">
        <thead>
            <tr>
                <th><strong>Old Owner</strong></th>
                <th><strong>Old Location</strong></th>
                <th><strong>Old Cost Center</strong></th>
                <th><strong>New Owner</strong></th>
                <th><strong>New Location</strong></th>
                <th><strong>New Cost Center</strong></th>
                <th><strong>Transfer On</strong></th>
                <th><strong>Transfer By</strong></th>
            </tr>
        </thead> <tbody>
        <?php
$result = $this->runQuery($query);

if(!empty($result)) {
    foreach($result as $k=>$v)
    {
        if(is_numeric($k)) {
            ?> <tr>
                <td><?php echo $result[$k]["OAsstOwn"]; ?></td>
                <td><?php echo $result[$k]["OAsstLoc"]; ?></td>

```



```

        <td><?php echo $result[$k]["OAsstCocs"]; ?></td>
        <td><?php echo $result[$k]["AsstOwn"]; ?></td>
        <td><?php echo $result[$k]["AsstLoc"]; ?></td>
        <td><?php echo $result[$k]["AsstCocs"]; ?></td>
        <td><?php echo $result[$k]["TransferOn"]; ?></td>
        <td><?php echo $result[$k]["TransferBy"]; ?></td></tr>
    <?php
    <tbody></table> </div>
    <?php }
//----- Create Asset Type Drop Down -----
public function create_asettype_dropdown($conn) {
$result = mysqli_query($conn, "SELECT * from asset_type");
while($row = mysqli_fetch_assoc($result))
    { echo "<option value= '$row[AsstType]'">$row[TypeDesc]</option>"; } }
//-----
public function edit_asettype_dropdown($conn)
{ $result = mysqli_query($conn, "SELECT * from asset_type");
if($result)
{ return $result;
}}
//----- Create Asset Category Drop Down -----
public function create_asetcat_dropdown($conn)
{ $result = mysqli_query($conn, "SELECT * from asset_cat");
while($row = mysqli_fetch_assoc($result))
    { echo "<option value= '$row[AsstCat]'">$row[AsstCatDesc]</option>";
    } }
//-----
public function edit_asetcat_dropdown($conn)
{ $result = mysqli_query($conn, "SELECT * from asset_cat");
if($result)
{ return $result;
}}
//----- Create Cost Center Drop Down -----
public function create_costcenter_dropdown($conn)

```

```

{ $result = mysqli_query($conn, "SELECT * from costcenter");
while($row = mysqli_fetch_assoc($result))
    { echo "<option value= '$row[AsstCocs]'>$row[CocsName]</option>"; } }
//-----

public function edit_costcenter_dropdown($conn)
{ $result = mysqli_query($conn, "SELECT * from costcenter");
if($result)
{ return $result; } }
//----- Create Asset Brand Drop Down -----

public function create_assetbrand_dropdown($conn) {
$result = mysqli_query($conn, "SELECT * from asst_brand");
while($row = mysqli_fetch_assoc($result))
    { echo "<option value= '$row[AsstBrand]'>$row[AsstBrandDesc]</option>";
    } }
//-----

public function edit_assetbrand_dropdown($conn)
{ $result = mysqli_query($conn, "SELECT * from asst_brand");
if($result)
{ return $result; } }
//-----Check asset-----

public function check_asset($asstno)
{ $sql = "SELECT AsstNo from asset WHERE AsstNo= '$asstno'";
$check = $this->db->query($sql) ;
$no_row = $check->num_rows;
if ($no_row == 1)
{ echo "false";
} else
{ echo "true"; } }
//----- Get Asset Details -----

public function asset_detail($asstno) {
$sql = "SELECT * from asset WHERE AsstNo= '$asstno'";
$result = mysqli_query($this->db,$sql);
if ($result)
{ return mysqli_fetch_array($result);} }

```

```

//-----Run Query -----
public function runQuery($query) {
    $result = mysqli_query($this->db, $query);

    while($row=mysqli_fetch_assoc($result))
    { $resultset[] = $row;
    }  if(!empty($resultset))
        return $resultset; }

//----- Exeicute Query -----
public function IndQuery($query)
{
    if($result = mysqli_query($this->db, $query))
    { return true;
    } else
    { return false;
    } }

//----- Update Query -----
public function UpdateQuery($query)
{
    $result = mysqli_query($this->db, $query);
    if($result)
    { return true; }
    else
    { return false;
    }}

