

**E-ECLAB INVENTORY SYSTEM FOR FKP
(SOFTWARE DEVELOPMENT)**

NURSAFINAZ BINTI JAAFAR

UNIVERSITI MALAYSIA PAHANG

E-ECLAB INVENTORY SYSTEM FOR FKP
(SOFTWARE DEVELOPMENT)

NURSAFINAZ BINTI JAAFAR

Report submitted in partial fulfilment of the requirements
for the award of Bachelor of Mechatronic Engineering

Faculty of Manufacturing Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2013

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: E-ECLAB INVENTORY SYSTEM FOR FKP (SOFTWARE DEVELOPMENT).

SESI PENGAJIAN: 2012/2013

Saya, **NURSAFINAZ BINTI JAAFAR(890821-14-5088)**
(HURUF BESAR)

mengaku membenarkan tesis Projek Tahun Akhir ini disimpan di perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis ini adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh:

(TANDATANGAN PENULIS)

Alamat Tetap:
**No. 23 Jalan 2/7B,
Taman Bukit Teratai,
56100 Kuala Lumpur**

Tarikh: 19 JUN 2013

(TANDATANGAN PENYELIA)

**MR ISMAIL BIN
MOHD KHAIRUDDIN**
(Nama Penyelia)

Tarikh: 19 JUN 2013

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

ACKNOWLEDGMENT

Alhamdulillah, I would like to acknowledge and extend my heartfelt gratitude to my supportive supervisor Mr Ismail bin Mohd Khairuddin . The supervision and support that he gave truly help the progression and smoothness of my final year project (FYP). The vital support and assistance are much indeed appreciated.

Great deals appreciated goes to the contribution of my faculty – Faculty of Manufacturing Engineering. I am also would like to thankful the Final Year Project Coordinator of Faculty of Manufacturing Engineering, Prof. Madya. Ajisman Apen, and all staff in the Faculty of Manufacturing Engineering who's always being patient in helping completing this project.

Last but not the least, my beloved parents: Jaafar bin Sharif and Salehah binti Ahmad, family: Nursafarina, Saifullizan, Nurshahirah, Amanina, Syakir and friend: Radhiah and Noor Zalika for always offering their endless support and preciously, for always believing in me and the one above all of us, ALLAH S.W.T: The one and only God for answering my prayers for giving me the strength to plod on, thank you so much Dear Lord.

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project report and in our opinion this project is satisfactory in terms of scope and quality for the award of the degree of Bachelor of Mechatronic Engineering

Signature :

Name of Supervisor : MR ISMAIL BIN MOHD KHAIRUDDIN

Position :

Date :

Signature :

Name of Panel : MR WAN HASBULLAH BIN MOHD ISA

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature :
Name : NURSAFINAZ BINTI JAAFAR
ID Number : FB10042
Date :

*Dedicated to my beloved father, Jaafar bin Sharif,
mother, Salehah binti Ahmad , brothers and sisters,
and my supervisor, Mr Ismail bin Mohd Khairuddin.*

ABSTRACT

This thesis describes about the Electronic Component Laboratory services or E-ECLAB. E-ECLAB is a one of the real time management of inventory component in catalog which is consistent with the aim of this project to facilitate inventory management system in term of requisition of component in laboratory. Where E-ECLAB support updating inventory information for all components, monitoring inventory depletion and importing and exporting inventory information to and from external system of record. The interfaces are design by using Visual Basic (VB), where VB a productively building type-safe and object-oriented applications while the database using Microsoft Office Access. There are several analyses has been made by identify the requirement specification, design the interface, implementation or debug the code, testing it with by releasing the application, make the installation after publish the testing part had running smoothly and lastly maintaining and enhancing the application. These applications are very friendly user, where user can track their requisition by scan the bar code given. The administrator that handles this inventory have easy way to update stock and approval the requisition sent by user. With this application, the inventory in laboratory will become more systematic and more accurate.

ABSTRAK

Tesis ini menerangkan mengenai perkhidmatan yang disediakan untuk makmal komponen elektronik atau dikenali sebagai E-ECLAB. E-ECLAB adalah salah satu aplikasi yang menggunakan masa sebenar bagi pengurusan komponen inventori dalam katalog dimana konsisten dengan matlamat projek ini, untuk memudahkan system pengurusan inventori dari segi permintaan komponen di dalam makmal. Dimana kegunaan E-ECLAB ialah mengemaskini segala maklumat inventori untuk semua komponen. Selain itu aplikasi ini juga digunakan memantau kekurangan yang terdapat didalam inventori, kegunaan seterusnya digunakan untuk mengimport dan mengeksport segala maklumat inventori dari luar rekod dan dalam rekod yang terdapat didalam sistem. Aplikasi ini direka dengan menggunakan '*Visual Basic*' (VB), dimana VB adalah salah satu aplikasi yang produktif membina aplikasi jenis-selamat dan berorientasikan objek. Sementara itu bagi penyimpanan data di dalam aplikasi ini menggunakan '*Mircosoft Office Access*'. Terdapat beberapa analisis telah dibuat untuk mengenal pasti spesifikasi keperluan, reka bentuk aplikasi, pelaksanaan atau '*debug*' kod, menguji aplikasi dengan '*release*' aplikasi tersebut, membuat pemasangan selepas menerbitkan sebahagian aplikasi yang telah berjalan dengan lancar dan akhirnya mengekalkan dan menaiktaraf aplikasi tersebut dari semasa ke semasa. Dengan penggunaan aplikasi yang sangat mesra kepada pengguna, di mana pengguna dapat mengesan permintaan mereka dengan mengimbas kod bar yang diberikan semasa permohonan dibuat. Penyelaras yang mengendalikan inventori ini mempunyai cara mudah untuk mengemaskini stok dan kelulusan tuntutan yang dihantar oleh pengguna. Dengan aplikasi ini, inventori di makmal komponen elektronik akan menjadi lebih sistematik dan lebih tepat.

