



# **WEB BASED IT ASSET MANAGEMENT SYSTEM FOR DIALOG AXIATA PLC**

**G. D. Achalan**

**Index Number: 1102702**

**BIT Registration Number: R1102702**

**Supervisor(s): Mr. Ragukumar S.T**

**November 2017**




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

# DECLARATION

## 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 interlibrary loans, and for the title and abstract to be made available to outside organizations."

Signature of Candidate:  .....

Date: 04/11/2017

Name of Candidate: G.D. Achalan

Countersigned by:

Signature of Supervisor/Advisor:  .....

Date: 04/11/2017

Name of Supervisor/Advisor: Mr.Ragukumar S.T

# ABSTRACT

Dialog Axiata PLC is Sri Lanka's largest telecommunications service provider with the country's largest mobile operator of 11.8 million subscribers and commands a 50% of the Sri Lanka mobile market. The company has over 3500 employees in island wide, across 12 main branches, 20 regional engineering offices, over 50 customer field support engineer offices. All IT asset data stored in spread sheets and IT asset management team is doing a manual processing to carry out operations such as asset issuing, maintenance, repair, clearance etc. This developing system will help to increase the efficiency of their work, by reducing paper work, spread sheet processing.

This system will provide many facilities such as asset release, asset transfer, asset maintenance, asset clearance, maintenance and warranty, asset disposal, asset clearance, user management, report generation and documents management etc.

Rational Unified Process (RUP) framework has been selected to the IT Asset Management System.

In the development environment, PHP has been selected as the programming language and the MySQL server database have been selected as the database to store the data of the system and NetBeans IDE (Integrated Development Environment) has been selected to implement the system. At the end of this development process, efficient, user friendly and effective system will be delivered to the company.

# ACKNOWLEDGEMENT

I would like to take this opportunity to express my greatest gratitude and deep regards to all those who helped me throughout my project and have been instrumental in the successful completion of this long venture. First and foremost, I would like to express my very deepest appreciation to my supervisor Mr. Ragukumar Steven Thanganathan for the guidance, encouragement, ceaseless cooperation and useful critiques provided throughout the course of this project in making this project a reality. His constant supervision and mentoring despite his busy schedule is greatly appreciated.

I would also like to offer my special heartfelt thanks to the management of Dialog Axiata PLC. I am particularly grateful for the staff for their support and being responsive in providing the necessary information. I offer my sincere appreciation for the learning opportunities provided by Esoft Metro Campus and special thanks of gratitude are extended to the committed and dedicated lecture panel for providing the necessary coaching and for their extraordinary support.

A special word of thanks goes to all my friends and colleagues who helped me in completing the project by exchanging various ideas, thoughts and suggestions and for the much-needed motivation, support and inspiration they extended. Last but not least, most importantly, none of this could have happened without my parents who were the driving forces to complete this project and whose perseverance kept me going on and made me to strive towards my goal and crown all efforts with success.

# TABLE OF CONTENTS

DECLARATION .....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT .....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES .....	vii
LIST OF TABLES .....	ix
LIST OF ACRONYMS .....	xi
CHAPTER 01 – INTRODUCTION .....	1
1.1 INTRODUCTION.....	1
1.2 MOTIVATION OF THE PROJECT.....	1
1.3 SCOPE OF THE PROPOSED PROJECT .....	2
1.4 OBJECTIVES OF THE SYSTEM.....	3
CHAPTER 02 – ANALYSIS.....	4
2.1 INTRODUCTION.....	4
2.2 ANALYZING THE CURRENT MANUAL SYSTEM.....	4
2.3 EXISTING SIMILAR SYSTEMS .....	5
2.4 FUNCTIONAL REQUIREMENTS .....	8
2.5 NON-FUNCTIONAL REQUIREMENTS .....	10
2.6 METHODOLOGY FOR THE PROPOSED SYSTEM .....	11
CHAPTER 03–DESIGN.....	12
3.1 INTRODUCTION.....	12
3.2 ALTERNATE SOLUTIONS EVALUATION .....	12
3.3 DESIGN DIAGRAMS .....	14
3.4 INTERFACE DESIGN .....	20
CHAPTER 04 – IMPLEMENTATION.....	24
4.1 INTRODUCTION.....	24
4.2 IMPLEMENTATION ENVIRONMENT.....	24
4.3 CODE AND MODULE STRUCTURE.....	26
4.4 REUSED COMPONENTS .....	39
CHAPTER 05 – EVALUATION .....	40
5.1 INTRODUCTION.....	40
5.2 TEST PLAN.....	40
5.3 TEST RESULTS .....	50

5.4	USER EVALUATION.....	53
CHAPTER 06 – CONCLUSION .....		54
6.1	INTRODUCTION.....	54
6.2	CRITICAL ASSESSMENT OF THE PROJECT .....	54
6.3	FUTURE ENHANCEMENT .....	55
REFERENCES .....		56
APPENDIX A - SYSTEM DOCUMENTATION.....		57
APPENDIX B - DESIGN DOCUMENTATION .....		59
APPENDIX C - USER DOCUMENTATION .....		67
APPENDIX D - MANAGEMENT REPORTS .....		71
APPENDIX E - TEST RESULTS .....		76
APPENDIX F - CODE LISTING.....		81
APPENDIX G - CLIENT CERTIFICATE.....		86

# LIST OF FIGURES

Figure 2.1 Use case diagram of the current manual system .....	4
Figure 2.2 logo of ManageEngine AssetExplorer.....	6
Figure 2.3 Home Page of ManageEngine AssetExplorer .....	6
Figure 2.4 logo of ManageEngine AssetExplorer.....	7
Figure 3.1 High level use case diagram .....	14
Figure 3.2 Use case diagram for Asset Release Management .....	15
Figure 3.3 Use case diagram for Asset Procurement Request Management .....	15
Figure 3.4 Class diagram of the system .....	16
Figure 3.5 Asset Release Management Sequence Diagram.....	17
Figure 3.6 Asset Procurement Request Management Sequence Diagram.....	18
Figure 3.7 Asset Release Activity Diagram.....	19
Figure 3.8 Asset Procurement Request Activity Diagram.....	19
Figure 3.9 Home Page Interface Design .....	21
Figure 3.10 Login Page with validation Interface Design .....	21
Figure 3.11 Data Tables Interface Design .....	22
Figure 3.12 Form Interface Design .....	22
Figure 3.13 Sample Reports Interface Design .....	23
Figure 3.14 Charts – Top 10 fast moving .....	23
Figure 4.1 MVC Interaction Environment.....	26
Figure 4.2 System Model Structure .....	26
Figure 4.3 Model Structure .....	27
Figure B.1 Login Sequence Diagram.....	64
Figure B.3 Asset Clearance Activity Diagram .....	66
Figure C.1 Login Page .....	67
Figure C.2 Dashboard Page .....	68
Figure C.3 Module Functional Page .....	69
Figure C.4 Required Fields .....	69
Figure C.5 Common Guidelines .....	70
Figure D.1 Asset Meter.....	71
Figure D.2 Top 10 Fast moving asset for last 30 days.....	72
Figure D.3 Asset Request throughout the year. ....	72

Figure D.4 Top 10 Top 10 Multiple Asset Owners ..... 73  
Figure D.5 Detailed Report for Asset requests for given period ..... 74  
Figure D.6 Detailed Report for Asset Allocation for given period..... 74  
Figure D.7 Detailed Report for Asset Maintenance Report for given period ..... 75  
Figure D.8 Detailed Report for Asset Clearance Report for given period..... 75



# LIST OF TABLES

Table 2.1 Pros and Cons for ManageEngine Asset Explorer.....	5
Table 2.2 Pros and Cons for BMC Track-It! .....	7
Table 4.1 Implementation Environment .....	24
Table 4.2 Technologies .....	25
Table 5.1 Test Cases for Deployment Management Module .....	41
Table 5.2 Test Cases for Maintenance Management Module.....	42
Table 5.3 Test Cases for Warranty Management Module .....	43
Table 5.4 Test Cases for Tracking Management Module.....	43
Table 5.5 Test Cases for Procurement Management Module.....	45
Table 5.6 Test Cases for Transfer Management Module.....	45
Table 5.7 Test Cases for Clearance Management Module .....	46
Table 5.8 Test Cases for User Management Module.....	47
Table 5.9 Test Cases for Report Management Module .....	48
Table 5.10 Test Cases for Document Management Module.....	48
Table 5.11 Test Cases for Common Functions .....	49
Table 5.12 Test Results for View a User Test Case .....	50
Table 5.13 Test Results for Add a User Test Case .....	51
Table 5.14 Test Results for Validation Test Case.....	52
Table 5.15 Ratings .....	53
Table 5.16 User Acceptance Test Results.....	53
Table A.1 Hardware Requirements.....	57
Table A.2 Software Requirements .....	57
Table B.1 Use-Case Description for Asset Allocation .....	59
Table B.2 Use-Case Description for Create New Asset Category.....	59
Table B.3 Use-Case Description for New Asset Procurement Request .....	60
Table B.4 Use-Case Description for Approve Asset Procurement Request.....	60
Table B.5 Use-Case Description for Reject Asset Procurement Request.....	61
Table B.6 Use-Case Description for Asset Transfer.....	61
Table B.7 Use-Case Description for Asset Clearance .....	62
Table B.8 Use-Case Description for Asset Clearance .....	62
Table B.9 Use-Case Description for Manager Users.....	63

Table B.10 Use-Case Description for Generate Reports .....	63
Table B.11 Use-Case Description for Login.....	64
Table E.1 Test Results for Deployment Management Module.....	77
Table E.2 Test Results for Maintenance Management Module.....	77
Table E.3 Test Results for Procurement Management Module.....	78
Table E.4 Test Results for Common Functions .....	80

# LIST OF ACRONYMS

AJAX	-	Asynchronous JavaScript and XML
BIT	-	Bachelor of Information Technology
CD	-	Compact Disc
CRUD	-	Create, read, update and delete
GNU GPL	-	GNU General Public License
HR	-	Human Resource
HTTP	-	Hypertext Transfer Protocol
IDE	-	Integrated Development Environment
IT	-	Information Technology
ITAMS	-	IT Asset Management System
JAD	-	Joint Application Development
MIT	-	Massachusetts Institute of Technology
MVC	-	Model View Controller
LDAP	-	Lightweight Directory Access Protocol
PLC	-	Public Limited Company
PO	-	Purchase Order
RAD	-	Rapid Application Development
ROM	-	Read-only Memory
RUP	-	Rational Unified Process
SCM	-	Supply Chain Management
SDLC	-	Software Development Life Cycle
UI	-	User Interface
URL	-	Universal Resource Locator
XML	-	Extensible Markup Language

# **CHAPTER 01 – INTRODUCTION**

## **1.1 INTRODUCTION**

Dialog Axiata PLC is Sri Lanka's largest telecommunications service provider with the country's largest mobile operator of 11.8 million subscribers and commands a 50% of the Sri Lanka mobile market. The company has over 3500 employees in island wide, across 12 main branches, 20 regional engineering offices, over 50 customer field support engineer offices.

All IT asset data stored in spread sheets and IT asset management team is doing a manual processing to carry out operations such as asset issuing, maintenance, repair, clearance etc. This developing system will help to increase the efficiency of their work, by reducing paper work, spread sheet processing. Furthermore, this system will enable decision making quickly and easily without any obstacle.

## **1.2 MOTIVATION OF THE PROJECT**

Each employee is entitled to have specific IT assets such as desktops, laptops, monitors, printers, scanners etc. and there is a separate team of 8 staff to manage IT assets.

Currently IT asset management team is consuming a lot of time using spreadsheets and following IT asset management related processes such as asset issuing, maintenance, repair, tracking, clearance etc.

Most of current IT asset data are stored in files, which is crucial to be digitized.

This is an opportunity for me to apply the knowledge gained in BIT degree program to my workplace by developing a web based IT Asset Management System (ITAMS) for Dialog Axiata PLC.

### **1.3 SCOPE OF THE PROPOSED PROJECT**

Web Based IT Asset Management System for Dialog Axiata PLC will include following modules.

- **Asset Deploy Management**  
Provide a module for basic asset management functions, such as issuing assets, view assets with a customized search, asset category and model review.
- **Asset Maintenance Management**  
All asset repair information will be captured by this module. IT technicians will be able to manage your job tasks such as asset repairing, replacing, disposal of unusable IT assets.
- **Asset Warranty Management**  
Warranties can often lower the cost of repair and maintenance of your assets. This module will enable track warranty efficiently.
- **Asset Tracking Management**  
Assets will be using by multiple end users and this system will allow to track usage information.
- **Asset Procurement Request Management**  
At Dialog, all procurements will be handled by SCM and divisional head needs to approve every request made by IT Asset Management team. This system will enable to provide procurement requirement to SCM and manage vendors.
- **Asset Transfer Management**  
Asset transfers can occur due to divisional changes and job responsibility changes. This module will manage asset transfers easily.
- **Asset Clearance Management**  
When an employee resigns the company, he/she must hand over his/her assets to IT Asset Management team and asset management team will verify if the asset is on working condition or damaged. If damaged, there is a procedure to follow and system should give information needed to IT Asset Management Team and HR.
- **User Management**  
This is a role based system and authority should be managed from this module.

- Document Management

This module will include report generation and document management. There are several process and procedure documents that should be reviewed periodically for audits.

## **1.4 OBJECTIVES OF THE SYSTEM**

Deliver an automated system that allows to allocate, track, maintenance, clearance etc.

- The system should be able to manage all IT Assets effectively and efficiently.
- Data regarding IT Assets should be properly organized and able to be received in a short period.
- The system should reduce time and manual effort related to deploying assets to employees, maintenance, tracking, clearance and procurement.
- Assets should be tagged to each employee and should manage asset clearance when an employee resigns the organization.
- Assist the management to get insights about IT Assets procurements.
- The data of the organization should be secure and should not allow unauthorized access to the system.
- The system and database should be backed up properly and should maintain high availability.

# CHAPTER 02 – ANALYSIS

## 2.1 INTRODUCTION

“The analysis phase involves gathering requirements for the system and it is one of the main phases in the software development life cycle. System analysts will focus on what the system will do in an effort that views all stakeholders, as viable sources of information and help to get an overall image of the system producing a high-level description of the system. Main objectives of this phase are what services system should provide, required performance of the system. Before analyzing the system, requirements gathering should be complete by using the fact-finding techniques, such as interviews, observations, sample documentations etc.” [WWW1]

## 2.2 ANALYZING THE CURRENT MANUAL SYSTEM

During analysis phase, below use cases were identified and all the functions were currently doing manually using spreadsheets.

Figure 2.1 shows the use case diagram of the current manual system.

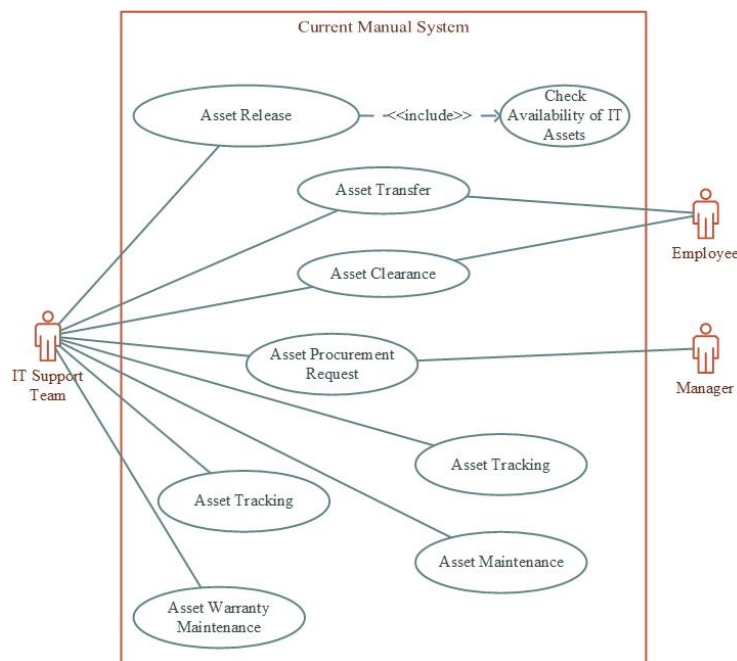


Figure 2.1 Use case diagram of the current manual system

## 2.3 EXISTING SIMILAR SYSTEMS

When considering the IT asset management systems, there are many propriety software available. Most of the systems don't include any data handling or processing systems as per client's requirement. As those are propriety software, the organizations business process need to change as per required standard. Below are some examples for IT asset management systems.

- **ManageEngine Asset Explorer**

ManageEngine Asset Explorer is a web-based IT Asset Management software that helps you monitor and manage assets in your network from Planning phase to Disposal phase. AssetExplorer provides you with a number of ways to ensure discovery of all the assets in your network. Below are main functionalities of ManageEngine Asset Explorer. [WWW2]

- Discover all the assets in your network.
- Manage and Monitor software and hardware assets.
- Manage the complete IT Asset lifecycle.
- Track Purchase Orders and Contracts.
- Know the total cost of ownership of an asset.

Table 2.1 shows the pros and cons for ManageEngine AssetExplorer.

Pros	Cons
Hardware changes automatically added.	User interfaces are not user friendly.
Understand asset total cost of ownership.	Difficult implementation.
	The business process should change according to the system.

*Table 2.1 Pros and Cons for ManageEngine Asset Explorer*



Figure 2.2 shows the logo of ManageEngine AssetExplorer.



Figure 2.2 logo of ManageEngine AssetExplorer

Figure 2.3 shows the Home Page of ManageEngine AssetExplorer.



Figure 2.3 Home Page of ManageEngine AssetExplorer

- **BMC Track-It!**

Track-It! Inventory is just one part of the fully integrated Track-It! solution. Each Track-It! Module helps to automate, streamline and organize issues specific to IT challenges, to help you reduce costs and improve service levels. Below are main functionalities of BMC Track-it Inventory management solution. [WWW3]

- Automating your inventory collection process.
- Manage assets and reporting on assets.
- Track asset changes.

Table 2.2 shows the pros and cons for BMC Track-It!

Pros	Cons
Track asset depreciation over time	User interfaces are not user friendly.
Mobile-friendly	Difficult implementation.
The business process should change according to the system	The cost is higher compared to other similar systems

*Table 2.2 Pros and Cons for BMC Track-It!*

Figure 2.4 shows the logo of ManageEngine AssetExplorer.



*Figure 2.4 logo of ManageEngine AssetExplorer*

All these proprietary software, are based on a standard operating procedure but when it comes to client's requirements we need to redesign the business process. The cost of this proprietary software is high and not customizable. So, that the best option is to develop the system to match with the client's requirements.

## **2.4 FUNCTIONAL REQUIREMENTS**

What the system is supposed to achieve, is known as functional requirements. Functional requirement can be a calculation, technical detail, etc. Following are the functional requirements that have been identified through the system analysis.

### **2.4.1 Asset Deploy Module**

- Asset administrator can be able to release assets by performing view, add update functions.
- Asset administrator can be able maintain asset catalogue by inserting new categories, deleting old categories, updating new details about assets items.

### **2.4.2 Asset Maintenance Module**

- Asset administrator can be able to view, add and modify all asset repair details.
- Asset administrator can be able to view repair history.

### **2.4.3 Asset Procurement Management**

- Most of the asset procurements are doing by SCM. IT asset management team will request the asset details that need to be purchased by the SCM. This module will enable the user to create, modify and delete the asset requests.
- Manager will be able to approve the purchase request, created by Asset administrator.
- After SCM processed the procurement this system will allows the user to insert items to the asset database and can be release using deploying module.
- Minor assets such as accessories are purchased by using a rate card provided by the SCM. Those assets can also insert in to the asset database.

### **2.4.4 Asset Tracking Management**

- Asset administrator can be able to track asset assignment details such as asset usage history.
- Manager can be able to get insights about asset procurement by analyzing older assets.

### **2.4.5 Asset Warranty Management**

- Asset administrator will receive alerts on recent asset warranty expiry items.
- Asset administrator can view and renew warranty by modifying asset warranty details.