//----- Delete User ----->
public function delete_asset($AsstNo)
{
    $sql = "delete from asset WHERE AsstNo= '$AsstNo' and AsstStatus = 'A'";
    $sql1 = "delete from asset_transfer WHERE AsstNo= '$AsstNo'";
    $result = mysqli_query($this->db, $sql);
    if($result)
    {
        mysqli_query($this->db, $sql1);
    }
}

```

```

        return true;
    } else
    { return false;
    }}

//----- Return No of Row for Pagination -----
public function numRows($query)
{ $result = mysqli_query($this->db,$query);
    $rowcount = $result->num_rows;
    return $rowcount; }

//-----Destruct-----
public function __destruct(){
    mysqli_close($this->db);
    }} ?>

```

class.documentmanagement.php (Document Management Class)

```

<?php
include (dirname(__FILE__)."../include/db_config.php")
class document
{ //----- Database connect -----
public $db;
public function __construct(){
    $this->db = new mysqli(DB_SERVER, DB_USERNAME,
DB_PASSWORD, DB_DATABASE);
if(mysqli_connect_errno()) {
    echo "Error: Could not connect to database.";
        exit; }}

//..... Create Category Drop Down -----
public function create_category_dropdown()
{ $result = mysqli_query($this->db, "SELECT * from document_cat");
while($row = mysqli_fetch_assoc($result))
    { echo "<option value= '$row[DocCat]'">$row[CatDesc]</option>"; }}

//..... Create Sub Category Drop Down -----
public function create_subcategory_dropdown()

```