TABLE OF CONTENTS

	Page
SUPERVISOR’S DECLARATION	ii
STUDENT DECLARATION	iii
DEDICATION	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
ABSTRAK	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Overview	1
1.2 Background of Study	1
1.3 Problem Statement	2
1.4 Definition of Terms	2
1.5 Objective	3
1.6 Scope of Project	3
1.6.1 User	3
1.6.2 Specific Platform	3
1.6.3 Where the System to be use	3
1.6.4 Module of the System	4
1.7 Organization of the Thesis	4
1.8 Significant of Study	4
CHAPTER 2 LITERATURE REVIEW	6

2.1	Overview	6
2.2	Inventory Management System	6
	2.2.1 Old Inventory Management System	6
	2.2.2 New Inventory Management System	8
2.3	Database	11
CHAPTER 3	METHODOLOGY	14
3.1	Introduction	14
3.2	Utilization of Database	18
3.3	Integrated Development Environment (IDE)	19
3.4	Network System	20
CHAPTER 4	RESULT AND DISCUSSION	26
4.1	Introduction	26
4.2	Requirement Specification	26
	4.2.1 System Conceptualization	26
	4.2.2 System Analysis	27
4.3	Software Design	28
	4.3.1 Software Construction	28
	4.3.2 Interface construction	30
4.4	Implementation and Integration	38
4.5	Testing	39
4.6	Deployment	39
4.7	Maintenance	39
CHAPTER 5	CONCLUSION AND RECOMMENDATION	40
5.1	Introduction	40
5.2	Conclusion	40
5.3	Recommendation	42
REFERENCES		43

APPENDICES		46
A	Gantt Chart Of The Project (Semester 01)	46
B	Gantt Chart Of The Project (Semester 02)	47
C	Coding For Home Form Design	48
D	Coding For Sign In Forms Design	49
D1	Coding For Sign In Forms Design	52
E	Coding Admin Tool Forms Design	53
E1	Coding User Tool Forms Design	55
E2	Coding Sign In Button Forms Design	57
E3	Coding Exit Button Forms Design	58
F	Coding For User Detail Form For Administrator Application Design	59
F1	Coding For Search User Detail Form For Administrator Application Design	60
F2	Coding For Add New User Detail Form For Administrator Application Design	62
F3	Coding For User Detail Form For Administrator Application Design	64
G	Coding For Inventory Stock Form For Administrator Application Design	66
G1	Coding For View Inventory Stock Form For Administrator Application Design	68
G2	Coding For Add New Inventory Stock Form For Administrator Application Design	71
G3	Coding For Update Inventory Stock Form For Administrator Application Design	72
G4	Coding For Update Stock In Form For Administrator Application Design	73
G5	Coding For Update Stock Out Form For Administrator Application Design	75
G6	Coding For Delete Inventory Stock Form For Administrator Application Design	77
H	Coding For User Requisition Form For	79

	Administrator Application Design	
H1	Coding For Check User Requisition Form For Administrator Application Design	80
H2	Coding For Add New User Requisition Form For Administrator Application Design	82
H3	Coding For Delete User Requisition Form For Administrator Application Design	84
I	Coding For Search Item Form For User Application Design	87
J	Coding For Requisition Form For User Application Design	88
J1	Coding For Request Application For User Application Design	89
J2	Coding For Request New Item Form For User Application Design	91
K	Coding For Tracking Item Form For User Application Design	92

LIST OF TABLES

Table No.		Page
2.1	Modules of system	10
2.2	Comparison of inventory management system	11
2.3	Comparison of database management	12

LIST OF ABBREVIATIONS

FKP	Faculty of Manufacturing Engineering
E-ECLAB	Component Electronic Inventory Management System
VB	Visual Basic
UMP	Universiti Malaysia Pahang
PC	Personal Computer
XP	Windows eXPerience
RMS	Risk Management System
AS/400	eServer iSeries/400
SQL	Structured Query Language
IBM	International Business Machines
DB2	Family of RDBMS
ORACLE	Oak Ridge Automatic Computer Logical Engine
SYBASE	Computer Software Company
DBMS	Database Management System
MYSQL	Database
RDBMS	Relational Database Management System
DLL	Dynamic-Link Library
ADO	ActiveX Data Objects
GUID	Globally unique identifier
IDE	Integrated Development Environment
RTL	Right to Left
LAN	Local area network
RDP	Remote Desktop Protocol
.NET	Top-Level Domain
.ETC	and other things