#### **2.4.6 Asset Transfer Management**

- Asset administrator can be able to transfer ownership of an asset from employee to another employee when employee is changing their job role or division.

#### **2.4.7 Asset Clearance Management**

- Asset administrator can be able to revoke ownership of an asset of an employee when he/she is leaving the organization.
- Disposal of the asset will be checked at this stage and if the asset is reusable it can be release to another employee.

#### **2.4.8 Document Management**

- Managers can be able to manage documents related to asset management such as process document by reviewing, updating and deleting.

#### **2.4.9 Reports Management**

- Managers can be able to get customized reports on asset management modules.

#### **2.4.10 User Management**

- Asset administrator can be able to view user accounts.
- Manager can be able to view, update, activate or deactivate user accounts.

#### **2.4.11 Notification Management**

- Asset administrator will receive notifications on low inventory items.

## **2.5 NON-FUNCTIONAL REQUIREMENTS**

Non-Functional requirements are also important facts which we need to consider when developing the system. Non-functional requirements describe the system properties and constraints of the system. Those requirements are applied to the whole system, not only for the individual parts of the system. If we do not consider about non-functional requirements system can be useless. Basically, Non-functional requirements describe how the system works.

### **2.5.1 Reliability**

The users should trust the system and the data generate by the system. There's a requirement that data created in the system will be retained for a number of years without data being changed by the system.

### **2.5.2 Availability**

Availability indicates when a system is operational as well as how reliable it is during operational periods. System should maintain availability with minimum downtime. The system should available at any time. Scheduled downtime is acceptable.

### **2.5.3 Security**

System should be achieving the security requirements. Because most of the important details user details, asset details, procurement related details are stored in this system. If that information is misplaced, then it will be a huge problem. Strict authentication will be used and scheduled backups will be maintained.

### **2.5.4 Usability**

Usability requirement has been achieved by using various techniques such as easy menu navigation, searching options, attractive interfaces and use of matching colors.

### **2.5.5 Accuracy**

Accuracy is an also important nonfunctional requirement which we need to consider. If we do not consider about the accuracy, database problems can be occurred. By applying validation techniques to the user input data accuracy has been achieved in this system.

## **2.6 METHODOLOGY FOR THE PROPOSED SYSTEM**

RUP (Rational Unified process) has been selected for the proposed system. RUP is not a single fixed process, but rather an adaptable process framework. RUP can be used when client's requirements are not clear.

In this project, initially the system requirements were not clear, so requirements may likely to be changed. Iterative and incremental development created as a response to inefficiencies and problems found in the waterfall model, so we should be able to avoid problems which might cause using waterfall model. Therefore, RUP is the most appropriate SDLC (Software Development Life Cycle) methodology. The RUP has determined a project life-cycle consisting of four phases such as Inception, Elaboration, Construction and Transition. Brief description is given below.

- **Inception Phase:** In this phase business case is defined and the procedural approach that is to be taken to implement the project.
- **Elaboration Phase:** In this phase create the project plan and construct an architectural baseline that implements a working application with limited functionality.
- **Construction Phase:** In this phase finish development based on the baseline architecture
- **Transition Phase:** In this phase supply the system to its end users.

# CHAPTER 03–DESIGN

## 3.1 INTRODUCTION

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. [WWW4]

System designers may use various system design approaches. Such as modern structured design, information engineering, prototyping, JAD, RAD and object-oriented design. Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a software problem and it is the widely-used approach in modern software designing. The main advantage of using an object-oriented design is that it can be reused and coding and understanding the coding will be easier and will minimize errors. [WWW5]

## 3.2 ALTERNATE SOLUTIONS EVALUATION

There are different strategies may involve the way of development, the development platform, choice of system software used to system design. Below are some of the alternative solutions available.

- **Packaged Software**

As mentioned in 2.3 Existing Similar Systems, ManageEngine AssetExplorer, BMC Track-It! are some of packaged software will install and configured on a web server. Software package will include similar features and is bundled together as set of components.

- **Cloud Solutions**

Software will be web-based; therefore, will not be installed locally on a server. Users able to use only client with any platform to use the system.

In this project, there are some drawbacks in alternative solutions as customer requirement is not satisfying due to many aspects. Below are some major drawbacks with regards to alternative solutions.

- As the business process is complex, it is difficult to implement a package software in the operational environment.
- Required changes that should match the current process which cannot be enclosed by the alternative solutions.
- The cost is expensive compared to similar systems.
- The user interfaces are not user friendly.

To overcome above drawbacks, below strategies were involved in the proposed IT asset management system.

- By using a standalone application, the data will not be centralized, therefore create a web-based system and use client server architecture. Hence, there is no need to install into any clients' PCs and data will be centralized.
- Object-oriented design concepts will useful to plan coding in order to have a better flowing program and can be reused.
- The system will use open-source components such as JQuery, Bootstrap.
- The system will host on a Linux web server which will reduce the cost.
- User friendly flat colors, fonts, icons will be used to enhance the usability of the system.



### 3.3 DESIGN DIAGRAMS

#### 3.3.1 High Level Use Case Diagram for the IT Asset Management

Figure 3.1 Shows the high-level use case diagram of the system

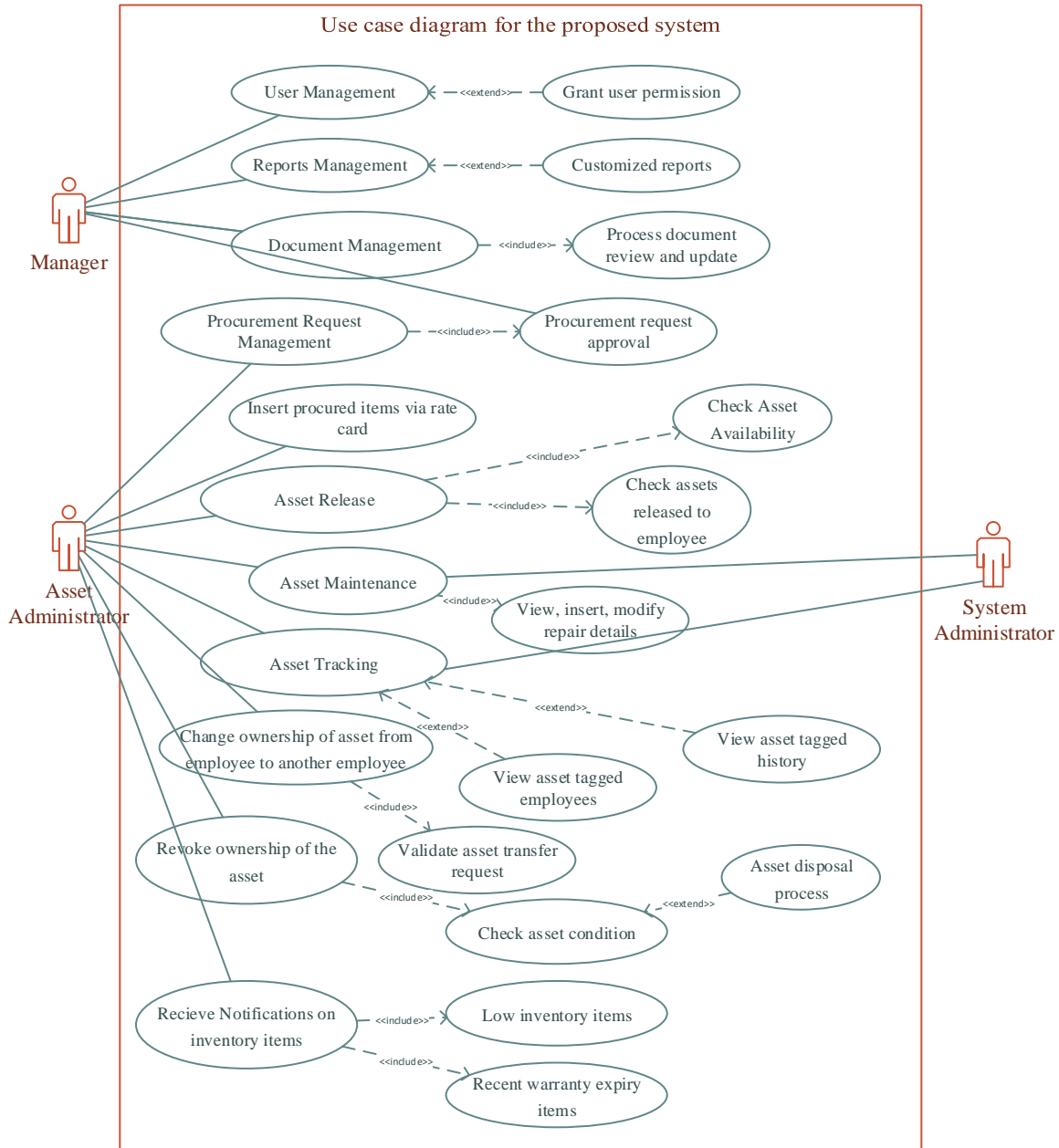


Figure 3.1 High level use case diagram

### 3.3.2 Use Case Diagram for Asset Release Management

After checking the availability of an asset and verify already given assets, Asset Administrator can be able to release a new/existing asset to an employee.

Figure 3.2 shows the use case diagram for Asset Release Management.

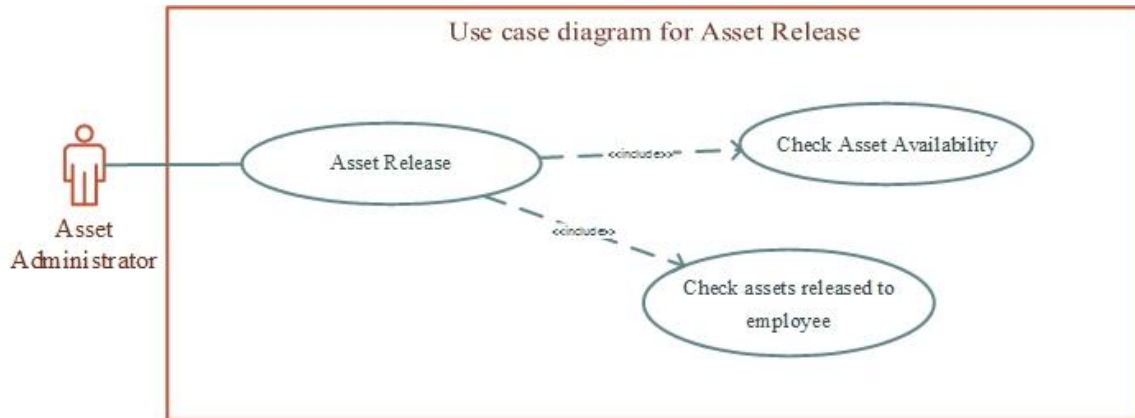


Figure 3.2 Use case diagram for Asset Release Management

### 3.3.3 Use Case Diagram for Asset Release Management

Asset Administrator will create a request and it will send to Manager for approvals, once approval granted, the request will send to supply chain management. Also, minor assets such as accessories will be procured via a rate card, which can be insert to the system.

Figure 3.3 shows the use case diagram for Asset Procurement Request Management.

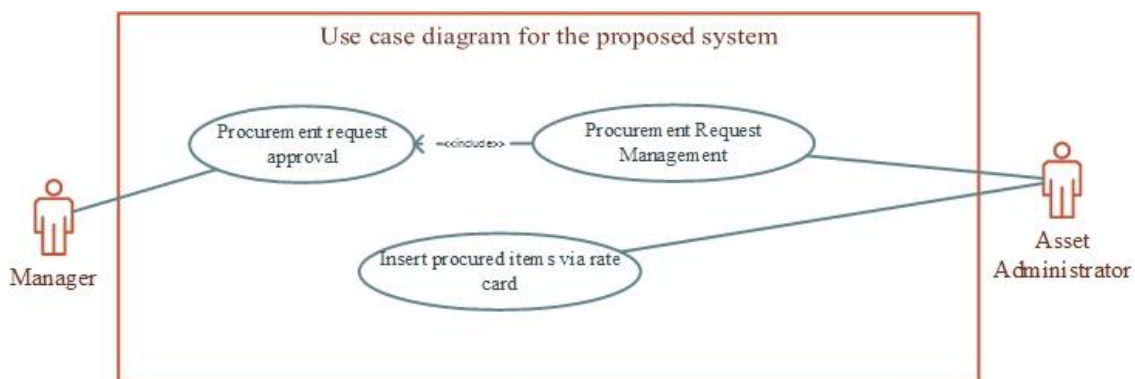


Figure 3.3 Use case diagram for Asset Procurement Request Management

### 3.3.4 Class Diagram for the IT Asset Management System

Figure 3.4 shows the Class diagram of the system.

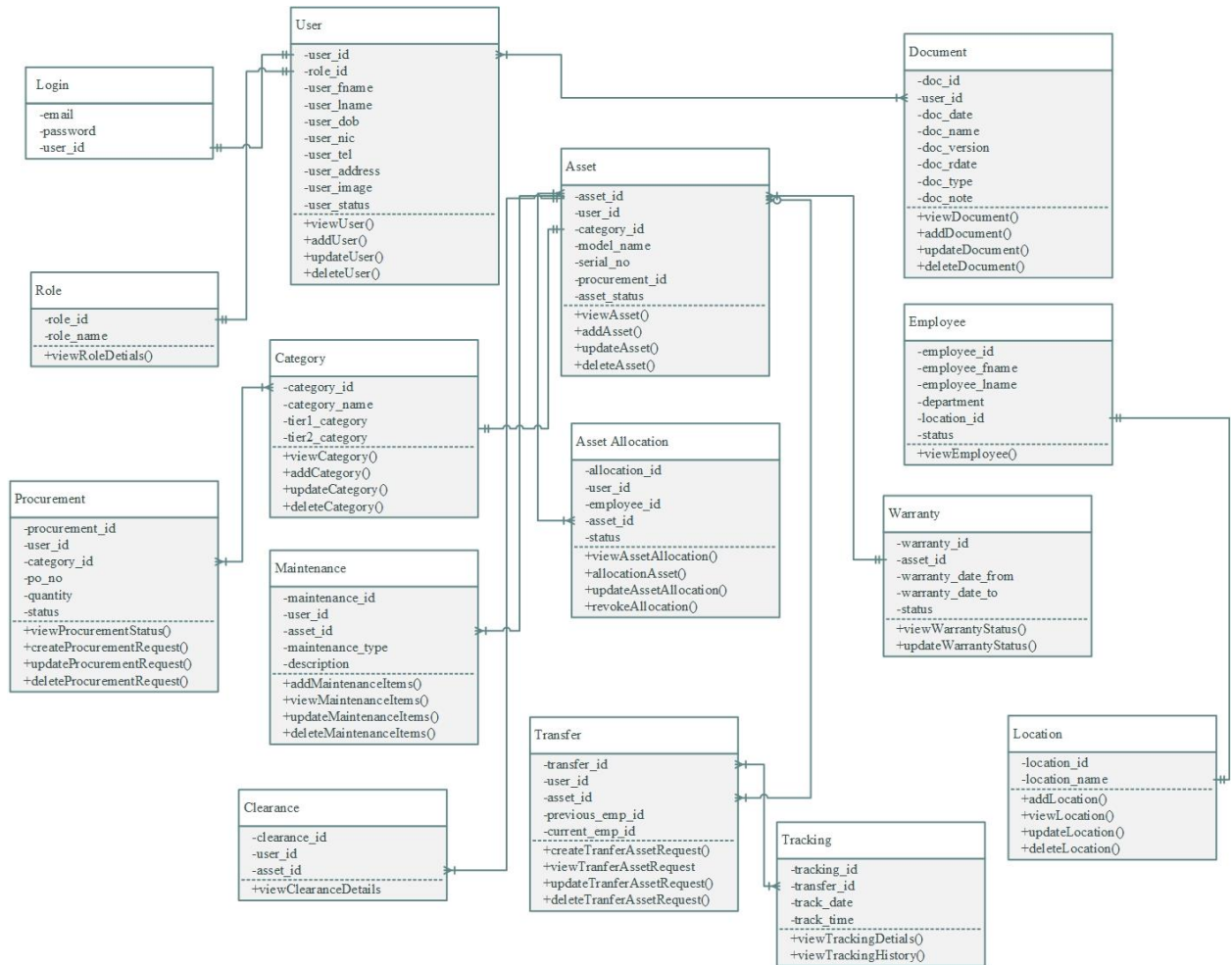


Figure 3.4 Class diagram of the system

### 3.3.5 Sequence Diagram for the Asset Release Management Module

The sequence diagram which relates to the asset release management is displayed by figure 3.5.

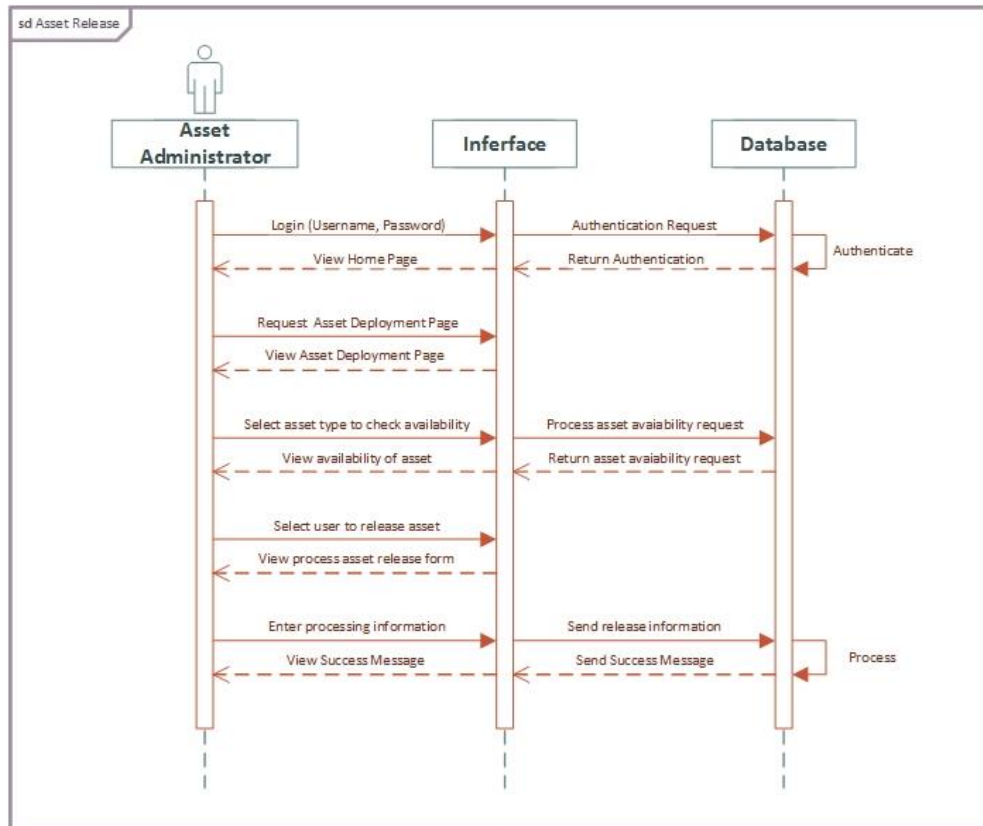


Figure 3.5 Asset Release Management Sequence Diagram

### 3.3.6 Sequence Diagram for the Asset Procurement Request Module

The sequence diagram which relates to the asset procurement request management is displayed by figure 3.6.