```

{ $result = mysqli_query($this->db, "SELECT * from document_subcat");
while($row = mysqli_fetch_assoc($result))
    { echo "<option value= '$row[DocSubCat]'">$row[SubCatDesc]</option>"; } }
//-----Run Query -----
public function runQuery($query)
{ $result = mysqli_query($this->db, $query);
    while($row=mysqli_fetch_assoc($result))
        { $resultset[] = $row; }
        if(!empty($resultset))
            return $resultset; }
//----- Exeicute Query -----
public function IndQuery($query)
{ if($result = mysqli_query($this->db, $query))
    { return true; } else
    { return false;
    } }
//----- Return No of Row for Pagination -----
public function numRows($query)
{ $result = mysqli_query($this->db,$query);
    $rowcount = $result->num_rows;
    return $rowcount; }
//-----Destruct-----
public function __destruct(){
    mysqli_close($this->db); }
}
?>

```

APENDIX F – Client Certificate



LANKEM CEYLON PLC - 46/56, Nawam Mawatha,
P.O. BOX 918, COLOMBO 02, SRI LANKA.

Telephone : 011 - 5588000, 011 - 7786000 Fax : 011 - 2478796
Company No. PQ 128

2017/11/06

BIT Coordinator,
University of Colombo School of Computing,
External Degree Center,
Colombo - 06.

Dear Sir/Madam,

Letter of Certification

This is to certify that Mr. Balanathan Virupasan (R091573) has successfully designed and developed a Web Based Information Security Management System for Lankem Ceylon PLC. This project was undertaken by him as a partial fulfillment of a requirement for the Bachelor of Information Technology Degree program.

I am pleased to certify that the system developed by Mr. Balanathan Virupasan fulfill the requirements of our Information Security Management documentation and could be used as our Information Security management Documentation System.

This Certification is issued on the request of Mr. Balanathan Virupasan.

Thank you

Yours Faithfully,

Dammika Weerasinghe
Asst. General Manager - IT,
Lankem Ceylon PLC.

LANKEM CEYLON PLC
IT DIVISION
46/56, Nawam Mawatha
Colombo-02
Tel : 011-7786000 Fax : 011-2478796

GLOSSARY

Alpha Testing – the software is tested in a developer controlled environment by the end-users.

Apache - is an open source web server. Mostly for Unix, Linux and Solaris platforms.

Beta Testing – the software is tested at end-user sites which are not controlled by the developers. All errors recorded by the user will be reported to the developer. These errors are corrected and ironed out the remaining problems with the product before it puts on the general release.

Black Box Testing - is a method to test the functionality of an application as opposed to its internal structures or workings.

Bootstrap - Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.

CSS (Cascading Style Sheet) - is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a mark-up language.

Database - is an organized collection of data for one or more purposes, usually in digital form. **Design pattern** - is a way of reusing abstract knowledge about a problem and its solution. It is simply a description of the problem and the essence of its solution.

Firewall - is a solution that is used to enforce security policies. This can be a Hardware or Software.

Graphical User Interface - is a type of user interface that allows users to interact with electronic devices with images rather than text commands.

HTTPS (Hypertext Transfer Protocol Secure) - is a combination of the Hypertext Transfer Protocol (HTTP) with SSL/TLS protocol to provide encrypted communication and secure identification of a network web server.

Internet - is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide.

Intranet - is a computer network that uses Internet Protocol technology to share.

ISO/IEC 27001 – is an information security standard that was published in September 2013 It supersedes ISO/IEC 27001:2005 and is published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) under the joint ISO and IEC subcommittee, ISO/IEC JTC 1/SC 27. It is a specification for an information security management system (ISMS). Organizations that meet the standard may be certified compliant by an independent and accredited certification body on successful completion of a formal compliance audit.

information, operational systems, or computing services within an organization.

JavaScript - is a prototype-based, object-oriented client-side scripting language It is also considered as a functional programming language. JQuery - is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML.

JQUERY - jQuery is a lightweight, "write less, do more", JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code. jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

Local host – This is the default name describing the local computer address

MVC - Model–view–controller (MVC) is a software architecture pattern which separates the representation of information from the user's interaction with it. The model consists of application data, business rules, logic, and functions.

Structured Query Language - is a database computer declarative language designed for managing data in relational database management systems (RDBMS). **Unified Modeling Language (UML)** - is a standardized general-purpose modeling language in the field of object-oriented engineering. This includes a set of graphic notation techniques to create visual models of object-oriented software-intensive systems.

Validation - Checking whether the system satisfies user requirements. **Verification** - Checking whether the system satisfies system specification.

Web Browser - is software which allows the user to access Internet or view HTML.

White Box Testing - is a testing method which focuses specifically on testing internal knowledge and the structure of the software.

WWW (World Wide Web) - is a system of interlinked hypertext documents accessed via the Internet.

XAMPP - is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes,

INDEX

A

Acceptance... 58,65
Analysis...4,5,13,20,30,69,78

B

Business...1,5,21,39,40,46,61
Bootstrap...32,42,43,105,155
Balckbox

C

Class Diagram...24,31,79
Client ...4,13,23,24,32,41,43
Controller...39,40,45,46,48,49
CSS...32,105,110,155

D

Database...4,20,31,32,40,43,44
Design pattern...39,45,46,156
Drawbacks...iv,ii

E

Existing system...4,5,10,11
Evaluation...4,55,65,67

F

Functional Requirements
13,14,15,67,69

G

General... 40,58,68,155,157
GUI... 24,123

H

Hardware.. 5,41,42,59,70,155
HTML.. 32,40,44,105,107,156

I

Include.. 3,34,44,47,48,68,106
Interactive Internet... 23,156,157
Implementation... 20,21,41,58
ISMS... 11,29,38,68,71,93,156
ISO... 1,3,9,10,12,16

J

JQUERY... 32,43,156
Javascript... 32,44,137,155,156

L

Language... 5,11,24,41,68,155
Levels... 55,56,57,65

M

Manual System... 6,10
Methodologies... ii, 4
MVC... 39,40,45,69

N

Network... 45,72,156
Non-Functional 15,16,67,69

O

Objective... 3,5,65,67
Open Source.. 6,19,23,41,155
Operating System.. 5,13,42,45

P

Password..25,47,60,81,103
Program... 44,56
Process... 1,3,4,12,16,19,20

Q

Quality... 44
Questionnaire... 13,65,66

R

RUP... 158
Reuse... 58
Requirements... 3,4,10,13,77

S

Security... 1,3,6,30,67,78,155
Server... 2,43,70,92,157
Solution... 5,12,42,155
System Design... 15,22

T

Tables... 44
Test Plan... 58
Test Result... 96,98,102
Tools... 4,18,41

U

UML... 20,24
Use-Case... 10,21,25,74
Update... 6,11,44,78,117
Unit Test... 56,58

V

Validate... 105,137,140
View... 11,14,36,39

W

Web Based... 3,6,12,23,45
White Box... 56,157

X

XAMPP... 41,59,70,157
XML... 40