LIST OF FIGURE

Figure No.		Page
3.1	Waterfall process models	15
3.2	Process for software development	16
3.3	Block diagram for system E-ECLAB	17
3.5	E-ECLAB system network block diagram	21
4.1	Relationship between tables in E-ECLAB	27
4.2	Software constructions for user application	29
4.3	Software construction for administrator application	30
4.4	Home interface design	31
4.5	Design of sign in and sign up	31
4.6	Design for tool forms	32
4.7	Design user detail forms for administrator application	33
4.8	Design inventory stock forms for administrator application	35
4.9	Design user requisition forms for administrator application	36
4.10	Design search item form for user application	36
4.11	Design Requisition form for user application	37
4.12	Tracking item form	38
4.13	Syntax for Bar code function	38
4.14	Syntax for SetParent function	39

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The Component Electronic Inventory Management System (E-ECLAB) will be developing to handle Electronic Laboratory in Faculty of Manufacturing, UMP. This chapter briefly discusses about background of study, the definition of terms, problem statement, objective of research, scope of the project, organization of thesis, and significant of project.

1.2 BACKGROUND OF STUDY

Previously, Electronic Laboratory are using the old inventory system, in which formerly students fill out the application form to apply the components manually. Faculty of Manufacturing Engineering which consist almost 500 students has given a burden to the lab assistant to manage all application by manual from students. In addition, laboratory coordinator has difficulty in time management for record all components. Before this coordinator only manually writing to insert the data of quantity of component and make a filing the paper inventory on laboratory of electronic components.

This project is named Component Electronic Inventory Management System (E-ECLAB). It will replace the current system with a more organized and efficient. The current system is developed using Microsoft Excel as database. Where a function of this system is trigger users about the expired date of working standard set. The system to be will be developed using Microsoft Office Access and VB language which act as

database and design interface language. It provides more functions compare to current system. Functions which add in are such as to keep record of all hardware and software which used in production line. With this system, users are able to maintain the hardware, software which used in a more systematic way.

The proposed system will be using distributed system approach. It means that each computer will be located one inventory management system, whereas the database will be located at the server PC. A computer will access data from the server PC. Data which modify by users will be sent to server PC as the database is located at there.

1.3 PROBLEM STATEMENT

In the electronics lab of FKP, laboratory coordinator had face a problem in manage the smaller components in large quantities when students who wish to apply for this component and will impact on throughout monthly and annually stock.

1.4 DEFINITION OF TERMS

Inventory management is primarily about specifying the size and placement of stocked goods. Inventory management is required at different locations within a facility or within multiple locations of a supply network to protect the regular and planned course of production against the random disturbance of running out of materials or goods. Application of inventory management widely use at warehouse and supermarket like IKEA, Courts Mammoth, TESCO, CARREFOUR.

There are a lot of definition terms to create the inventory management. An inventory management is the process of overseeing the constant flow of units into and out of an existing inventory. This process usually involves controlling the transfer of the unit to prevent inventory from being too high, or is significantly reduced to a level that can be put into operation a system is in danger. Competent inventory management also seeks to control costs and quantities associated with inventories, both from the perspective of the total value added goods and tax burden generated by the cumulative value of inventory [1]. But another term of inventory management is when all

operations keep inventory, which refers to the stored materials that exist within an organization and applies to raw materials, purchased parts, work in progress and finished goods. The terms inventory and stock are used interchangeably, as is the case here, although by definition inventory is usually referred to in value for example in pounds and dollars while for stock in quantity, example in kg, 100s and meter. Inventory primarily arises because of differences in the timing or rate of supply and demand and is used to balance these. Inventory may also occur due to economic batch sizes for an operation, product seasonality and investment for new product ranges. [2]

1.5 OBJECTIVE

The objective of the research project summarizes what is to be achieved by the study. These objectives should relate to the research problem. Which is:-

- i. To create database for inventory Management system of Electronic laboratory.
- ii. To develop an interface for E-ECLAB that can connect with database.

1.6 SCOPE OF THE PROJECT

1.6.1 User

The users of this system consist of coordinator or administrator, and client or know as students, staffs of Faculty of Manufacturing Engineering. Each class of user has their access level.

1.6.2 Specific platform

This system will be run by using Microsoft XP/ WINDOW7 operating system. Besides that, other platform Visual Basic input output references are used in this project where the function is to link interface and machine device.

1.6.3 Where the system to be use

Target use of this system is around Manufacturing Faculty's laboratory in University Malaysia Pahang so that users can access the system in certain link within UMP server.

1.6.4 Module of the system

There are seven module will be implementing in the E-ECLAB application which in management module is stock inventory management module, user management module, requisition management module, requisition new item management module, user registrations module, info component management module, tracking requisition management module.