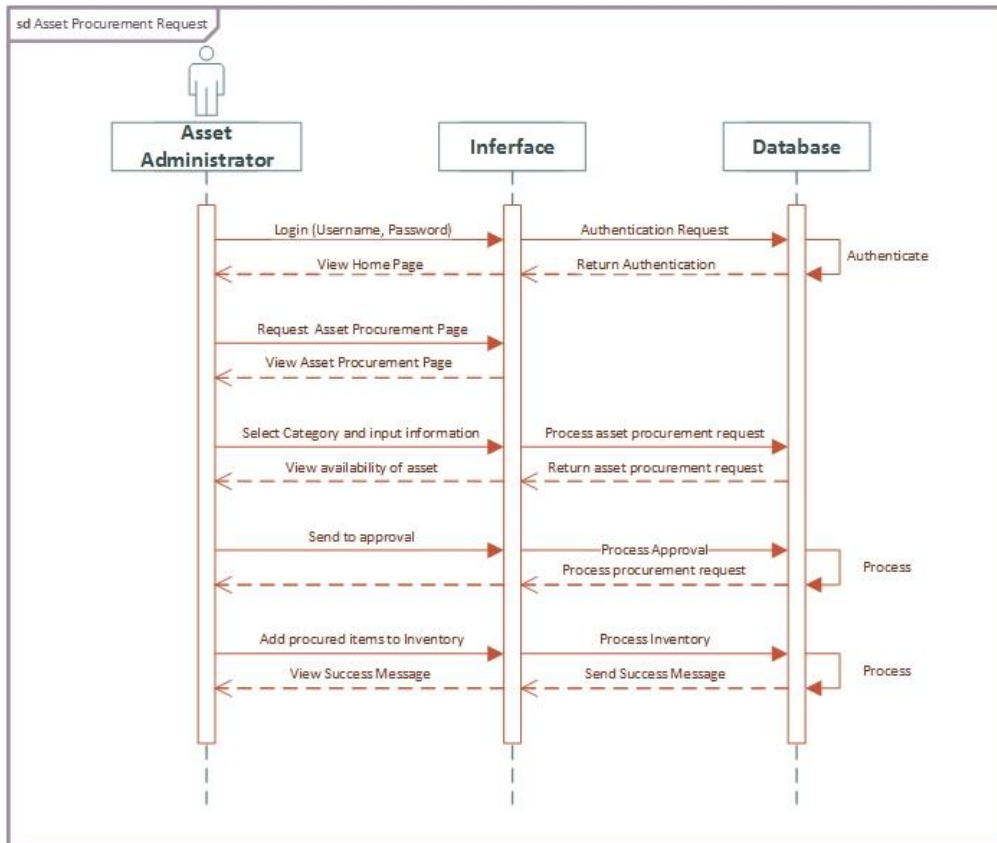


Figure 3.6 Asset Procurement Request Management Sequence Diagram

### 3.3.7 Activity Diagrams

Asset Release Management and Asset Procurement Request Management modules is displayed by figure 3.7 and figure 3.8 respectively.

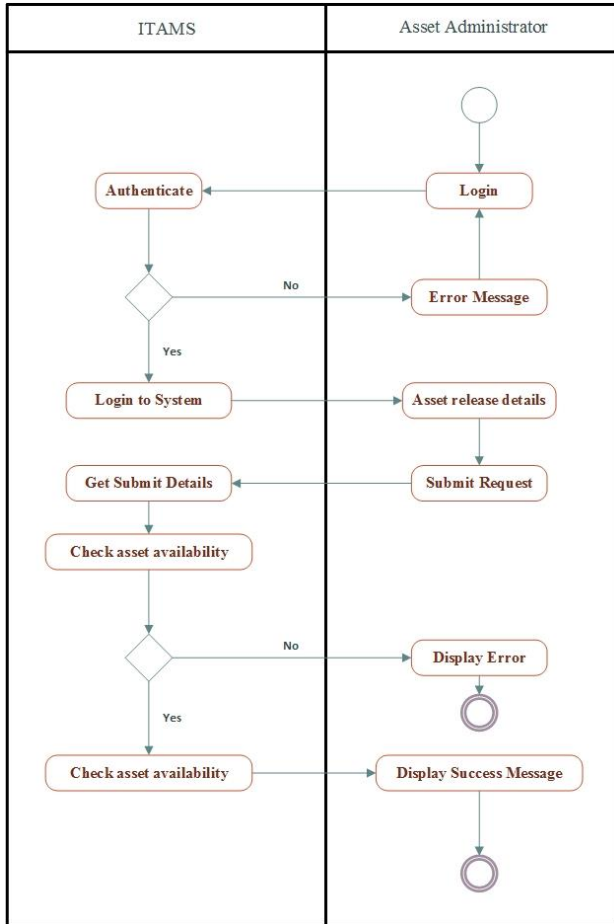


Figure 3.7 Asset Release Activity Diagram

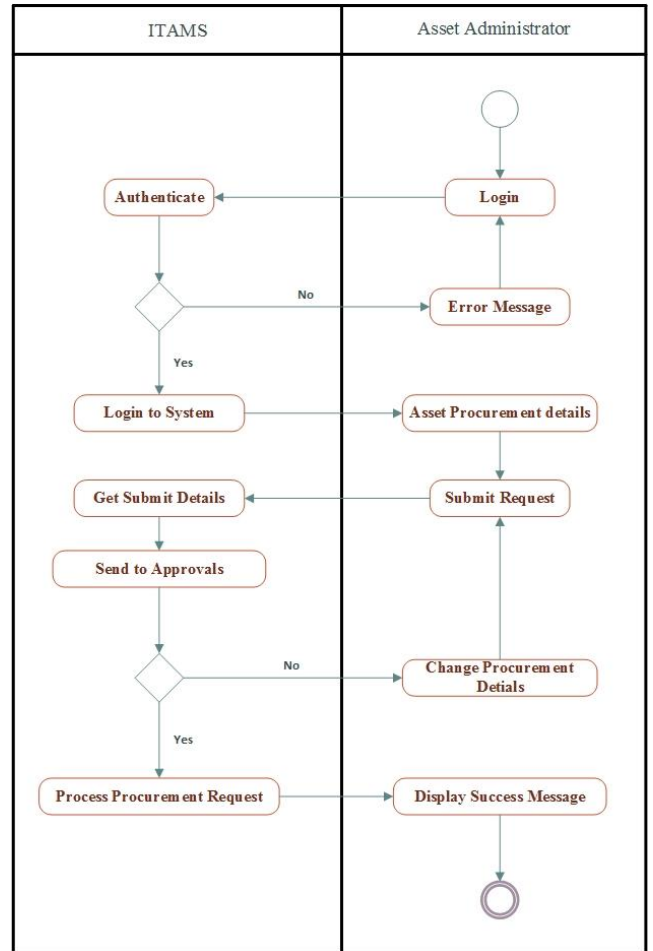


Figure 3.8 Asset Procurement Request Activity Diagram

### 3.4 INTERFACE DESIGN

User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the elements of interface that are easy to access, understand, and use to facilitate those operations. UI brings together concepts from interaction design, visual design, and information architecture. Below practices will be used when designing the interfaces.

- Keep the interface simple.
- Create consistency and use common UI elements.
- Strategically use color and texture.

There are several actions were taken in order to ensure the good and user-friendly interfaces throughout whole system.

- Eye friendly colors were used for this system. (e.g. light blue has been used)
- Buttons are designed with flat colors which is user friendly and eye catching.
- Tab indexes were used for easily navigation.
- Easy accessibility provided by using keyboard function which will increase the effectiveness of the system.
- Meaningful names and texts were used for button, forms and labels for increase the effectiveness of the system.

Home Page interface design is displayed by figure 3.9.

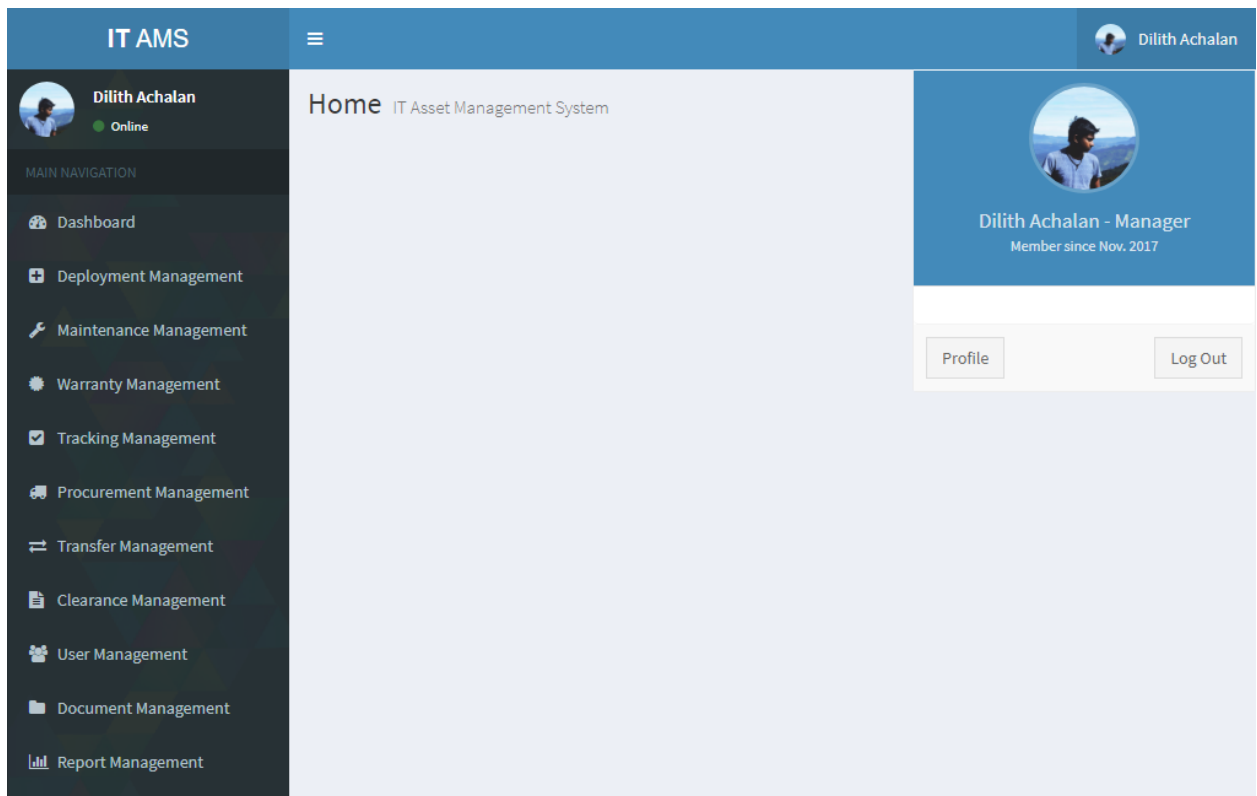


Figure 3.9 Home Page Interface Design

Login interface design with validation is displayed by figure 3.10.

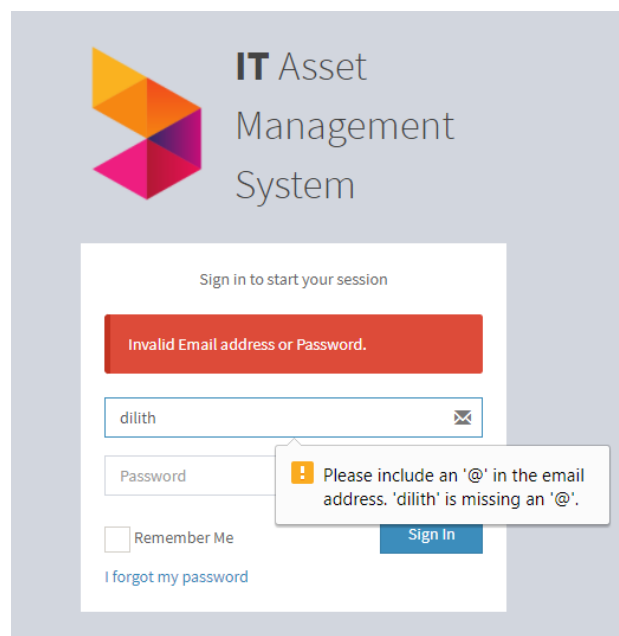


Figure 3.10 Login Page with validation Interface Design



Data table design, with search, pagination and sortable functions is displayed by figure 3.11.

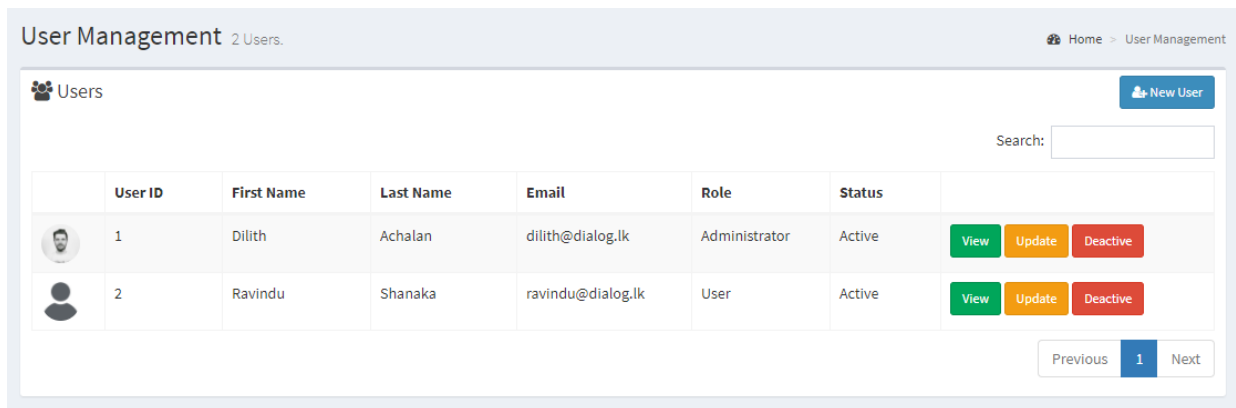


Figure 3.11 Data Tables Interface Design

Form and form validation is displayed by figure 3.12.

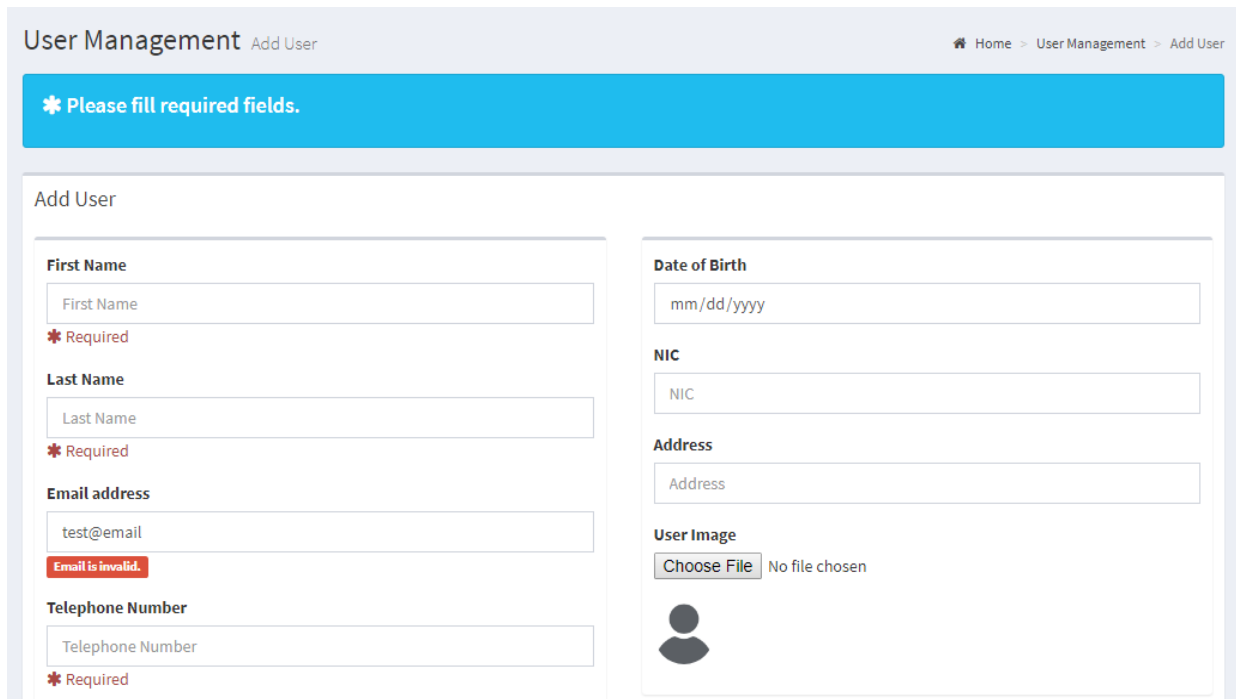


Figure 3.12 Form Interface Design

Sample report design is displayed by figure 3.13.

### IT Asset Management System | Report Management

PO ID	Category	Brand/Model	Quantity	Request Date	Description	Status
P00001	Laptop	Dell Latitude 7370	0	2017-08-28	renew	Approved
P00001	LCD	HP 19 Inch	0	2017-07-28	New	Approved
P00002	Dotmatix	Epson LQ310	14	2017-06-28	Brach printing	Approved
P00002	LED	HP ProBook 4515s	15	2017-09-28	new	Approved
P00003	IP Phone	Cisco 7911	13	2017-03-28	Extension	Pending
P00004	MPS - Lexmark Print Management	Lexmark 611w	9	2017-02-28	Printing	Pending
P00005	IP Phone	Cisco 1706	7	2017-01-28		Pending
P00006	Dotmatix	Epson LQ-310	10	2017-05-28		Pending
P00007	LCD	HP 19 Inch	10	2017-04-28		Pending
P00008	Desktop	Dell Optiplex 3010	31	2017-10-28		Pending
P00009	Desktop	Dell Latitude E5470	3	2017-11-05		Pending

Figure 3.13 Sample Reports Interface Design

Sample chart design is displayed by figure 3.14 and figure 3.15.

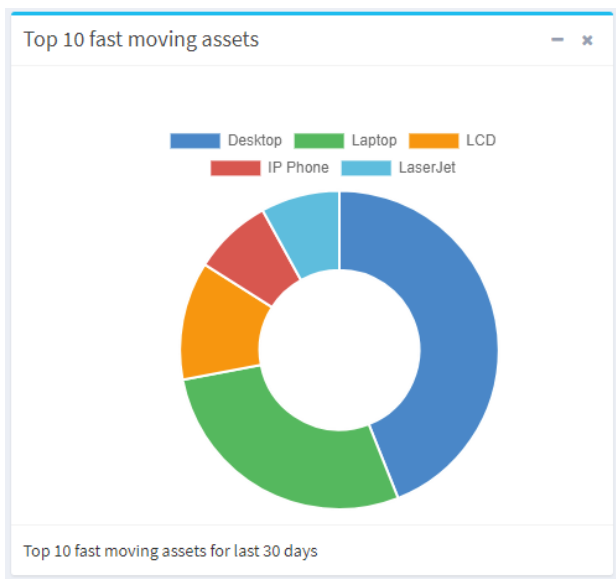


Figure 3.14 Charts – Top 10 fast moving

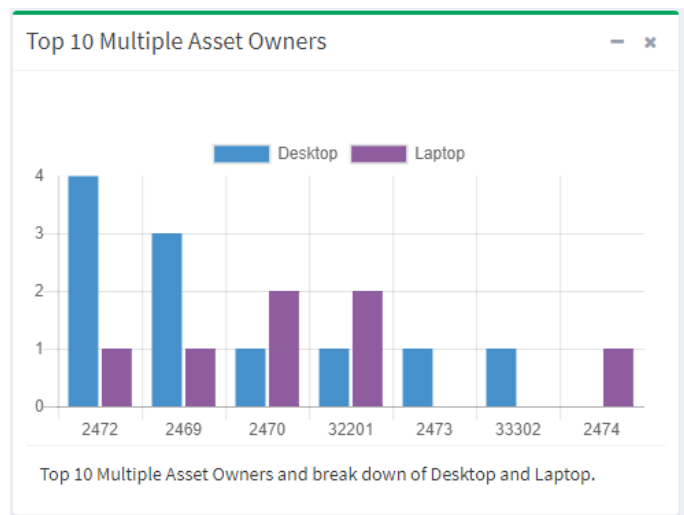


Figure 3.15 Charts – Top 10 Asset owners

# CHAPTER 04 – IMPLEMENTATION

## 4.1 INTRODUCTION

In this phase, system design converts into a complete information system using suitable tools and techniques. In the Software Development Life Cycle, the actual code is written here, and if the system contains hardware, then the implementation phase will contain configuration and fine-tuning for the hardware to meet certain requirements and functions.