1.7 ORGANIZATION OF THE THESIS

This thesis consists of five chapters. Chapter 1 is describing an introduction about E-ECLAB. Chapter 2 is the literature review on article and journal it more highlights on argument that find by make a comparison of journal. Chapter 3 where it's explained the methodology will use in this the project. Chapter 4 is discuss on the analysis of the design made, from the testing, result and discussion regarding on the project. Chapter 5 is to conclude all the project has been made and some recommendation for future works.

1.8 SIGNIFICANT OF STUDY

This application of E-ECLAB inventory management given many advantage especially to lab assistant of electronic laboratory because can manage the system much easier, arrangement of data, and labor efficient. This E-ECLAB consist of Visual Basic Express Developer software and by using language VB to design a module interface

The system is dividing into two sites which is master site and slave site. The slave site can be an interface application that can be access by user only. The master site is the

developer coordinator can access this site. Where, it function to add new design interface or to delete the selected design. At master site the coordinator are allow adding more databases and deleted it. The slave site as default provides the read only file to the public. Only authorize user are allowed to access and modified current data in the system. The user of this site is coordinator, staff, and students. The coordinator is responsibility to maintain the system. The coordinator will approve the proper user account registration. The system will allow coordinator to create a save location's information into the database. The system also provide report module so that, administrator can generate the report regarding the inventory.

Besides, the time taken in application transaction also can be reducing. The transaction such as requisition item and new item can be done in online. At the end of online application each client will provide with the different barcode to easier the client to take a component in machine. The record of each transaction will be saving in the database.

The system will allow administrator to store and retrieve data in a highest security level database. The risk of losing data and the problem of having redundant data can be avoided. Other than that, clients are provided with the info about each item datasheet which provides details information of component. Only registered visitor are allowed to make collection's reservation and borrow.

CHAPTER 2

LITERATURE REVIEW

2.1 OVERVIEW

In this chapter discussed the research that has been carried out. Most of the materials used are based on the book and journal publications

2.2 INVENTORY MANAGEMENT SYSTEM

2.2.1 Old inventory management system

An inventory control system is a set of hardware that integrates with software based tools that automate the process of tracking inventory. Inventory management systems cannot be considered as isolated systems but are linked to many different management areas within companies [3]. The old system of inventory management was bin-card system, which had been adopted by the raw material store some ten or more years earlier. The characteristics of the old bin card system are each bin card signified one stock item with the stock code ranging. The bin card contained six columns: date receipt, issue, balance, M/R or invoice number, and location of goods. The store keeper was solely responsible for the accurate recording of information on the bin card [4].

The preparation of delivery orders for sub- contracting work is done using the bill of materials (BOM) and standard reject rates. The store is authorized to issue the necessary materials to sub-contractors according to sub-contracting purchase orders or delivery orders approved by the purchasing manager [4].

The old inventory system was too "conservative" and relied on a considerable amount of manual effort, and its deficiencies were constantly being exposed as business increased. It was apparent that, as activities increased, the old system would face the following problems with increasing severity such as:-

a. Clerical equipment

Clerical requirements will increase, especially as the number of products and associated materials increase. The purchasing group needed to spend considerable effort in calculating the materials requirement for sub-contracted work.

b. Low efficiency

Low efficiency knows as a result of errors in recording and computations. Mistakes due to human error could easily be made and propagated, and this could lead to the occurrences of stock out situations or over stocking.

c. Bin card storage

Bin-card storage problems occurred whenever new models or new product lines were launched. Each new stock item would require the assignment of an additional stock code number.

d. Insufficient and slow information

Insufficient and slow information flow, where could only come from the physical card system. Each stock card had to be checked in order to obtain inventory status information. Daily transactions reports from the store were not immediately available.

e. Material location

Material location problems occurred. The location of each stock item should have been clearly stated in the column of the particular stock card. In practice, the location column was often left blank, with locations being memorized by the store keeper.

f. Difficulty in controlling

Difficulty in controlling transactions related to subcontracted work. Transactions related to sub-contracted work became more frequent and complex, thus requiring greater effort. This would lead to over-issue or under-issue of materials, the referencing of the wrong bill of materials, and numerous other accounting errors. After completion, sub-contracted work was often delivered in more than one lot over different accounting periods, and the accounts department had difficulties in reconciling the costs of materials with each lot. Often all the costs of materials were allocated to the lot delivered first, giving the first lot a high unit cost; while the second and later lots were accounted as having no elements of material cost.

2.2.2 New Inventory Management

The new inventory system is designed around the needs of the RMS, and generates updated Stock status reports with the use of PCs [5]. It consists of the six modules as shown in Table 2.1, with the following specific functions.

a. Transaction processing.

This handles the daily transactions and physical count adjustments and can be sub-divided into the following units:-

- i. The daily transaction processing unit. This accepts all types of transactions, such as purchase receipts, material requisitions, and returns from the shop. The on-line updating of data to other files immediately after entry of a transaction. If

data entry errors are found on updating, the reserving/adjustment transaction can be entered into too in order to correct the stock.