Various tools and technologies were devised to aid in the implementation and coding aspects of the system. Comments are included for later references which will be very useful when incorporating changes to the system. Various coding standards and best practices were followed when engaging in the coding process.

## 4.2 IMPLEMENTATION ENVIRONMENT

The System Implementation Environment can be divided into two categories, Hardware Environment as well as Software Environment and listed below in table 4.1.

Hardware Environment	Software Environment
Intel ® Core™ 2 Duo CPU 2.00 GHz	Microsoft Windows 7
4GB RAM	XAMPP Version 3.2.2
500GB HDD	PHP Version 5.6.15
	Apache 2.4.17
	MariaDB 10.1.9

*Table 4.1 Implementation Environment*

### 4.2.1 Development Tools

- XAMPP - Free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database [WWW6]
- NetBeans IDE 8.2 – Used for coding.
- Google Chrome – Used as the standard browser.

## 4.2.2 Technologies

The following technologies were used during implementation and listed below in table 4.2.

Technologies	Description
PHP	PHP used as the primary development language for the system. PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language
MariaDB	MariaDB is used to implement all aspects related to database. It's community-developed fork of the MySQL relational database management system intended to remain free under the GNU GPL
HTML	Hypertext Markup Language is the standard markup language for creating web pages and web applications and widely used for web development
CSS	CSS is a style sheet language for describing the presentation of web pages and to style the HTML elements to make the interface more user friendly and attractive.
JavaScript and jQuery	JavaScript and jQuery used for performing client-side validations. JavaScript is a lightweight, object-oriented language with many in-built functions to cater to different needs while jQuery is a cross-platform JavaScript library which provides various widgets and other effects to simplify JavaScript coding.
AJAX	Ajax is a set of Web development techniques using many Web technologies on the client side to create asynchronous Web applications which means can be retrieve data from database without refreshing the page.

*Table 4.2 Technologies*

## 4.3 CODE AND MODULE STRUCTURE

### 4.3.1 Application Architecture

The system was built on top of MVC architecture. Model View and Controller divides a given application into three interconnected parts. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development. Figure 4.1 and figure 4.2 shows how MVC is being interacted.

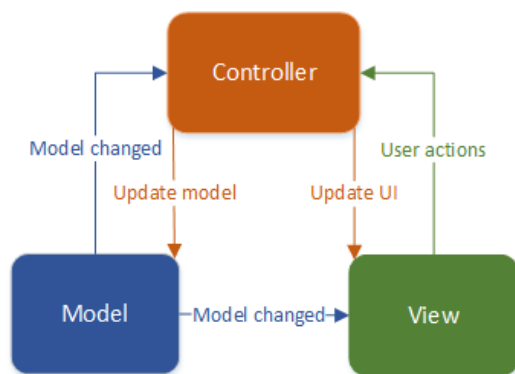


Figure 4.1 MVC Interaction Environment

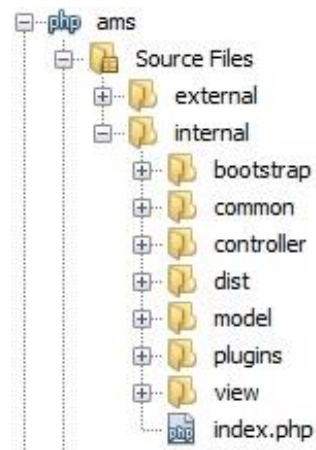


Figure 4.2 System Model Structure

**Model** - Stores data that is retrieved according to commands from the controller and displayed in the view which includes all CRUD operations.

**View** - The GUI generates new output to the user based on changes in the model.

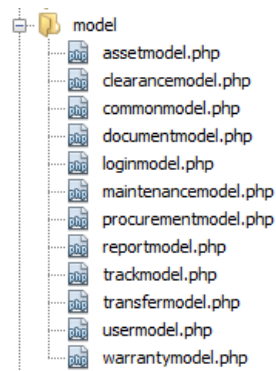
**Controller** – Controller acts as a mediator between model and view. Controller can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model.

## Model Structure

Below main modules were created for the major modules of the system as shows in figure 4.3.

- Asset model
- Procurement model
- Maintenance model
- Warranty model
- Transfer model
- Clearance model
- Track model.
- Document model
- User model
- Login model
- Report model

All other common modules were represented by the common model such as for managing category, employee and products etc.



*Figure 4.3 Model Structure*

## 4.3.2 Code Structure

### Database Connection

The following code fragment used to establish database connection using MySQLi object-oriented programming language. The parameters passed to SQL statement which should be executed. The die condition is being used to display any errors that might occur during database connectivity process.

```
<?php
// Database Connection Class
class dbconnection {

    public $host = "localhost";
    public $user = "root";
    public $password = "";
    public $database = "db_ams";

    public function connection() {
        //Connection string
        $con = new mysqli($this->host, $this->user, $this->password, $this->database);

        if (!$con) {
            // If query not executed
            die('Connect Error: ' . mysqli_connect_error());
        }
        return $con;
    }
}
?>
```

## **Login to the system**

Users are provided a login screen to enter username and password to access the system. After the validation process in view is done, the username and password sent to the controller. The controller sends the values fed from view to model. Model perform server-side validation and check with the database values. If the values provide by the user are valid, the controller will be redirecting to the home page. If the values are invalid, user will be redirected to login screen with error messages.



```

<?php

// Start the session
if (!isset($_SESSION)) {
    session_start();
}

// Get the values, username and password
// Server side validation
if ($_POST['email'] == "" || $_POST['pass'] == "") {
    $msg = base64_encode("Empty Email Address or Password.");
    //Data passing through URL
    header("Location:../view/login.php?msg=$msg");
    exit();
}

// Database connection
include '../common/dbconnection.php';
$obj = new dbconnection();
$con = $obj->connection();

// Retrieve the username and password
$email = trim($_POST['email']);
//Encrypt the password text to SHA1
$pass = sha1(trim($_POST['pass']));

// Include login model
include '../model/loginmodel.php';

// Create a object using class
$objlogin = new login();
//Call function using the object and get result 0 or 1
$result = $objlogin->loginvalidate($email, $pass);

if ($result->num_rows == 1) {
    // To get user details
    $row = $result->fetch_array();
    $_SESSION['user_info'] = $row;
    $msg = base64_encode("Login Successful.");
    header("Location:../view/dashboard.php");
} else {
    $msg = base64_encode("Invalid Email address or Password.");
    header("Location:../view/login.php?msg=$msg");
}
?>

```

As per above code, if the number of records from the model is equal to 0, it explains that the intended user is entering invalid username or password and user will redirect to login screen. If a corresponding value is found on database, the session will create with logged in user's details and will be redirected to user's homepage.

```
<?php
class login {
    public function loginvalidate($email, $pass) {
        $con = $GLOBALS['con'];
        //sql query; To join (Login, User, Role) tables
        $sql = "SELECT * FROM tbl_login l, tbl_user u, tbl_role r WHERE l.email='$email' AND"
            . " l.password='$pass' AND l.user_id=u.user_id"
            . " AND u.role_id=r.role_id AND u.user_status='Active'";
        //Execute Query
        $result = $con->query($sql);
        //check if any email and password matched, and return 0 or 1
        return $result;
    }
}
?>
```

## Logout from the system

The below code fragment is unset the session and user will be redirected to login page and ready to be used by user.

```
<?php
//To start the Session
if(!isset($_SESSION)){
    session_start();
}
// to Set Default timezone
date_default_timezone_set('Asia/Colombo');

include '../common/dbconnection.php';
$obj=new dbconnection();
$con=$obj->connection();

// To remove session by session
unset($_SESSION['user_info']);
// To redirect within 5 seconds
header("refresh:5; url=../view/login.php");
?>
```

## User role and privileges

Below code fragment is in place in session.php and make sure to get the user role. It will grant access to relevant authorized user to carry out certain tasks.

```
$user_info = $_SESSION['user_info'];  
$name = $user_info['user_fname']." ".$user_info['user_lname'];  
$role_name = $user_info['role_name'];  
$user_id = $user_info['user_id'];  
$role_id = $user_info['role_id'];  
$user_image = $user_info['user_image'];
```

## Controller structure

Controller acts as a mediator between model and view. Controller can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model. Below code fragment shows how different functions based on the action received from the view.

```

<?php
include '../common/dbconnection.php';
$obj = new dbconnection();
$con = $obj->connection();

//Include asset model
include '../model/assetmodel.php';
$assetObj = new asset();

$action = strtolower($_REQUEST['action']);

switch ($action) {
    case "issue":
        // Issue asset to a employee
        $assetArr = $_POST;
        $assetObj->issueAsset($assetArr);
        header("Location:../view/asset.php?action=issue");
        break;

    case "add":
        // Add new assets
        $assetArr = $_POST;
        $request_id = $_REQUEST['request_id'];
        print_r($assetArr);
        $assetObj->addAsset($assetArr,$request_id);
        header("Location:../view/assetacquisition.php");
        break;
}
?>

```

## Model structure

The following code fragment shows how asset issue function is processed.

```
public function issueAsset($arr) {
    $con = $GLOBALS['con'];
    //sql query;
    $user_id = $arr['user_id'];
    $employee_id = $arr['employee'];
    $asset_id = $arr['asset_id'];
    $issue_date = $arr['issue_date'];
    $allocation_status = $arr['allocation_status'];

    $sql = "INSERT INTO tbl_asset_allocation (user_id, employee_id, asset_id, issue_date, "
        . "allocation_remarks, allocation_status) VALUES ('$user_id','$employee_id', '$asset_id',"
        . " '$issue_date', NULL, '$allocation_status')";
    //Execute Query
    $result = $con->query($sql);

    $update="UPDATE tbl_asset SET asset_status='$allocation_status' WHERE asset_id='$asset_id'";
    $con->query($update) or die($con->error);
    return $result;
}
```

The following code fragment shows the process of adding assets to the database. In the same function system will enter warranty details and set remaining value to procurement request asset count.

```
public function addAsset($arr,$request_id) {
    $con = $GLOBALS['con'];
    //sql query;
    $user_id = $arr['user_id'];
    $category_id = $arr['category_id'];
    $brand_name = $arr['brand_name'];
    $model_name = $arr['model_name'];
    $po_no = $arr['po_number'];
    $added_date = $arr['added_date'];
    $asset_status = "New";
    $asset_quantity = $arr['asset_quantity'];
    $asset_quantity_select = $arr['asset_quantity_select'];
    $vendor = $arr['vendor'];
    $purchase_date = $arr['purchase_date'];
    $warranty_period = $arr['warranty_period'];
}
```

```
switch ($warranty_period) {
    case "3Month":
        $warranty_period = "3 Months";
        $wexpire_date = strtotime('+3 month', strtotime($purchase_date));
        $wexpire_date = date('Y-m-j', $wexpire_date);
        break;
    case "6Month":
        $warranty_period = "6 Months";
        $wexpire_date = strtotime('+6 month', strtotime($purchase_date));
        $wexpire_date = date('Y-m-j', $wexpire_date);
        break;
    case "1Year":
        $warranty_period = "1 Year";
        $wexpire_date = strtotime('+1 year', strtotime($purchase_date));
        $wexpire_date = date('Y-m-j', $wexpire_date);
        break;
    case "2Year":
        $warranty_period = "2 Years";
        $wexpire_date = strtotime('+2 year', strtotime($purchase_date));
        $wexpire_date = date('Y-m-j', $wexpire_date);
        break;
    case "3Year":
        $warranty_period = "3 Years";
        $wexpire_date = strtotime('+3 year', strtotime($purchase_date));
        $wexpire_date = date('Y-m-j', $wexpire_date);
        break;
}
```

```

$warranty_status = "Valid";

for($x=1; $x<=$asset_quantity_select; $x++){
    $serial_nox = strtoupper($arr['serial_no'].$x);
    $sql = "INSERT INTO tbl_asset (category_id, user_id, brand_name, model_name, serial_no, "
        . "po_no, asset_status, added_date) VALUES ('$category_id','$user_id', '$brand_name', "
        . "'$model_name', '$serial_nox', '$po_no', '$asset_status', '$added_date')";
    //Execute Query - Insert asset
    $result = $con->query($sql);
    $asset_id = $con->insert_id; //Last ID

    //Execute Query - Insert asset warranty details
    $sqlWarranty = "INSERT INTO tbl_warranty (asset_id, purchase_date, warranty_period, "
        . "wexpire_date, vendor_id, warranty_status) VALUES ('$asset_id','$purchase_date', "
        . "'$warranty_period', '$wexpire_date', '$vendor', '$warranty_status')";
    $result = $con->query($sqlWarranty);
}

if($asset_quantity_select<=$asset_quantity){
    $remainQuantity = $asset_quantity - $asset_quantity_select;

    // Set Remaining Value
    $update="UPDATE tbl_request SET asset_quantity='$remainQuantity' WHERE request_id='$request_id'";
    $con->query($update) or die($con->error);
}

return $result;
}

```

## View overview

The following code fragment shows an instance of a view section.

```
<div class="box-body">
  <table id="categoryTable" class="table table-bordered table-striped">
    <thead>
      <tr>
        <th>&nbsp;</th>
        <?php if($action=="transfer" || $action=="clearance"){ ?>
        <th>Allocation ID</th>
        <?php } ?>
        <th>Asset ID</th>
        <th>Category Name</th>
        <?php if($action=="transfer" || $action=="clearance"){ ?>
        |   <th>Employee Name</th>
        <?php } ?>
        <th>Brand Name</th>
        <th>Model Name</th>
        <th>Serial Number</th>
        <th>Asset Status</th>
        <th>&nbsp;</th>
      </tr>
    </thead>
  </table>
</div>
```



```
|  |
| --- |
| <td>&nbsp;</td>                 <?php if($action=="transfer" || $action=="clearance"){ ?>                 <td><?php echo $row['allocation_id']; ?></td>                 <?php } ?>                 <td><?php echo $row['asset_id']; ?></td>                 <td><?php echo $row['category_name']; ?></td>                 <?php if($action=="transfer" || $action=="clearance"){ ?>                 <td><?php echo $row['employee_fname'].' '.$row['employee_lname']; ?></td>                 <?php } ?>                 <td><?php echo $row['brand_name']; ?></td>                 <td><?php echo $row['model_name']; ?></td>                 <td><?php echo $row['serial_no']; ?></td>                 <td><?php echo $row['asset_status']; ?></td> |

```

```

                <td>
                <?php
                if($action=="issue"){
                ?>

                <a href="issueasset.php?asset_id=<?php echo $row['asset_id']; ?>&action=issue">
                <button type="button" class="btn btn-sm btn-success">Issue</button></a>
                <?php }else if($action=="transfer"){ ?>
                <a href="transferasset.php?allocation_id=<?php echo $row['allocation_id']; ?>
                &asset_id=<?php echo $row['asset_id']; ?>&action=transfer">
                <button type="button" class="btn btn-sm btn-warning">Transfer</button></a>
                <?php }else if($action=="maintenance"){ ?>
                <a href="addmaintenance.php?asset_id=<?php echo $row['asset_id']; ?>
                &action=maintenance">
                <button type="button" class="btn btn-sm btn-warning">Maintenance</button></a>
                <?php }else if($action=="clearance"){ ?>
                <a href="" data-href="../controller/clearancecontroller.php?
                allocation_id=<?php echo $row['allocation_id']; ?>
                &asset_id=<?php echo $row['asset_id']; ?>&employee_id=<?php echo $row['employee_id']; ?>
                &action=clearance" data-toggle="modal" data-target="#confirm-status"
                aria-label="category-continue-discontinue">
                <button type="button" class="btn btn-sm btn-warning">Clearance</button></a>
                <?php } ?>
                </td>
            </tr>
        <?php } ?>
    </tbody>
</table>
</div>

```

## 4.4 REUSED COMPONENTS

- AdminLTE - AdminLTE is an open source project that is licensed under the MIT license and can be used as open source admin dashboard & control panel theme. [WWW7]
- DataTables - It is a plug-in for the jQuery JavaScript library which is equipped with advanced interaction controls over HTML tables. Inbuilt pagination facility and advanced searching ability makes this a highly flexible tool. [WWW8]
- jQuery Datepicker – A jQuery plugin which provides a simple, highly configurable datepicker to easily navigate through months and years. [WWW9]
- Bootstrap - Bootstrap is a free and open-source front-end web framework for designing websites and web applications [WWW10]
- Font Awesome and Ionicons – For Icons and graphics [WWW11]
- iCheck – for responsive inputs, such as check box and radio box. [WWW12]
- Chart.js – Chart.js used to create interactive management reports and dashboards. There are so many varieties of charts available to display information effectively. [WWW13]

# CHAPTER 05 – EVALUATION

## 5.1 INTRODUCTION

Testing is the process of evaluating a system with the intent to find whether application meets the business and technical requirements and works as expected. Software Testing is a critical element of software quality assurance. Verification involves ensuring whether the built software conforms to its specification while validation refers to the set of activities that ensures that the software has been built in traceable to the expectations of the customer. [WWW14]

## 5.2 TEST PLAN

“A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from test engineers.”

Planning a test case is a very important aspect for developing system as well as for the completed system. A suitable test plan was designed prior to the implementation of the system which should have the ability to test the functionality of the overall system. By properly testing a system, it can identify the errors which generate from the system and can correct them. The implemented system was tested using different test cases. The following tables specify a selection of test cases designed for the testing process.

[WWW15]

The following tables specify a selection of test cases designed for the testing process.

### 5.2.1 Deployment Management Module

ID	Test Description	Testing Procedure	Expected Result
1	View all allocated assets	Click the allocated asset button.	The system should display all allocated assets.
2	View new or reused assets	Click asset release button.	The system should display all new and reused assets.
3	Retrieve all eligible employees.	Select employee ID from the drop down or type part of the employee ID	The user should able to select employee ID or type part of the employee ID. List of employees corresponding to employee ID should display

4	Allocate assets to a specific employee	Select an employee who has currently allocated assets	A list of allocated assets to a selected employee should display.
5	Issue a new asset to an employee	Select or type employee number to allocate asset and click submit	The asset will be allocated to an employee and update the asset status.
6	Add a new asset category (positive)	Fill all the required fields with appropriate data.	A new asset category will add to asset catalogue
7	Add a new asset category (negative)	Fill all the required fields with invalid data such as special characters and numbers.	An error should appear saying invalid characters for category
8	Edit existing asset category	User can either type or select category and enter valid data.	Asset category will be updated with edited values.
9	View category	Select a category to view detailed information.	A new page will open with category information corresponding to category ID.
10	Continue or discontinued asset category	User can continue or discontinue an asset category	Confirmation message should pop up and click confirm button and the asset should be set as either continued or discontinued depending on the current status.
11	Data validation.	Enter invalid data or leave the mandatory fields blank.	Display appropriate error messages.

*Table 5.1 Test Cases for Deployment Management Module*

## 5.2.2 Maintenance Management Module

ID	Test Description	Testing Procedure	Expected Result
1	Select asset to add maintenance details	Click asset maintenance button.	The system should display all assets to add any maintenance details.
2	Add asset maintenance details (positive)	Fill all required fields in the New asset maintenance page.	Asset maintenance details should be added to the database.
3	Add asset maintenance details. (negative)	Fill invalid or leave required fields blank.	Asset maintenance details should not be added to the database. The system should display appropriate error messages.
4	View asset repair history.	Select the relevant record and click the view button in the asset maintenance history page.	The system should display view asset maintenance page according to other selected record.
5	Update asset repair history. (positive)	Select the relevant record and click the update button in the asset maintenance history page and then fill all the required fields.	The system will display update asset maintenance history page according to the selected record and update the database.
6	Update asset repair history. (negative)	Select the relevant record and click the update button in the asset maintenance history page and fill invalid or leave required fields blank.	The system will display update asset maintenance history page according to the selected record and should not update the database due to blank or invalid data
7	Data validation.	Enter invalid data or leave the mandatory fields blank.	Display appropriate error messages.

*Table 5.2 Test Cases for Maintenance Management Module*

### 5.2.3 Warranty Management Module

ID	Test Description	Testing Procedure	Expected Result
1	View expired warranty details.	Click the warranty expired assets button in the warranty management interface.	System will display the expired assets details in the warranty status interface.
2	View warranty status	Click the view warranty button in the warranty management interface.	System will display both warranty valid and warranty expired assets details in the warranty status interface.
3	Renew warranty details.	Search a specific asset from view warranty interface and click the renew button and then fill all required details in the renew warranty page.	The system will display all the records which include warranty details and update the database.
4	Data validation.	Enter invalid data or leave the mandatory fields blank.	Display appropriate error messages.

*Table 5.3 Test Cases for Warranty Management Module*

### 5.2.4 Tracking Management Module

1	View transfer history.	Click the transfer tracking button in the tracking management interface.	System will display transfer history page.
2	View record of individual asset.	Click the view button in the transfer history page.	System will display transfer history of individual asset.
3	Data validation.	Enter invalid data or leave the mandatory fields blank.	Display appropriate error messages.

*Table 5.4 Test Cases for Tracking Management Module*

### 5.2.5 Procurement Management Module

1	Select asset category to request asset	Click new request and select asset category	All asset categories should retrieve except discontinued assets.
2	Select asset model and product name	Click new request and select or type product name and asset model	All asset product names and model names should retrieve from database.
3	New asset request.	Click the new request button in the procurement request management page and then fill all required fields in new procurement request page.	System will display new procurement request page and update the database.

4	Assets add to same purchase order	Once new request added, add more assets to the same request	Assets should add to the same purchase order ID
5	View all pending and approved request.	Click the request status button in the procurement request management interface.	Request status page will be displayed. Both approved and pending request should be displayed
6	View individual pending or approved request.	Click the view button on the selected record in request status.	System should display request details of the corresponding record.
7	Edit approved request.	Login to the system as any user and check if the edit button is available.	Once the asset request is approved, system should not display edit button and should not allow to edit any approved request.
8	Edit pending request.	Click the edit button on the selected record and then update the details in the update procurement request page.	System will display update procurement request page with current values and update the database.
9	Manager should be able to approve the asset request.	Login to system as a manager and click approve button in request status page.	The manager is only able to approve the request. For any other users approve button must not visible.
10	Other user roles cannot be able to approve asset request.	Login to system as a system administrator or asset administrator.	System should not display approve button and should not able to approve any asset request.
11	Reject Pending request.	Login to system as a manager and click reject button in request status page.	The record should be rejected and updated to the database.
12	Add asset acquisition details.	Click the asset acquisition button in the procurement request management page and click the add button in the asset acquisition page and then fill required fields in the add purchased assets page	The system will display asset acquisition and add purchased assets pages and update the database.
13	Select add asset count	Select asset count from add purchased assets page.	Corresponding to the asset count serial number fields should be generated.

14	Add warranty details	In add purchase assets, check warranty is available and enter appropriate details	Warranty details will update corresponding to the purchased assets
15	Data validation.	Enter invalid data or leave the mandatory fields blank.	Display appropriate error messages.

*Table 5.5 Test Cases for Procurement Management Module*

### 5.2.6 Transfer Management Module

ID	Test Description	Testing Procedure	Expected Result
1	Select asset to transfer	Click asset transfer button and select specific asset to transfer	Only allocated assets should be displayed to transfer
2	Transfer assets to a specific employee	Select or type an employee ID who has currently allocated assets.	A list of allocated assets to a specific employee should display.
3	Transfer asset.	Click transfer button on asset inventory and select or type employee ID and submit to transfer	System should update the database.
4	Data validation.	Enter invalid data or leave the mandatory fields blank.	Display appropriate error messages.

*Table 5.6 Test Cases for Transfer Management Module*



### 5.2.7 Clearance Management Module

ID	Test Description	Testing Procedure	Expected Result
1	View all issued assets.	Click asset clearance button in the clearance management page.	System should show all issued assets.
2	Clear assets.	Click the clearance button on the selected row.	Once click the confirm button, confirmation message should pop up and the asset should be cleared.
3	View clearance history.	Click clearance history button in the clearance management page.	System will display all cleared assets.
4	View individual details of a cleared asset.	Click view button in the selected record in the clearance history page.	System will display details of selected record.
5	Reallocate asset.	Click reuse button in the selected record in the clearance history page.	Confirmation message should pop up and once click confirm button and the asset should be able to reuse.
6	Dispose asset.	Click dispose button in the selected record in the clearance history page.	Confirmation message should pop up and once click confirm button and the asset should be dispose and should not able to reallocate.
7	Data validation.	Enter invalid data or leave the mandatory fields blank.	Display appropriate error messages.

*Table 5.7 Test Cases for Clearance Management Module*

## 5.2.8 User Management

ID	Test Description	Testing Procedure	Expected Result
1	Add a new user.	Click new user in user management page.	A new user should be able to be added to the system.
2	Check email address availability (positive)	Enter a new email address.	A success message should show that the email address is available to use.
3	Check email address availability (negative)	Enter an existing email address.	An error message should show that the email address is not available to use.
4	Check email address availability (negative)	Enter email address in invalid format	An error message should show that the email address is invalid.
5	View all users.	Click user management in menu.	The system should display all users with minimal details.
6	View a specific user.	Click view button for a selected user in user's page.	The system should display a detailed information about selected user in read only view user page.
7	Update a specific user.	Click update button for a selected user in user's page.	The system should display current values in update user page and should be able to update the user once submit.
8	Activate user.	Click active button of any deactivated user.	Confirmation message should pop up and click confirm button and user should be activated.
9	De activate user.	Click deactivate button of any activated user.	Confirmation message should pop up and click confirm button and user should be deactivated.
10	Data validation.	Enter invalid data or leave the mandatory fields blank.	Display appropriate error messages.

*Table 5.8 Test Cases for User Management Module*

### 5.2.9 Report Management Module

ID	Test Description	Testing Procedure	Expected Result
1	Generate reports.	Select the necessary parameters and click generate button.	Necessary reports should be generated and displayed on screen.
2	Save and print reports.	Generate the reports by fill in the necessary parameters.	Save or print the report according to the relevance.
3	View asset insights	Click asset insights button on reports management page.	Asset insights should display to managers to get managerial decisions.
4	Data validation.	Enter invalid data or selection.	Display error messages.

*Table 5.9 Test Cases for Report Management Module*

### 5.2.10 Document Management Module

ID	Test Description	Testing Procedure	Expected Result
1	Add new document	Click new document in document management page.	Add document page should open to enter document details.
2	Upload multiple files	Select document count in add document page.	File attachment fields should display corresponding to the count of documents.
3	View all document	Click view document in document management page.	The system should display all documents with minimal details.
4	View a specific document	Click view button in view documents page.	The system should display a detailed information about selected document in view files page.
5	Review a document	Click revise button for any revise pending document.	A confirmation message should pop up and the document should mark as revised.
6	View a specific file in a document	Click view button in view files page.	A new tab should open with the file.
7	Remove a specific file	Click remove button in view files page.	The specific file should be removed from the system.

*Table 5.10 Test Cases for Document Management Module*

### 5.2.11 Common Functions Module

ID	Test Description	Testing Procedure	Expected Result
1	Login testing (negative).	Enter an invalid or a wrong combination of username and password.	An error message must be displayed indicating the error.
2	Login testing (positive).	Enter a valid username and a password.	Direct to the dashboard for quick access.
3	Login to the system without entering necessary details (negative).	Click the login button leaving the username and/or password fields blank.	An error message is displayed indicating the fields are required.
4	Logout from the system.	Click the logged in user name and click logout button given.	User must be logged out and must be redirected to the login page.
5	System visualization across browsers and different screen sizes	Access the system in various browsers and use different screen sizes.	The system interfaces must look the same across different browsers and the interface should scale to the screen size.
6	Identify the user and display in homepage.	Login to the system by entering a valid username and password.	Display the user's first name and last name with the profile picture in the homepage.
7	Request to view a record.	Click on the view button with a document icon displayed at the end of the record set.	Directs to a form loaded with the values stored in the database displayed in a read-only mode.
8	Request to edit a record.	Click on the edit button with a document and pen icon displayed at the end of the record set.	Directs to the update form filled with the values stored in the database with a submit button.
9	Clear all the field values in a form.	Click the reset button	The field values in the form must be cleared.
10	Trying to access a page without logged in.	Enter the URL of a page without logged in.	Deny access and redirect to login page.
11	Trying to access a page without authorization (logged in).	Enter the URL of a page without the privilege to access or view that page.	Deny access and redirect to login page.

Table 5.11 Test Cases for Common Functions

### 5.3 TEST RESULTS

The data fields that are found in various individual forms were adequately tested with the entry of sample dummy data to ensure the accuracy and reliability of each form field values. A common set of error messages were used in appropriate places and highlight input field to keep the flow of consistency across the system and to reduce confusion among system users.

Some test results are given below for some test cases and the rest of the test results can be found in Appendix E.

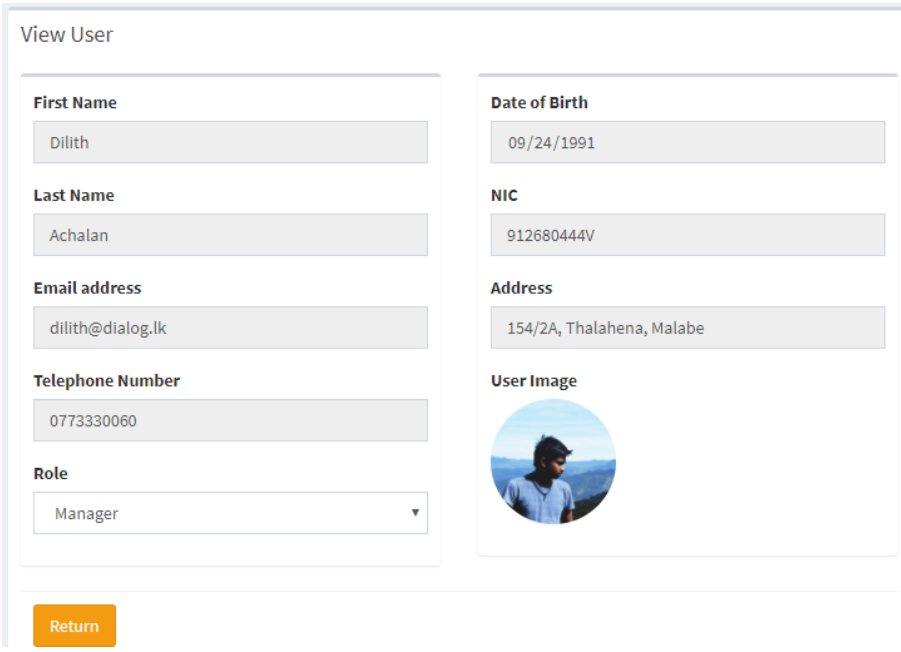
Test Case ID	6	
Module Name	User Management Module	
Test Case	View a specific user	
Expected Output	The system should display a detailed information about the selected user in read only view.	
No	Output	Status
1		✓

Table 5.12 Test Results for View a User Test Case

Test Case ID	1	
Module Name	User Management Module	
Test Case	Add a user	
Expected Output	A new user should be able to be added to the system.	
No	Output	Status
1		✓

Table 5.13 Test Results for Add a User Test Case

Test Case ID	10		
Module Name	User Management Module		
Test Case	Data validation.		
Expected Output	Display appropriate error messages.		
No	Test Case	Output	Status
1	Email address availability testing.	<p>Email address</p> <input type="text" value="nadun@lk"/> <p>Email is invalid.</p> <p>Email address</p> <input type="text" value="dilith@dialog.lk"/> <p>Email is already taken.</p> <p>Email address</p> <input type="text" value="nadun@dialog.lk"/> <p>Email is available to use.</p>	✓
2	Required inputs are empty.		✓

3	Wrong NIC format.		✓
4	Invalid telephone number		✓

Table 5.14 Test Results for Validation Test Case

## 5.4 USER EVALUATION

The Asset Management System was tested in an operational environment at the client premises in order to verify whether the intended functionalities specified in the specification were been implemented. Selected set of users with different user levels and rights were asked to work with the system and the system was positively accepted by the users. It is quite evident that the users were impressed of the system since most of the feedbacks received were positive. Some feedbacks and suggestions given by the client were incorporated to the system to give better satisfaction. The final results of the test indicated that the system has met its objectives and is easy to work with.

A questionnaire was used to assess the customer feedback. The following is a sample questionnaire result.

Ratings	A – Excellent	B - Good	C - Acceptable	D - Need to Improve
---------	---------------	----------	----------------	---------------------

*Table 5.15 Ratings*

Evaluation User Role: Asset Administrator

No	Evaluation Criteria	A	B	C	D
1	Simplicity - Ease of understanding system functions.		✓		
2	Attractiveness - Look and feel of the interfaces.	✓			
3	Effectiveness - Potential benefits derived from the system.	✓			
4	Efficiency - Easiness of navigating across the system.	✓			
5	Functionality - Extent of the functionalities covered.	✓			
6	Ease of generating reports.		✓		
7	Meaningful error messages.	✓			
8	Degree of information provided in the reports.		✓		
9	Task execution speed.		✓		
10	Clarity and consistency.	✓			

*Table 5.16 User Acceptance Test Results*

Moreover, some suggestions which were put forward by the users are displayed below.



# **CHAPTER 06 – CONCLUSION**

## **6.1 INTRODUCTION**

Dialog Axiata PLC is Sri Lanka's largest telecommunications service provider with the country's largest mobile operator of 11.8 million subscribers and commands a 50% of the Sri Lanka mobile market. The company has over 3500 employees in island wide, across 12 main branches, 20 regional engineering offices, over 50 customer field support engineer offices.

They need to enhance their IT Asset utilization by maintaining a proper IT asset management system. Earlier they faced many difficulties using spread sheets and using manual processing. Currently because of this IT asset management system, IT staff can carry out their daily activities effectively and efficiently. Managers can get insights or forecasts to make decision making easier.

All in all, it can be concluded that IT Asset Management System has a positive impact on Dialog Axiata PLC which is quite evident from the high level of user satisfaction witnessed.

## **6.2 CRITICAL ASSESSMENT OF THE PROJECT**

Each project objective was implemented successfully at client's location and below key achievements were able to get by using the implemented system. All spreadsheet processing was replaced with the new system. The client's expected level was achieved.

- The system was able to manage all IT Assets effective and efficiently.
- Data regarding IT Assets were properly organized and able to receive in a short period.
- The system was able to reduce time and manual effort related to deploying assets to employees, maintenance, tracking, clearance and procurement.
- Managers were able to get insights about IT Assets procurements.

## 6.3 FUTURE ENHANCEMENT

After successfully developing and deploying the entire system and based on the positive user feedback received in the preceding evaluation phase, the developed system incorporates all the features and functionalities expected by the client along with some additional features exceeding their expectations. Although the IT Asset Management System has achieved its main objectives, overcoming most problems and simplifying the operations, there are some improvements which are not carried out as the proposed system was developed within a time constrains and functionalities which were agreed by the client.

The following are the future enhancements that are to be incorporated into ITAMS in the days to come:

- Integrate with LDAP for authentication and get more employee details. System users will able to login to system by their domain credentials.
- Create Self portal for employees where they can review their own allocated assets.
- Conduct periodic asset review using self-portals.
- Add customized user roles. So that manager can create customized user roles.

# REFERENCES

- [WWW1] System Analysis and Design,  
[https://en.wikibooks.org/wiki/Systems\\_Analysis\\_and\\_Design/Introduction](https://en.wikibooks.org/wiki/Systems_Analysis_and_Design/Introduction)
- [WWW2] Existing Similar Systems,  
<https://www.manageengine.com/products/asset-explorer/documents.html>
- [WWW3] Existing Similar Systems,  
<https://www.trackit.com/whats-new/>
- [WWW4] System Design,  
[https://en.wikipedia.org/wiki/Systems\\_design](https://en.wikipedia.org/wiki/Systems_design)
- [WWW5] Object-oriented Design,  
[https://en.wikipedia.org/wiki/Object-oriented\\_design](https://en.wikipedia.org/wiki/Object-oriented_design)
- [WWW6] XAMPP,  
<https://www.apachefriends.org/index.html>
- [WWW7] AdminLTE,  
<https://adminlte.io/docs/2.4/license>
- [WWW8] JQuery DataTables,  
<https://datatables.net/>
- [WWW9] jQuery Datepicker,  
<https://jqueryui.com/datepicker/>
- [WWW10] Bootstrap,  
<http://getbootstrap.com/>
- [WWW11] Font Awesome and Ionicons,  
<http://fontawesome.io/license/>  
<http://ionicons.com/>
- [WWW12] iCheck,  
<http://icheck.fronteed.com/>
- [WWW13] Chart.js,  
<http://www.chartjs.org/>
- [WWW14] Evaluation,  
<https://en.wikipedia.org/wiki/Evaluation>
- [WWW15] Test Plan,  
[https://en.wikipedia.org/wiki/Test\\_plan](https://en.wikipedia.org/wiki/Test_plan)

# APPENDIX A - SYSTEM DOCUMENTATION

This documentation provides the necessary guidelines to aid in the installation process of IT Asset Management System. This also includes the minimum set of hardware and software requirements that should be met to install and operate the system.

## HARDWARE REQUIREMENT

Hardware	Recommended Minimum Requirements
Processor	Intel ® Core™ 2 Duo 2.00 GHz or similar.
Memory	1GB RAM or higher.
Display	1024 x 768 resolution or above. High Color 32-bit display.
Hard Disk space	Minimum 1GB free disk space or higher.
Printer	Dot-matrix, Inkjet or Laser Printer.
Internet	Minimum 512kbps connection.

*Table A.1 Hardware Requirements*

## SOFTWARE REQUIREMENT

Software	Recommended Minimum Requirements
Operating System	Microsoft Windows XP/Windows Vista/Windows 7/Windows 8, Linux Ubuntu 9.0 or above, or any other compatible OS.
Bundle Packages	XAMPP Version 3.2.2/LAMP/WAMP or Individual packages – MariaDB 10.1.9, PHP 5.6.15 Apache 2.4.17 phpMyAdmin 4.5.1
Code Editor	NetBeans IDE 8.2
Web Browser	Google Chrome/Mozilla Firefox/Internet Explorer/Opera/Safari.

*Table A.2 Software Requirements*

### Installation of IT Asset Management System

1. Install the necessary relevant software.
  - Download and install the XAMPP server solution stack.
  - Download and install a suitable web browser.
2. Setting up IT Asset Management System.
  - Browse through the given supplementary CD and copy the **ams** folder and paste it inside the following paths.
    - Windows Environment with XAMPP installed – “C:\xampp\htdocs”
    - Linux Environment with XAMPP installed – “/opt/lampp/htdocs”
3. Database Setup.
  - Open the browser and type the URL „http://localhost/phpmyadmin/”.
  - Create a new database named as “db\_ams” with the collation set to “utf8\_general\_ci”.

- Select the “db\_ams” database from the database list and click the Import tab.
- Browse through the supplementary CD and select the “db\_ams.sql” file and click on the go button.

### **Launching the System**

- Before launching the system, make sure whether Apache and MySQL services are running. To verify that, open the XAMPP Control Panel and find it out.
- Now open your preferred web browser and type the following URL:  
“http://localhost/ams/internal/view/login.php” or  
“http://127.0.0.1/ams/internal/view/login.php”
- Login and get access to the system by providing the correct username and the password.
- The above-mentioned URL can be bookmarked and you can then drag and drop that bookmark to your desktop for ease of access.

# APPENDIX B - DESIGN DOCUMENTATION

## USE-CASE DESCRIPTIONS

Some of the main use-case descriptions are mentioned below which provides an overview understanding about the design of the system.

### 1. Use-Case Description for Asset Allocation

<b>User-Case</b>	Allocate an asset
<b>Actors</b>	Asset administrator
<b>Overview</b>	
Asset are allocated to employees.	
<b>Pre-conditions</b>	
User should login to the system as an asset administrator	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Employee request for an asset.</li> <li>2. Asset administrator select the asset from asset inventory.</li> <li>3. Asset administer select the employee ID.</li> <li>4. Click submit button.</li> </ol>	
<b>Post conditions</b>	
Selected asset allocated to selected employee.	