- ii. The physical count adjustment unit accepts the actual quantities counted as the true record.
- iii. The display adjusted transactions unit lists a summary of transactions adjusted on the screen sorted by item number. This allows quick scan on what has been entered to ensure accuracy.
- iv. The print transaction edit list section allows checking of data entered before posting. A batch total is also provided at the end of report. The post to update files unit updates the actual quantities to the item file. It also creates a historical record in the transaction history file. This can be printed out through the "print daily transaction list" function of the reports menu. The system will also automatically backup the current data before proceeding. In case of interruption, the transaction can be recovered using the "restore" function of the system utility option.

b. Module maintenance.

The file maintenance module deals with the maintenance of the item master file, password file, and the maintenance of the control file. There is no need to enter the quantity and location of an item since this is shown on the screen and is equal to the distribution quantity of the item in the different locations. Up to sixteen locations are shown on any one screen, and additional locations are displayed on appropriate prompting.

c. Module of printing.

The reporting module prints daily and monthly reports. The daily transaction list program prints all transactions within a day of operation, sorted by item number. It also includes reports on the current stock status of all items requested.

Table 2.1: Modules of system.

Module	Function
Transactions processing	Entry of daily transactions and physical count adjustments transactions
File maintenance	Allows user to set-up basic item information and password
Reporting	Prints daily and monthly reports
Inquiry	Displays item stock status and history of transactions
End period and closing	Clean all transactions and updates internal item usage tables
System utility	Saves data from hard disk on to floppy disk and organizes data files

Adapted from “A computer system for inventory management of lighting products: A case study” by S.O.E. Ng, E.C.Partington and D. Sculli, 1992, copyright 1993 by Elsevier. Adapted with Permission

Based on Table 2.2, clearly show that the advantages using new inventory is a good way to improved the system in laboratory.

Table 2.2: Comparison of inventory management system

Old inventory management	Comparison	New inventory management
All process is written in piece of paper.	Method	Overall processes are online only certain processes are manually.
Low. Occur during record the data	Efficiency	High. Every transaction has been record in database
Take time up to one to five work day	Duration	Take time minimum transaction less than one to five hour only
High cost efficient. This because still using stationery tools for all transaction made.	Cost	Low cost efficient. This because all transaction has been made online in system.

2.3 DATABASE

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images. In computing, databases are sometimes classified according to their organizational approach. The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. A distributed database is one that can be dispersed or replicated among different points in a network. An object-oriented programming database is one that is congruent with the data defined in object classes and subclasses.

Computer databases typically contain aggregations of data records or files, such as sales transactions, product catalogs and inventories, and customer profiles. Typically, a database manager provides users the capabilities of controlling read/write access, specifying report generation, and analyzing usage. Databases and database managers are prevalent in large mainframe systems, but are also present in smaller distributed workstation and mid-range systems such as the AS/400 and on personal computers. SQL (Structured Query Language) is a standard language for making interactive queries

from and updating a database such as IBM's DB2, Microsoft's SQL Server, Microsoft Access and database products from Oracle, Sybase, and Computer Associates. [6]

By comparing two database management system or DBMS which is MySQL and Microsoft Access which one is suitable to use in this application system of E-ECLAB. First of all, there are no Forms or Reports in SQL Server. Access is a program that contains a design interface for a relational database management system or RDBMS, and also an application design interface. SQL Server is just an RDBMS. There are several tools that come with SQL Server (Standard Edition and up) which help to manage the databases, but SQL Server itself is just an RDBMS. To create a SQL Server database as a back end panel, what need is to use some other development environment. Access can be used for that. Create a new Access project and tell it what SQL Server database that wants to connect to, and allow to Forms and Reports that interact with the database.

Access also calls as a Passive DBMS. Access does not have to be open and running for someone to access the data in an Access database. When connects to an Access database, it communicates through a few DLL files that interpret the SQL statements, or commands from ADO, DAO, etc. There is no active program that is managing the database. SQL Server on the other hand, is an active DBMS. When it is installed, it becomes a Windows Service that runs in the background. It is an active program that is constantly monitoring all activity that goes on with the databases it manages. If the SQL Server service is stopped, then all database activity will be stopped. Databases that has created can't access a SQL Server database if the SQL Server instance it is connected to is not running.

Because of this point, SQL Server is a much more stable, robust and secures DBMS. It allows to do much more to manage databases; to protect them and keep them running. Also it gives the ability to monitor all activity in real time. Access is simply a tool to build a database, while SQL Server provides ways to really manage database performance. Now, because it is more comprehensive, it is also much more complex to manage. SQL Server will take longer to learn, and requires a more concentrated study to

be able to make the most of it different with access database where access is friendly to uses. [7]

Finally based on the finding on differential between My SQL and Access, the suitable for this system is Access as show in Table 2.3, this because it more suited for desktop use with a small number of users accessing it simultaneously. One reason why choose to use Access over SQL Server is for compatibility/sharing. The system might need to email someone a copy of data that have database. People are more likely to have Access on their desktop computer than SQL Server. Generally use of SQL suitable for developers' computers/servers or on production server machines. For administrator that handle this system is not the developer so access is more suitable.