*Table B.1 Use-Case Description for Asset Allocation*

### 2. Use-Case Description for Create New Asset Category

<b>User-Case</b>	Create new asset category
<b>Actors</b>	Asset administrator
<b>Overview</b>	
New asset category should be able to create.	
<b>Pre-conditions</b>	
User should login to the system as an asset administrator	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Check if the asset type is available.</li> <li>2. If available, select the asset category type or enter new asset category details.</li> <li>3. Sub-category should display corresponding to the asset category name.</li> <li>4. Click submit button to add new asset category.</li> </ol>	
<b>Post conditions</b>	
Selected asset allocated to selected employee.	

*Table B.2 Use-Case Description for Create New Asset Category*

### 3. Use-Case Description for New Asset Procurement Request

<b>User-Case</b>	New procurement request.
<b>Actors</b>	Asset administrator, System administrator
<b>Overview</b>	
New asset request should be able to create.	
<b>Pre-conditions</b>	
User should login to the system as an asset administrator or system administrator. Asset category should be available to before asset request.	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Check if the asset category is available.</li> <li>2. If available, select the asset category type or enter new asset category details.</li> <li>3. Sub-category should display corresponding to the asset category name.</li> <li>4. Click submit button to add new asset category.</li> </ol>	
<b>Post conditions</b>	
New asset request should generate and user may add more items to the same request.	

*Table B.3 Use-Case Description for New Asset Procurement Request*

### 4. Use-Case Description for Approve Asset Procurement Request

<b>User-Case</b>	Approve asset procurement request.
<b>Actors</b>	Manager
<b>Overview</b>	
Manager should able to approve asset procurement request.	
<b>Pre-conditions</b>	
User should login to the system as a manager.	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Click approve button on any pending asset procurement request.</li> <li>2. Confirm message should pop up and should able to click confirm.</li> <li>3. The asset procurement request should approve.</li> </ol>	
<b>Post conditions</b>	
User should not able to edit approved asset procurement request.	

*Table B.4 Use-Case Description for Approve Asset Procurement Request*

### 5. Use-Case Description for Reject Asset Procurement Request

<b>User-Case</b>	Reject asset procurement request.
<b>Actors</b>	Manager
<b>Overview</b>	
Manager should able to reject asset procurement request.	
<b>Pre-conditions</b>	
User should login to the system as a manager.	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Click reject button on any pending asset procurement request.</li> <li>2. Confirm message should pop up and should able to click confirm.</li> <li>3. The asset procurement request should reject</li> </ol>	
<b>Post conditions</b>	
The procurement request should reject.	

*Table B.5 Use-Case Description for Reject Asset Procurement Request*

### 6. Use-Case Description for Asset Transfer

<b>User-Case</b>	Transfer asset
<b>Actors</b>	Asset Administrator
<b>Overview</b>	
Asset administrator should able to transfer asset to a specific employee	
<b>Pre-conditions</b>	
<p>User should login to the system as an asset administrator.          User should only transfer asset which are currently allocated.</p>	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Click transfer button on any issued asset.</li> <li>2. User should able to select or enter employee ID in transfer to field.</li> <li>3. Current allocations should display corresponding to the entered employee ID.</li> <li>4. Click submit button to transfer asset</li> </ol>	
<b>Post conditions</b>	
The asset ownership should change to selected employee.	

*Table B.6 Use-Case Description for Asset Transfer*



### 7. Use-Case Description for Asset Clearance

<b>User-Case</b>	Clearance Asset
<b>Actors</b>	Asset Administrator
<b>Overview</b>	
Asset administrator should able to clear asset of a selected employee	
<b>Pre-conditions</b>	
User should login to the system as an asset administrator User should only transfer asset which are currently allocated.	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Click clearance button on any issued asset.</li> <li>2. Confirm message should pop up and should able to click confirm.</li> <li>3. Asset should clear from selected employee.</li> <li>4. User should able to view all cleared assets in clearance history.</li> </ol>	
<b>Post conditions</b>	
The asset ownership should revoke from the selected user.	

*Table B.7 Use-Case Description for Asset Clearance*

### 8. Use-Case Description for Add Document

<b>User-Case</b>	Add Document
<b>Actors</b>	Manager
<b>Overview</b>	
Manager should able to add new documents	
<b>Pre-conditions</b>	
User should login to the system as a manager. Maximum document down is five attachments.	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Click new document button in document management page.</li> <li>2. Enter all required fields and select document count.</li> <li>3. User should allow to upload files using file attachment fields.</li> <li>4. Click submit button to add new document.</li> </ol>	
<b>Post conditions</b>	
A new document should add with details.	

*Table B.8 Use-Case Description for Asset Clearance*

### 9. Use-Case Description for Manage Users

<b>User-Case</b>	Manage Users
<b>Actors</b>	Manager
<b>Overview</b>	
Manager should able to add a new user and manage	
<b>Pre-conditions</b>	
User should login to the system as a manager. The user who request a user account does not already possess an account.	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. The person who is in need to access the system requests an account from the manager.</li> <li>2. The status of the person requesting the account is checked by the manager.</li> <li>3. If the person does not already own an account, the account is created.</li> <li>4. The access level of the user must be confined to the user type assigned when creating the account</li> </ol>	
<b>Post conditions</b>	
Authorized personnel are granted access to the system.	

*Table B.9 Use-Case Description for Manager Users*

### 10. Use-Case Description for Generate Reports

<b>User-Case</b>	Manage Users
<b>Actors</b>	Manager
<b>Overview</b>	
Manager should able to get several types of reports.	
<b>Pre-conditions</b>	
User should login to the system as a manager and required data and input should have been fed into the system.	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. Select the relevant report to be generated under the reporting module.</li> <li>2. Input the necessary parameters.</li> <li>3. Click the generate button to generate the report.</li> </ol>	
<b>Post conditions</b>	
Generated details will be displayed in a printable format.	

*Table B.10 Use-Case Description for Generate Reports*

### 11. Use-Case Description for Login

<b>User-Case</b>	Login
<b>Actors</b>	System Users.
<b>Overview</b>	
Authorized users log into the system.	
<b>Pre-conditions</b>	
The user should possess an account and have the necessary privileges to login to the system.	
<b>Flow of events</b>	
<ol style="list-style-type: none"> <li>1. The user enters the username and password.</li> <li>2. User authentication takes place.</li> <li>3. If the entered values are invalid, the user will be redirected to the login page with an error message. Rather, if the entered values are valid, the user will be redirected to dashboard.</li> </ol>	
<b>Post conditions</b>	
Only authorized users can have access to the system whereas access is restricted to others.	

Table B.11 Use-Case Description for Login

## SEQUENCE DIAGRAMS

### Sequence Diagram for Login

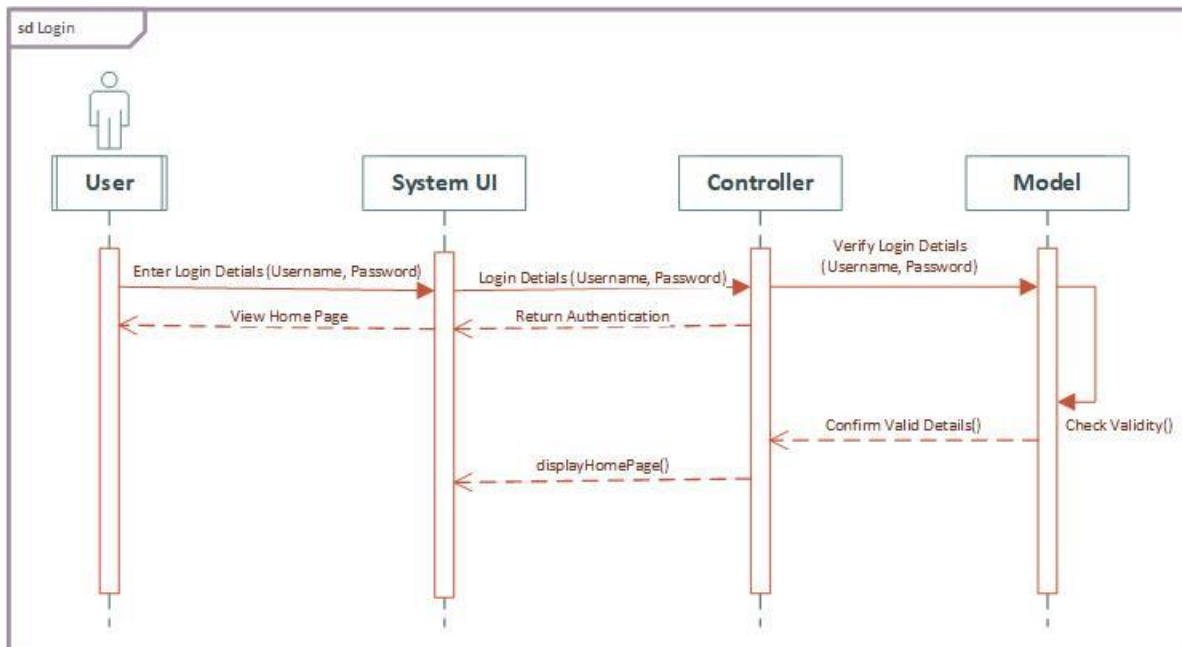


Figure B.1 Login Sequence Diagram

## Sequence Diagram for Asset Transfer

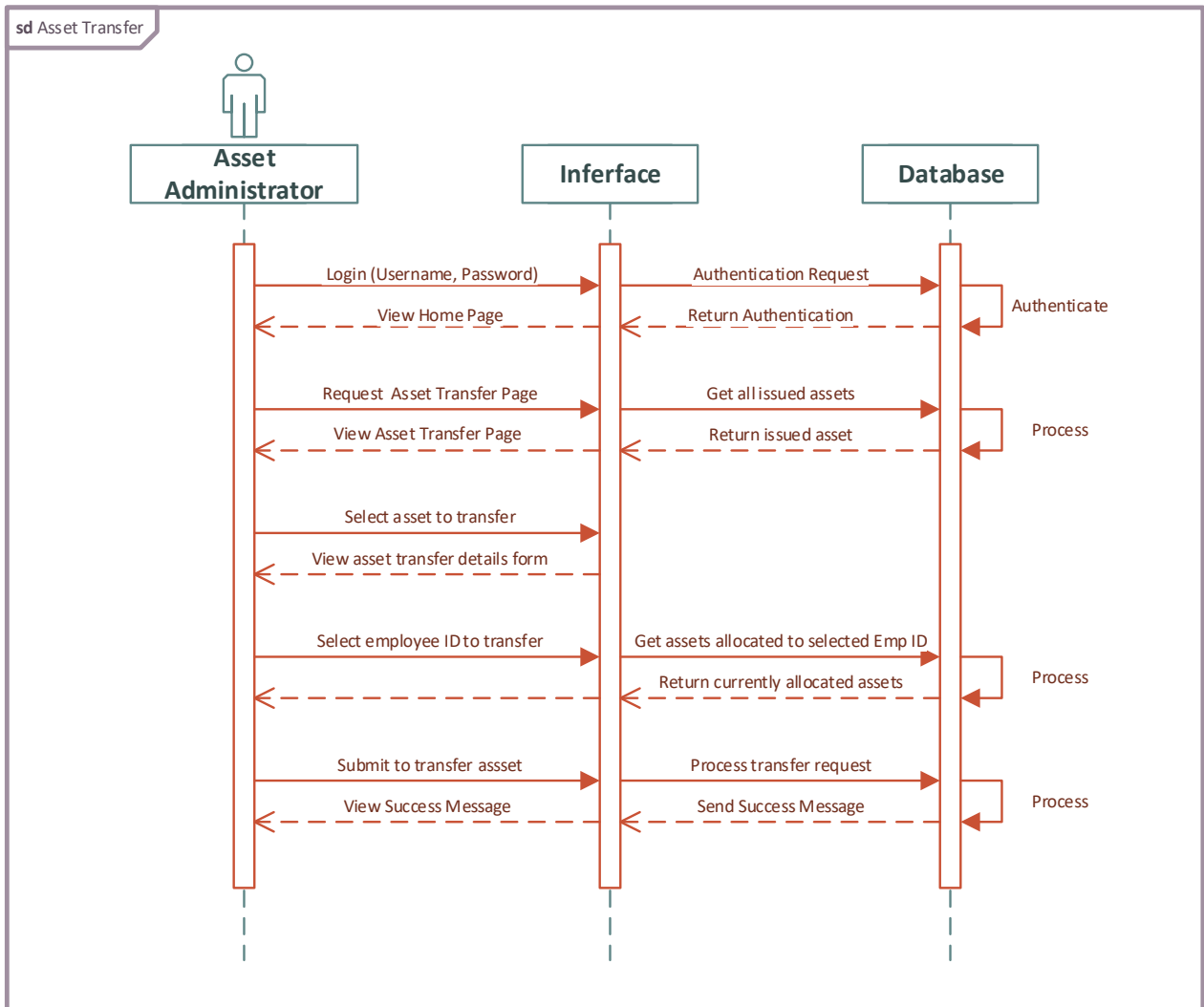


Figure B.2 Asset Transfer Sequence Diagram

## ACTIVITY DIAGRAMS

### Activity Diagram for Asset Clearance

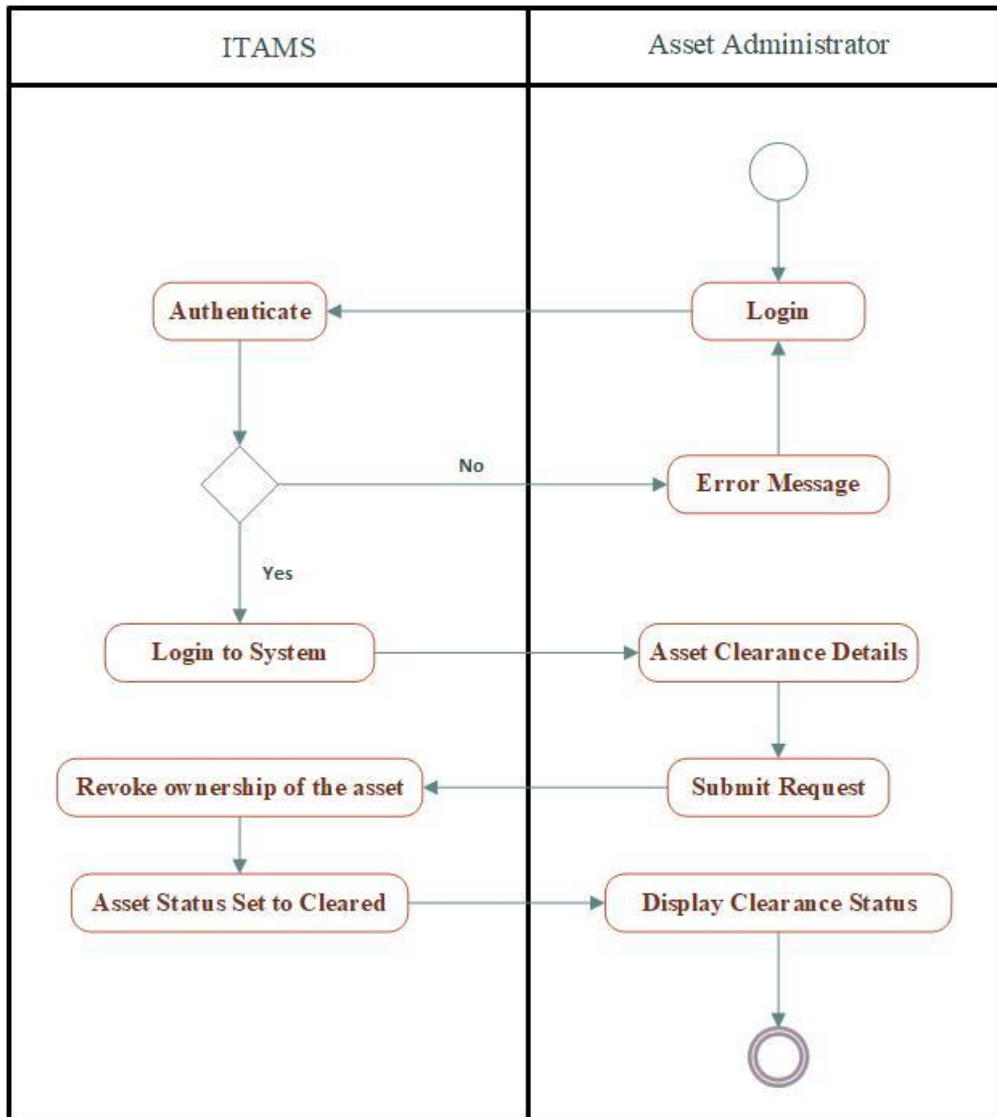


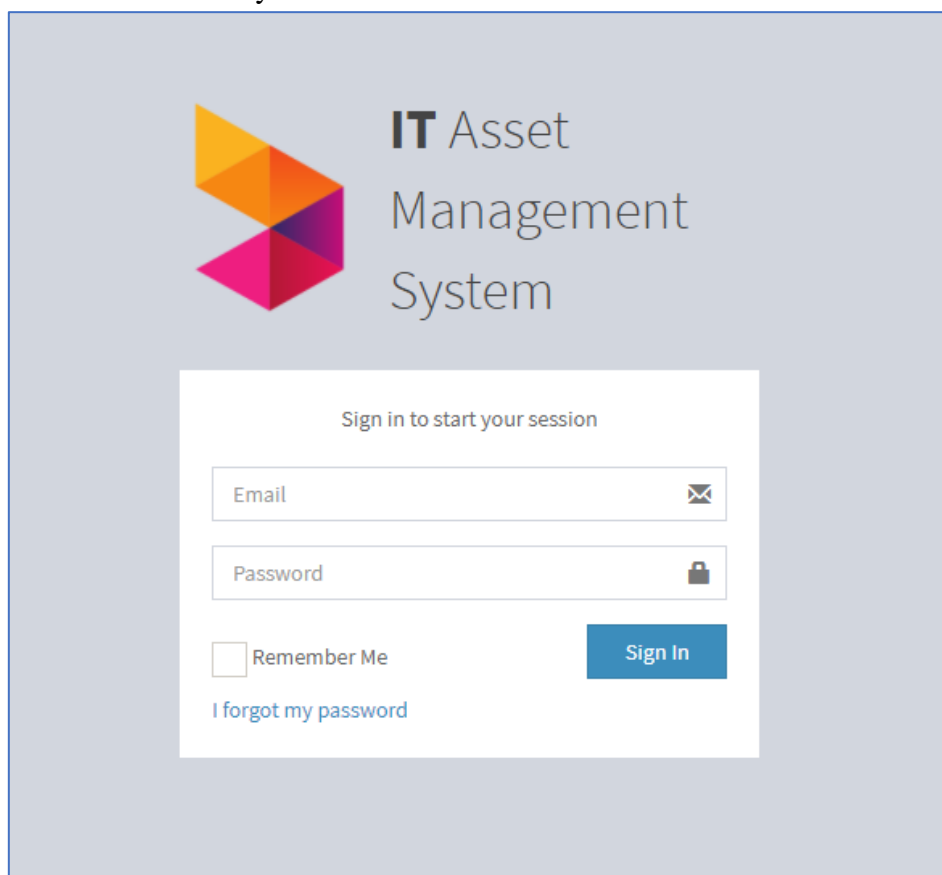
Figure B.3 Asset Clearance Activity Diagram

# APPENDIX C - USER DOCUMENTATION

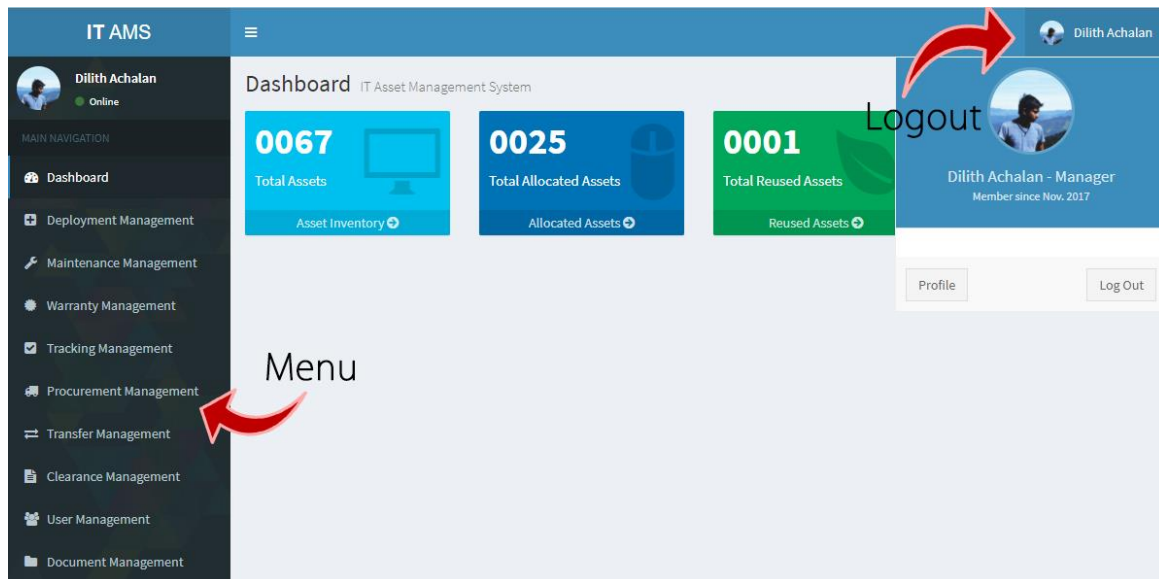
This document is a reference to the users of IT Asset Management System and provides an overview picture on how to use the main functions in the system. The main expectation of this document is to guide the user in the general use of Asset Management System without the need to emphasize on extensive training program.

## LOGIN TO THE SYSTEM

Authorized users can access the system by providing a valid username and a password via the login page shown in Figure C.1. Once the credentials entered by the user is authenticated and found to be valid, the user will be redirected to the dashboard as shown in Figure C.2 which comprises of links to all the system modules.



*Figure C.1 Login Page*



*Figure C.2 Dashboard Page*

- Modules can be accessed by clicking the relevant menu on left-side corner.
- The users can logout from the system by clicking on the user name on the top right-side corner of the window and click logout button given. The username and logout button is displayed in every page of the system.
- The profile image of the logged in user is displayed for further confirmation of the login.
- The profile of the logged in user can be viewed by clicking on the name or the profile image and then clicking the profile button. Once the profile information page is displayed, the user can change the password by clicking the change password link given under the profile image.

## **GETTING FAMILIAR WITH THE SYSTEM**

Once a relevant module is selected from the dashboard that is illustrated in Figure C.2, the user will be redirected to the module homepage which consists of the various functions that can be performed under the relevant module. The following Figure C.3 depicts a module homepage.

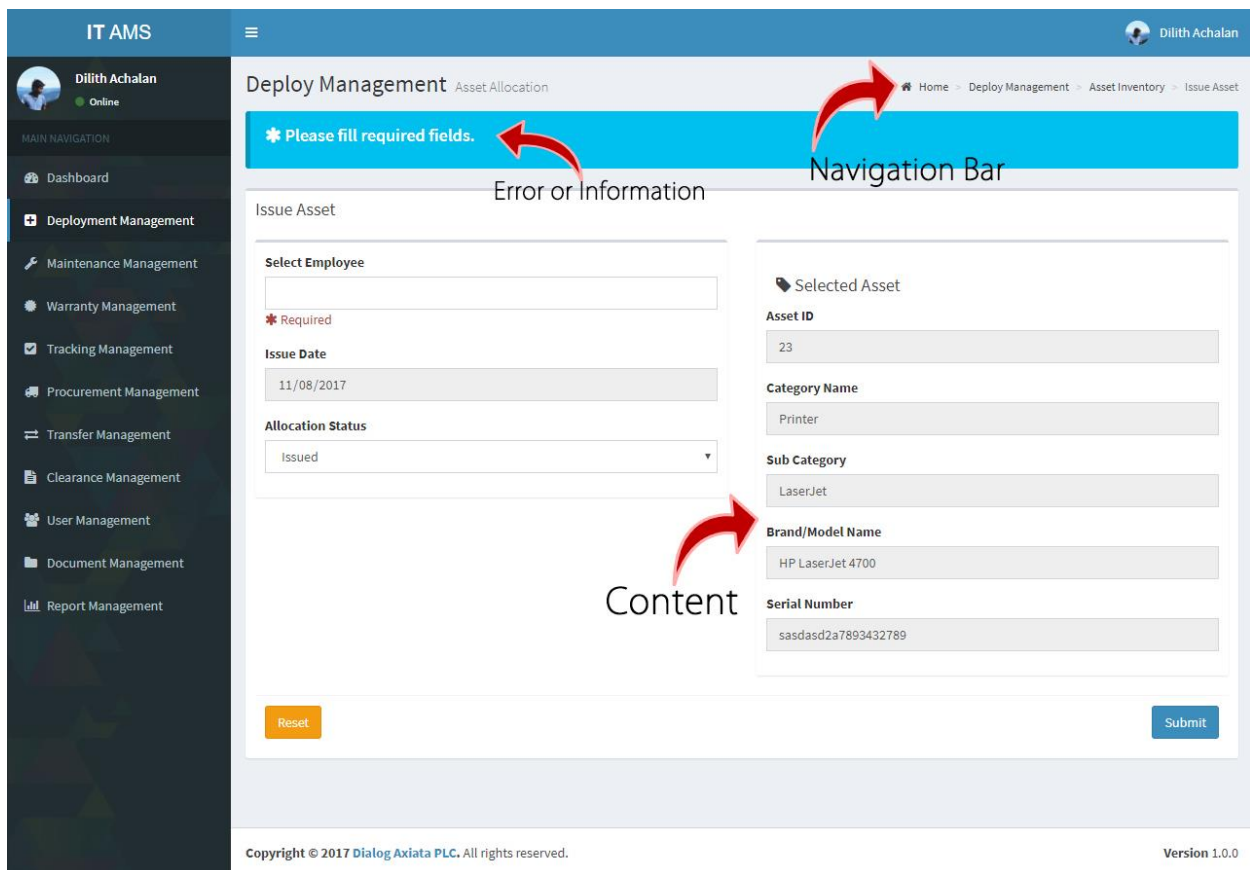


Figure C.3 Module Functional Page

- Profile Area – It identifies and displays the name of the logged in user and provide logout button to logout from the system.
- Menu – Hold each of a module. Once click each module, the system direct user to several functions depending on the module function.
- Content – The forms, results and other information is loaded into this area.

## GENERAL GUIDELINES IN USING THE SYSTEM

- Mandatory fields are marked as required with a red asterisk mark.

Warranty is available

**Purchase Date**

11/08/2017

\* Required

**Warranty Period**

Select Warranty Period ▼

\* Required

Figure C.4 Required Fields



Maintenance History

Search:

Maintenance ID	Maintenance Type	Brand/Model	Serial Number	Description	
1	Repair	HP EliteBook 840	3164413800441	HDD Replaced Due to Bad Sector	<a href="#">View</a> <a href="#">Update</a>
2	Upgrade	HP DC 5700	SGH458U1452	Ram Upgrade - 4GB	<a href="#">View</a> <a href="#">Update</a>
3	Repair	Lenovo G50	TA7FTN1W	Battery replaced	<a href="#">View</a> <a href="#">Update</a>
4	Repair	Dell 17 Inch Wide Screen	YUATU1S	Repair by Vendor - Thakral IT	<a href="#">View</a> <a href="#">Update</a>

Previous 1 Next

*Figure C.5 Common Guidelines*

- Text boxes and text areas allows to enter text and/or numeric data.
- Drop-down lists and radio buttons allows to choose just one option of a series of choices whereas checkboxes are used to select more options of several choices.
- Various buttons allow to instruct the system to carry out specific tasks.
- Clicking the edit button with pen icon displayed at the end of the record set, lets to edit a record by redirecting to the update form filled with the values stored in the database with an update button.
- Clicking the view button with file icon at the end of the record set lets to view a record.

Please refer the CD-ROM for the complete set of this documentation since only a limited version is given here due to space constraints.

# APPENDIX D - MANAGEMENT REPORTS

Management reports are essential for any system as it helps to take vital decisions in managing all aspects of an organization. Managers gain insight on specific areas and able to evaluate the performance. IT Asset Management System includes various kinds of reports that aids in the decision-making process. Moreover, charts are used in appropriate places to provide rich interactivity and better understanding.

## SUMMARY REPORTS

### Asset Meter

The following figure D.1 illustrates overall asset allocation status. The percentage of utilized assets vs remaining. This graph gives insight on asset reorder level to prevent inadequate assets.

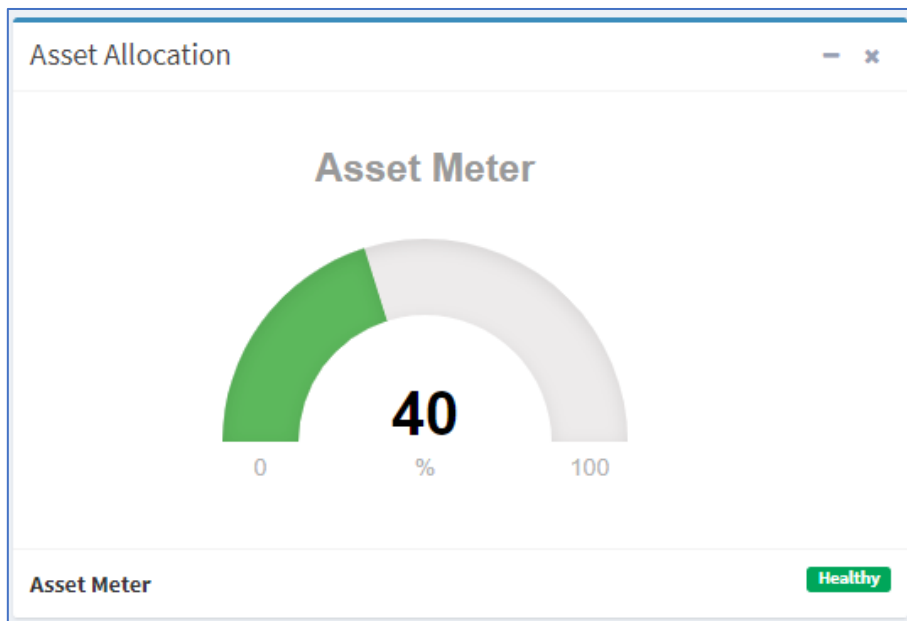


Figure D.1 Asset Meter

### Top 10 fast moving Assets

The following figure D.2 illustrates the breakdown of top 10 fast moving assets within the period of last 30 days. This graph depicts insight to increase the reorder level of a specific asset category.

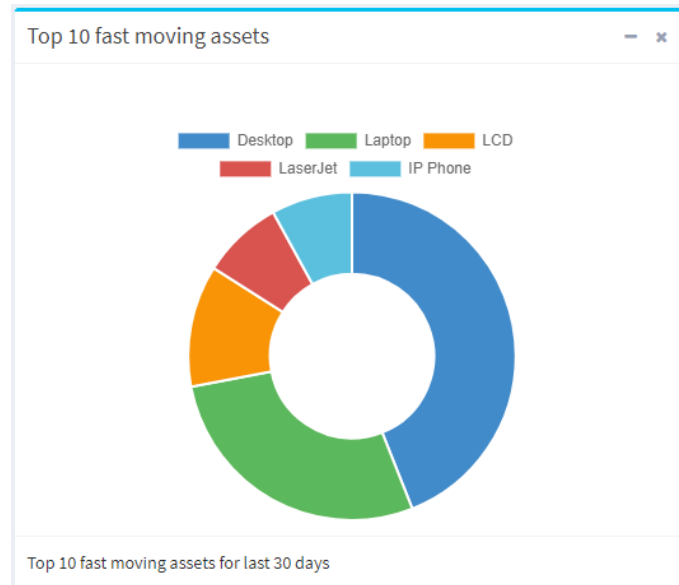


Figure D.2 Top 10 Fast moving asset for last 30 days.

The following figure D.3 illustrates the asset request trend throughout the year.

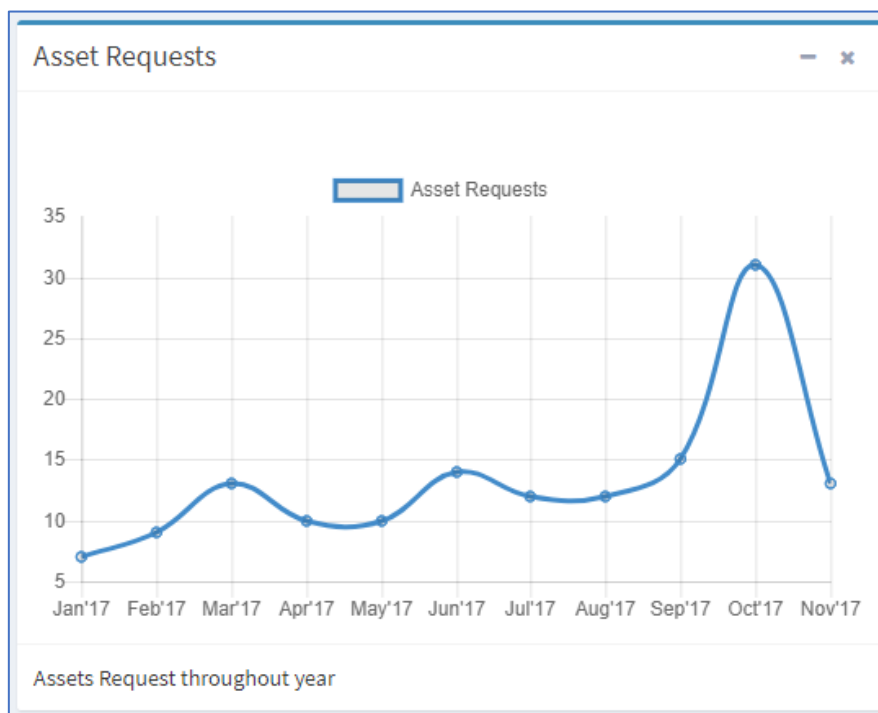


Figure D.3 Asset Request throughout the year.

The following figure D.4 illustrates top 10 multiple desktop and laptop owners.

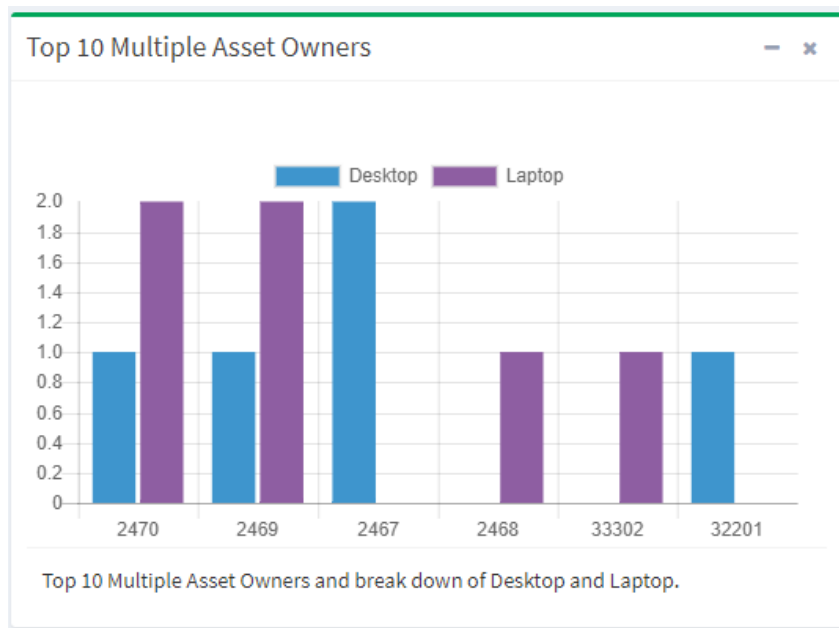


Figure D.4 Top 10 Top 10 Multiple Asset Owners

## DETAILED REPORT

The following figure D.5 illustrates an instance of a detailed report which provides comprehensive information that shows asset request for a given period.

### IT Asset Management System | Asset Request Report

PO ID	Category	Brand/Model	Quantity	Request Date	Description	Status
P00001	Desktop	Dell Optiplex 3040	14	2017-08-08	Regional PC change	Approved
P00001	LCD	HP 17 Inch	14	2017-08-15	Regional PC change	Approved
P00002	Laptop	HP ProBook 4520s	4	2017-09-05	Laptop Change	Approved
P00003	Laptop	Dell Latitude E7270	8	2017-10-26	Laptop Change	Approved
P00003	LCD	Dell 19 Inch	4	2017-10-31	Senior Mgmt Extra Display	Approved
P00003	IP Phone	Cisco 7960	6	2017-11-22	IP phone new batch	Approved
P00004	LaserJet - BW	HP LaserJet 3100n	16	2017-11-26	Office Printers	Pending

Figure D.5 Detailed Report for Asset requests for given period

The following figure D.6 illustrates an instance of a detailed report which provides comprehensive information that shows asset allocation for a given period.

### IT Asset Management System | Asset Allocation Report

Allocation ID	Employee Name	Category	Brand/Model	Serial Number	Asset Status
3	Amali Nanayakkara	PC	Dell Optiplex 3040	76HY262	Issued
4	Azwan Khan	PC	Dell Optiplex 3040	4B5XD12	Issued
5	Thimali Amarasekara	Monitor	HP 17 Inch	CND7371T1G	Issued
6	Janith Peiris	Monitor	HP 17 Inch	CNC812PJY1	Issued
7	Amali Nanayakkara	Monitor	HP 17 Inch	CND705438C	Issued
8	Janith Peiris	Monitor	HP 17 Inch	CND7243XH1	Issued
9	Amali Nanayakkara	PC	HP ProBook 4520s	FCH11218TP3	Issued
11	Janith Peiris	PC	HP ProBook 4520s	FCH112195QR	Issued
12	Amali Nanayakkara	PC	HP ProBook 4520s	FCH1215FWKG	Issued
13	Amali Nanayakkara	PC	Dell Optiplex 3040	F2QJC12	Issued
14	Kusal Perera	PC	Dell Optiplex 3040	JK34042	Issued
15	Azwan Khan	PC	HP ProBook 4520s	FCH1148AE9B	Issued
16	Azwan Khan	PC	HP ProBook 4520s	FCH1215FWYX	Issued
17	Imal Meewala	PC	HP ProBook 4520s	FCH1215F824	Issued
18	Imal Meewala	Monitor	HP 17 Inch	CNC025Q92J	Issued
19	Kusal Perera	PC	Dell Optiplex 3040	78FYF72	Issued
20	Roshan Indika	PC	HP EliteBook 8470p	DFGDF4534	Issued

Figure D.6 Detailed Report for Asset Allocation for given period

The following figure D.7 illustrates an instance of a detailed report which provides comprehensive information that shows asset maintenance for a given period.

## IT Asset Management System | Asset Maintenance Report

Maintenance ID	Maintenance Type	Category	Sub-Category	Brand/Model	Serial Number	Description	Maintenance Date
1	Repair	PC	Desktop	Dell Optiplex 3040	GK34042	HDD failure	2017-12-02
2	Upgrade	PC	Desktop	Dell Optiplex 3040	GK34042	RAM upgrade	2017-12-02
3	Repair	Telephone Extension	IP Phone	Cisco 7960	FCH1215FW8R	Handset changed	2017-12-02
4	Repair	PC	Laptop	HP ProBook 4520s	FCH1215FWYX	Motherboard Repaired	2017-12-02
5	Repair	Telephone Extension	IP Phone	Cisco 7960	FCH11179AEW	Handset changed	2017-12-02
6	Repair	PC	Desktop	Dell Optiplex 3040	GK34042	repair	2017-12-02

Figure D.7 Detailed Report for Asset Maintenance Report for given period

The following figure D.8 illustrates an instance of a detailed report which provides comprehensive information that shows asset clearance for a given period.