Another reason for use Access instead of SQL Server is money. Many of PC's are installed Access as part of the Microsoft Office suite. Purchasing SQL Server would be an extra expense that may not be necessary - depending on this situation. SQL Server can also be quite expensive. [8]

Table 2.3: Comparison of database management

MySQL	Comparison	Microsoft Office Access
No forms or reports	Interface	Contain with interfaces for a relationship database management (RDBMS)
If turn Off the SQL Server, all the database operation will be halted	Operation	When its installed, it become windows service that run in background of windows. It is active program that constantly monitor all activities that goes on with the database it manage

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter will describe the tools and equipment used to analyst software development in inventory of E-ECLAB. Methodologies are divided into two segments, the gathering process and the data analysis procedure. Data gathering is one primary task that involved in researching, by find the hypothesis and tests it to get the true validity. In order of that this researching must have to gather the data analysis. There are many procedures for gather the data such as make an interview with the laboratory coordinator to gather the information about the old inventory system and testing the system that had design.

On other hand, data analysis is done once there have gather enough data for tests the validity hypothesis. Usually, to ensure the applications has finally validity the hypothesis there are several steps that has to consider in this analysis by using the waterfall model that will be discuss in chapter 4. [9]

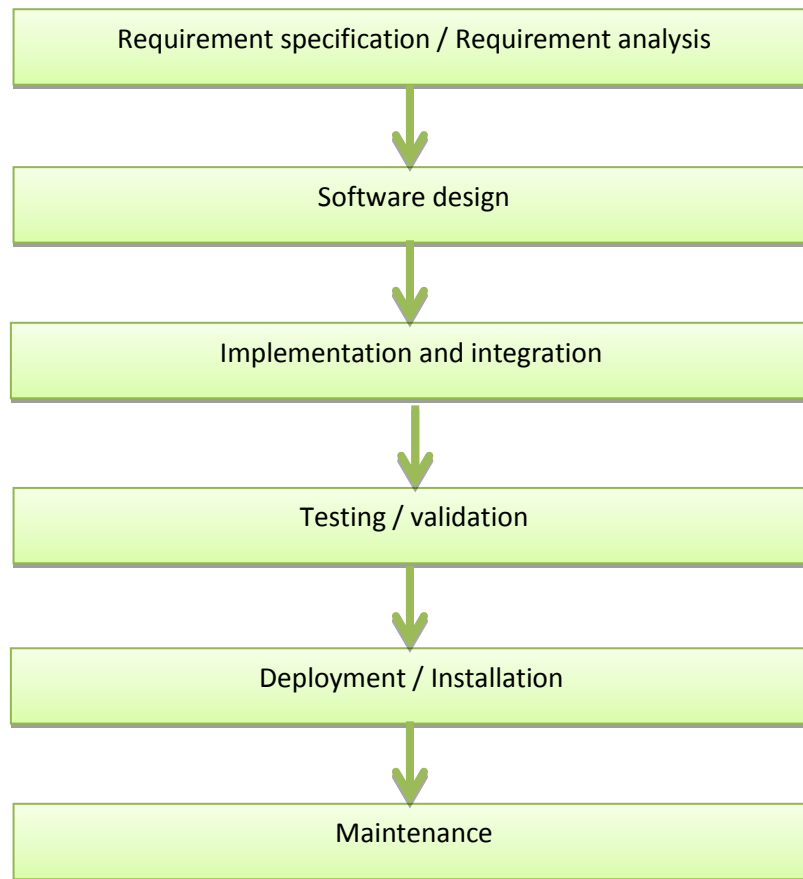


Figure 3.1: Waterfall process models [9]

The works begin with finding the information about the project. This is where the research problems are identified. In this project, need to identify the databases that need to use in this inventory and design the form for an application. It has been the problems that need to be solved. The works continues with the literature reviews on comparing database between SQL and Microsoft Office Access and comparing between developer Visual Basic Express and Microsoft Office Access. This is important for the researcher to understand the fundamental concept and operations carried out. Figure 3.2 show the flow chart for the process need in software development.