## IT Asset Management System | Clearance Report

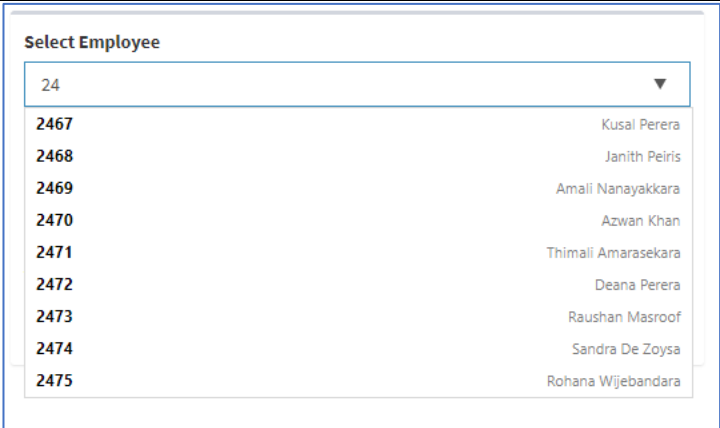
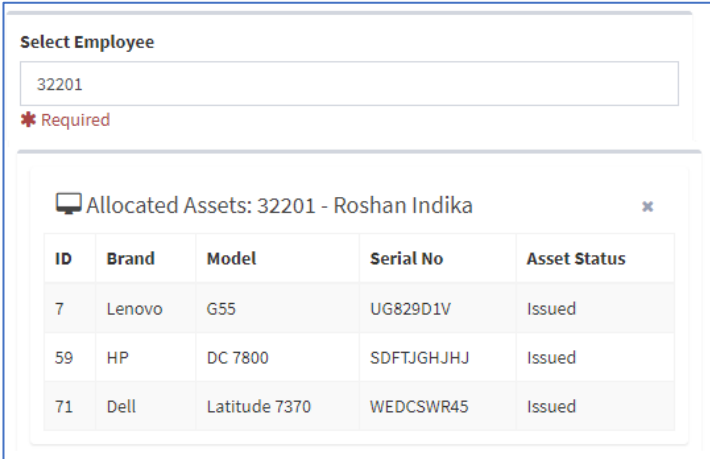
Clearance ID	Brand/Model	Serial Number	Status	Clearance Date
1	Dell Optiplex 3040	78FYF72	Reused	2017-12-02
2	HP 17 Inch	CND705438C	Reused	2017-12-02
3	Dell Optiplex 3040	GK34042	Reused	2017-12-02
4	Dell Optiplex 3040	76HY262	Cleared	2017-12-02
5	HP 17 Inch	CND7371T1G	Cleared	2017-12-02
6	HP 17 Inch	CND705438C	Cleared	2017-12-02

Figure D.8 Detailed Report for Asset Clearance Report for given period

# APPENDIX E - TEST RESULTS

Based on the test plan given in Chapter 5 – Evaluation, test results were derived by the application of test cases. The following test cases and the resulting screenshots are given below to give an overview understanding of how the testing process was carried out in IT Asset Management System.

## Deployment Management Module

ID	Actual Output	Status																				
3	<p>The user should able to select employee ID or type part of the employee ID. List of employees corresponding to employee ID should display</p>  <p>The screenshot shows a 'Select Employee' dropdown menu. The search input contains '24'. The dropdown list displays the following employees:</p> <table border="1"> <thead> <tr> <th>Employee ID</th> <th>Employee Name</th> </tr> </thead> <tbody> <tr><td>2467</td><td>Kusal Perera</td></tr> <tr><td>2468</td><td>Janith Peiris</td></tr> <tr><td>2469</td><td>Amali Nanayakkara</td></tr> <tr><td>2470</td><td>Azwan Khan</td></tr> <tr><td>2471</td><td>Thimali Amarasekara</td></tr> <tr><td>2472</td><td>Deana Perera</td></tr> <tr><td>2473</td><td>Raushan Masroof</td></tr> <tr><td>2474</td><td>Sandra De Zoysa</td></tr> <tr><td>2475</td><td>Rohana Wijebandara</td></tr> </tbody> </table>	Employee ID	Employee Name	2467	Kusal Perera	2468	Janith Peiris	2469	Amali Nanayakkara	2470	Azwan Khan	2471	Thimali Amarasekara	2472	Deana Perera	2473	Raushan Masroof	2474	Sandra De Zoysa	2475	Rohana Wijebandara	✓
Employee ID	Employee Name																					
2467	Kusal Perera																					
2468	Janith Peiris																					
2469	Amali Nanayakkara																					
2470	Azwan Khan																					
2471	Thimali Amarasekara																					
2472	Deana Perera																					
2473	Raushan Masroof																					
2474	Sandra De Zoysa																					
2475	Rohana Wijebandara																					
4	<p>A list of allocated assets to a selected employee should display.</p>  <p>The screenshot shows the 'Select Employee' form with '32201' entered in the search field. Below the form, a modal window titled 'Allocated Assets: 32201 - Roshan Indika' displays a table of assets:</p> <table border="1"> <thead> <tr> <th>ID</th> <th>Brand</th> <th>Model</th> <th>Serial No</th> <th>Asset Status</th> </tr> </thead> <tbody> <tr><td>7</td><td>Lenovo</td><td>G55</td><td>UG829D1V</td><td>Issued</td></tr> <tr><td>59</td><td>HP</td><td>DC 7800</td><td>SDFTJGHJHJ</td><td>Issued</td></tr> <tr><td>71</td><td>Dell</td><td>Latitude 7370</td><td>WEDCSWR45</td><td>Issued</td></tr> </tbody> </table>	ID	Brand	Model	Serial No	Asset Status	7	Lenovo	G55	UG829D1V	Issued	59	HP	DC 7800	SDFTJGHJHJ	Issued	71	Dell	Latitude 7370	WEDCSWR45	Issued	✓
ID	Brand	Model	Serial No	Asset Status																		
7	Lenovo	G55	UG829D1V	Issued																		
59	HP	DC 7800	SDFTJGHJHJ	Issued																		
71	Dell	Latitude 7370	WEDCSWR45	Issued																		
10	<p>Confirmation message should pop up and the asset should be set as either continued or discontinued depending on the current status, once click confirm.</p>	✓																				

<div style="border: 1px solid black; padding: 5px;"> <p>Confirm Action <span style="float: right;">x</span></p> <p>Do you want to proceed?</p> <p style="text-align: right;"> <input type="button" value="Cancel"/> <input type="button" value="Confirm"/> </p> </div>		
VDI	Continued	<input type="button" value="View"/> <input type="button" value="Update"/> <input type="button" value="Discontinued"/>
CRT	Discontinued	<input type="button" value="View"/> <input type="button" value="Update"/> <input type="button" value="Continued"/>
LCD	Continued	<input type="button" value="View"/> <input type="button" value="Update"/> <input type="button" value="Discontinued"/>

Table E.1 Test Results for Deployment Management Module

### Maintenance Management Module

ID	Actual Output	Status
3	Display appropriate error messages and asset maintenance details not added to the system.	
	<div style="border: 1px solid black; padding: 10px;"> <div style="background-color: #00aaff; color: white; padding: 5px; margin-bottom: 10px;"> <p><b>* Please fill required fields.</b>  <b>Select Maintenance Type</b></p> </div> <div style="border: 1px solid #ccc; padding: 10px;"> <p>New Asset Maintenance</p> <hr/> <p><b>Select Asset</b></p> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">1</div> <p><b>* Required</b></p> <p><b>Maintenance Type</b></p> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">Select Maintenance Type ▼</div> <p><b>* Required</b></p> </div> </div>	✓

Table E.2 Test Results for Maintenance Management Module



## Procurement Management Module

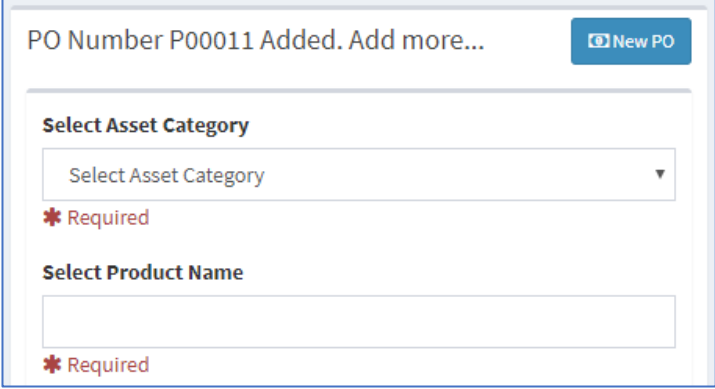
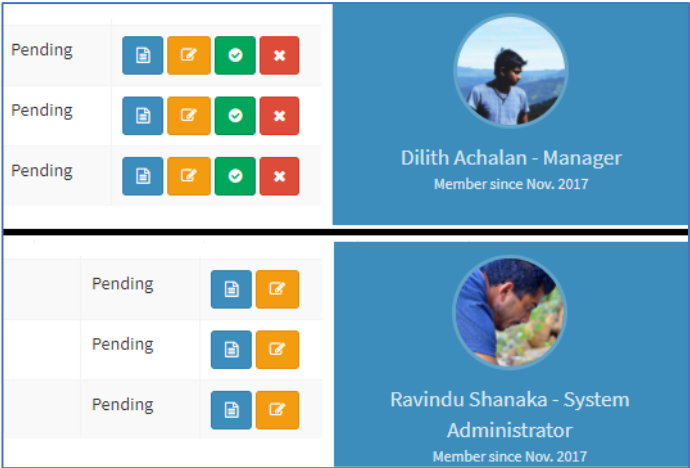
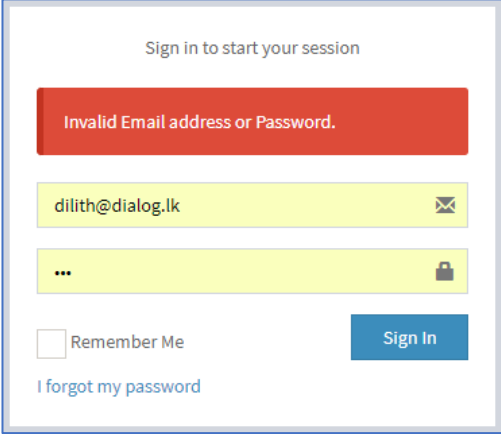
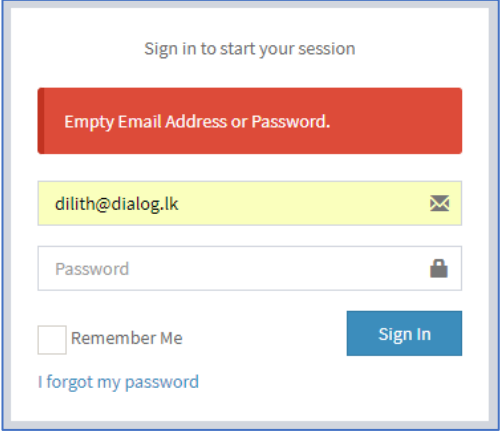
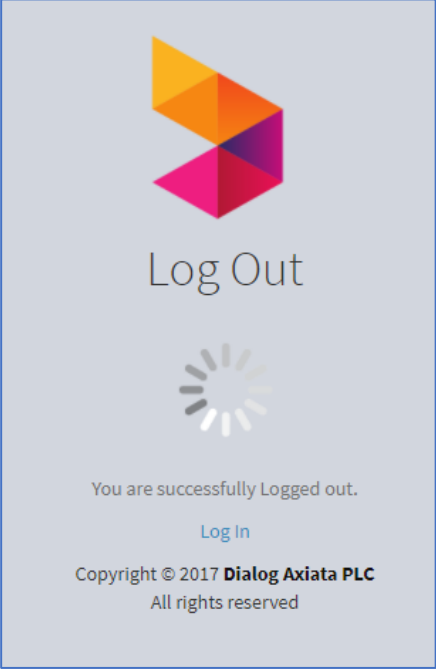
ID	Actual Output	Status
4	<p>Assets should add to the same purchase order ID</p> 	✓
9	<p>The manager is only able to approve the request. For any other users approve button must not visible</p> 	✓

Table E.3 Test Results for Procurement Management Module

## Common Functions

ID	Actual Output	Status
1	<p>An error message must be displayed indicating the error.</p>  <p>The screenshot shows a login form titled "Sign in to start your session". A red error message box at the top reads "Invalid Email address or Password.". Below it, the email field contains "dilith@dialog.lk" and the password field is masked with "..." and a lock icon. There is a "Remember Me" checkbox, a "Sign In" button, and a link for "I forgot my password".</p>	✓
3	<p>An error message is displayed indicating the fields are required.</p>  <p>The screenshot shows a login form titled "Sign in to start your session". A red error message box at the top reads "Empty Email Address or Password.". Below it, the email field contains "dilith@dialog.lk" and the password field is empty with a lock icon. There is a "Remember Me" checkbox, a "Sign In" button, and a link for "I forgot my password".</p>	✓
4	<p>User logged out and redirecting to the login page.</p>	✓

			
--	--	--	--

*Table E.4 Test Results for Common Functions*

## APPENDIX F - CODE LISTING

This section consists of an overview of the codes used in the development of this system. Due to space constraints, only the important code snippets are represented here apart from those mentioned in Chapter 4 - Implementation.

Please refer the CD-ROM for the complete system source code.

### AJAX OPERATIONS

The following code segment shows the AJAX code for checking whether the email enter while adding a new user is already an invalid, existing one or not.

```
<script type="text/javascript">
    function checkEmail(str)
    {
        var xmlhttp;
        if (str == "")
        {
            document.getElementById("txtHint").innerHTML = "";
            return;
        }
        if (window.XMLHttpRequest)
        { // code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp = new XMLHttpRequest();
        } else
        { // code for IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        xmlhttp.onreadystatechange = function ()
        {
            if (xmlhttp.readyState == 4 && xmlhttp.status == 200)
            {
                document.getElementById("emailAvailable").innerHTML = xmlhttp.responseText;
            }
        }
        xmlhttp.open("GET", "getemail.php?q=" + str, true);
        xmlhttp.send();
    }
}
```

Below is the function used by above AJAX function.

```
// Validate if email is existing
public function checkEmail($email){
    $con=$GLOBALS['con'];
    $sql = "SELECT * FROM tbl_login WHERE email='$email'";
    $result = $con->query($sql);
    $noOfResult = $result->num_rows;
    return $noOfResult;
}
```

The following code fragment depicts the graphical representation of availability of email address.

```
<?php
include '../common/dbconnection.php';
$obj=new dbconnection();
$con=$obj->connection();

//Include user model
include '../model/usermodel.php';

$email = $_GET['q'];
$emailPattern = "/^([a-zA-Z0-9_\\.\\-])+\\@(((a-zA-Z0-9\\-]+\\.)+([a-zA-Z]{2,6})+)$/";

//Create a object using class
$userobject = new user();

if(preg_match($emailPattern,$email)){
    //Call function using the object and get result 0 or 1
    $result = $userobject->checkEmail($email);
    if($result == 0){
        // If Email address is available to use
        echo "<span class='label bg-green'>Email is available to use.</span>";
    }else{
        // If Email address is already in use
        echo "<span class='label bg-orange'>Email is already taken.</span>";
    }
}else{
    // If Email address in invalid
    echo "<span class='label bg-red'>Email is invalid.</span>";
}
?>
```

The following code segment shows the AJAX code used to get category details using category ID.

```
<script type="text/javascript">
$('#category_select').change(function(){
    $.ajax({
        type:'get',
        url:'getAjaxCategoryByID.php',
        data: 'category_id='+ $(this).val(),
        success: function(value){
            var data = value.split(",");
            $('#category_id').val(data[0]);
            $('#category_type').val(data[1]);
            $('#category_name').val(data[2]);
            $('#category_sub1').val(data[3]);
            $('#category_sub2').val(data[4]);
        }
    });
});
</script>
```

## GENERATE PO NUMBER

Following code segment shows how PO number generates, by incrementing last PO number.

```
<?php
session_start();
// To start the Session
include '../common/session.php';

// Database connection
include '../common/dbconnection.php';
$obj = new dbconnection();
$con = $obj->connection();

//Include procurement model
include '../model/procurementmodel.php';
$procurementObj = new procurement();

// Retrieve last PO number
$resultLastPONumber = $procurementObj->checkLastPONumber();
$rowLastPONumber = $resultLastPONumber->fetch_array();

// Get last PO number created
$last_po_number = $rowLastPONumber['last_po_number'];

// Incrementing PO number
$po_number = $last_po_number + 1;

// Set PO number to session so that PO number will not change when page redirects
$_SESSION['po_number'] = "P".sprintf("%05d", $po_number);
//Redirect to New Procurement request page
header("location: newpurchase.php");
?>
```

## REPORT GENERATION

The following code segment shows the how data fed into charts, using JavaScript.

```
<script>

// Doughnut - Fast moving items
new Chart(document.getElementById("doughnutFastMovingAssets"), {
  type: 'doughnut',
  data: {
    labels: [ <?php foreach ($arrCategory as $k => $v) { ?>
      "<?php echo $v; ?>",
      <?php } ?>
    ],
    datasets: [
      {
        label: "Count",
        backgroundColor: ['#428bca', '#5cb85c', '#F89406', '#d9534f', '#5bc0de'],
        data: [
          <?php foreach ($arrCategoryCount as $k => $v) { ?>
            "<?php echo $v; ?>",
            <?php } ?>
          ]
        }
      ]
    ],
    options: {
      title: {
        display: true,
        responsive: true,
        maintainAspectRatio: false,
        text: '',
      }
    }
  }
});
// ---end Doughnut - Fast moving items
```

## REPORT GENERATION

The following code segment shows how documents are being upload to the system.

```
<?php
include '../common/dbconnection.php';
$obj = new dbconnection();
$con = $obj->connection();

//Include document model
include '../model/documentmodel.php';
$documentObj = new document();
$fileObj = new file();

$action = strtolower($_REQUEST['action']);

switch ($action) {
    case "add":

        // Add new documents
        $documentArr = $_POST;
        $doc_attach=$_FILES['doc_attach'] ['name'];
        $doc_attach_tmp=$_FILES['doc_attach'] ['tmp_name'];
        print_r($documentArr);
        $doc_id = $documentObj->addDocument($documentArr);

        $noOfDoc = count($doc_attach);
        if ($noOfDoc != 0) {
            foreach ($doc_attach as $k => $e) {
                if ($e != "") { // To check existence of file
                    $doc_name = $doc_attach[$k];
                    $doc_name_tmp = $doc_attach_tmp[$k];
                    // Send Docuemnt details to Database
                    $fileObj->uploadDocument($doc_name, $doc_id);
                    $path = "../dist/documents/" . $doc_name;
                    // Upload files
                    move_uploaded_file($doc_name_tmp, $path);
                }
            }
        }
        // Redirection
        header("Location:../view/document.php");
        break;
}
```



# APPENDIX G - CLIENT CERTIFICATE



06<sup>th</sup> November 2017

Project Examination Board,  
University of Colombo School of Computing,  
No: 35, Reid Avenue,  
Colombo 7,  
Sri Lanka.

Dear Sir/Madam,


## LETTER OF CERTIFICATION

This is to certify that Mr. G. D. Achalan has successfully developed a Web Based **IT Asset Management System** for Dialog Axiata PLC. Project was undertaken by him as a partial fulfillment of a requirement for the Bachelor of Information Technology Degree.

The system has proven to be successful and has been accepted as a viable solution to meet the needs of an IT Asset Management System. I'm pleased to certify that the developed system can fulfill the requirements of Group IT.

This certification letter is issued at the request of Mr. G. D. Achalan.

Your faithfully,

  
.....  
Sudesh Benthararachchi,  
Unit Manager - IT Infrastructure Planning & Operations,  
Dialog Axiata PLC.

**Dialog Axiata PLC**  
475, Union Place,  
Colombo 02.

Internal