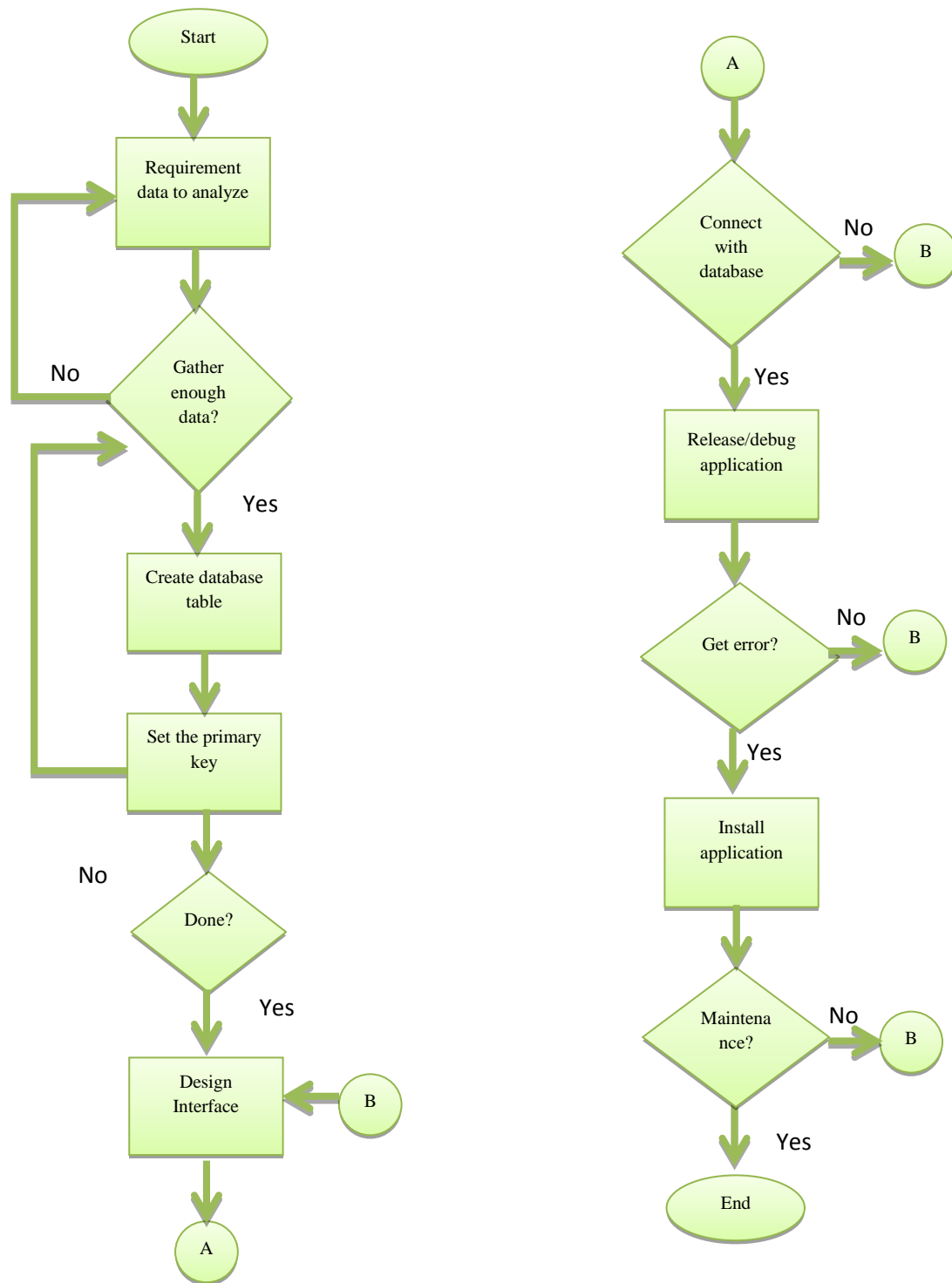


Figure 3.2: Process for software development

First step in designing an E-ECLAB database is to analyze data requirements by determining the purpose of the database and how it is to be used. This way tells what information that need for the database. From that, it can determine what subjects that uses to stored data in database and what steps that need to stores data for each subject by the fields the data in tables.

By use the data given by coordinator, the data must have to enough to gather the information about modules that will use in database. Brainstorm about the questions that database like to answer. Sketch out the block diagram of the system look like it to produce in Figure 3.3. Gather the forms by currently use to record the data that had sketch. In that way, it will use all this information in the remaining steps of the design process.

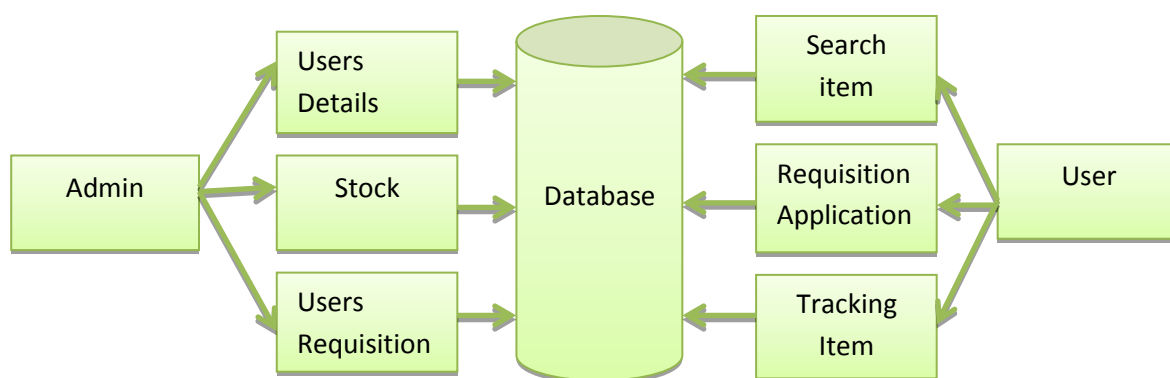


Figure 3.3: Block Diagram for system E-ECLAB

In this system there are two parts that have to consider in this application they are administrator segment and user segment. For the administrator segment there are had three modules, which for user detail, stock application, user requisition. User details are the module that the administrator are allow to access to view the detail of users. For stock application is the main module for the administrator where admin are allow to access the stock with add new item, view monthly stock, view yearly stock and update stock. At this module the stock can show where the components quantities flow. After that, admin also are allow to access to view the new requisition have made by user there are requisition item that have in laboratory and other requisition is the new item that don't have in laboratory.

Segment for user, there are about 3 modules that uses in this activities. Which are search components, requisition form and tracking item; For search item, user are only allow to access by searching selected item that have in laboratory and check item ethers the selected component available in laboratory or not. In this module also have with cart section where it function as component laboratory catalog and ordering process, where to allowing the user to select component, review what the user have selected, make modifications or additions and request item in requisition. Next user can apply the requisition by fill up the form that creates in module requisition application; there are two forms that can fill up by user. Which is the requisition item and new item section, for request item section, user must know the code for each component that want to request or other way to find the component by search the item first or use the cart service. While for requisition new item, there several things that user must know by knowing the item name and the details of the component. In this modules, after user successful submit their application they will receive the barcode that automatically generate for the each application that have made by user. After user had made the requisition, user can track the requisition by enter the student number in the tracking section; in this section, it show the status of requisition application that have made by user, if the requisition form is approved the user can use the barcode given to scan for collect the component that have request.

3.2 UTILIZATION OF DATABASE

Early in chapter 2 have discuss about the database system, shows the Microsoft Office Access is much easier to create desktop databases. It's an easy-to-use tool for quickly creating browser-based database applications that help to run E-ECLAB application. Data that have inserted by user is automatically stored in a SQL database in Access, so it's more secure and scalable than ever, and it can easily share the data applications with other user. These databases are come out with one database that uses five tables to present data of:-

- i. The user account
- ii. Components detail
- iii. Inventory stock of component
- iv. User requisition
- v. cart service

3.3 INTEGRATED DEVELOPMENT ENVIROMENT (IDE)

An integrated development environment (IDE) also known as integrated design environment or integrated debugging environment is a software application that provides comprehensive facilities to computer programmers for software development.

An IDE normally consists of a:-

- i. Source code editor,
- ii. Compiler and/or interpreter,
- iii. Build automation tools, and
- iv. Debugger (usually).

IDEs are designed to maximize programmer productivity by providing tight-knit components with similar user interfaces. Typically an IDE is dedicated to a specific programming language, so as to provide a feature set which most closely matches the programming paradigms of the language. A programming paradigm is a fundamental style of computer programming, in contrast to a software engineering methodology, which is a style of solving specific software engineering problems. Paradigms differ in the concepts and abstractions used to represent the elements of a program (such as objects, functions, variables, constraint) and the steps that comprise a computation (assignments, evaluation, continuations, and data flow).[10]

By using the Visual Basic as the developer for design an interface for this system, this system has design using FlowLayoutPanel. The FlowLayoutPanel control arranges its contents in a horizontal or vertical flow direction. Its contents can be wrapped from one row to the next or from one column to the next. Alternatively, its contents can be clipped instead of wrapped. By design it with specify the flow direction by setting the

value of the FlowDirection property. The FlowLayoutPanel control correctly reverses its flow direction in right-to-left (RTL) layouts. Using this tool are easily by specify whether the contents of the FlowLayoutPanel control are wrapped or clipped by setting the value of the WrapContents property. Any Windows Forms control, including other instances of FlowLayoutPanel, can be a child of the FlowLayoutPanel control. With this capability, system can construct sophisticated layouts that adapt to system form's dimensions at run time. Docking and anchoring behaviors of child controls differ from the behaviors in other container controls. Both docking and anchoring are relative to the largest control in the flow direction. [11]

3.4 NETWORK SYSTEM

Network Operating System refers to software that implements an operating system of some kind that is oriented to computer networking. These network systems are design to allow the user access by shared the files and printer among multiple computer in network, commonly the system use a local area network or LAN, a private network or to other networks. In a peer-to-peer network operating system users are allowed to share resources and files located on their computers and access shared resources from others. This system is not based with having a file server or centralized management source. A peer-to-peer network sets all connected computers equal; they all share the same abilities to use resources available on the network. [12]

Software that been used in this application is allow the user made link with the database server is remote desktop protocol. Remote Desktop Protocol (RDP) is a proprietary protocol developed by Microsoft, which provides a user with a graphical interface to connect to another computer over a network connection. The user employs RDP client software for this purpose, while the other computer must run RDP server software. [13] By using software that integrate in window which is Remote Desktop Connection. With these applications it made the network become easier to handle by coordinator. This system is allowed user and coordinator to access database it show in Figure 3.5.

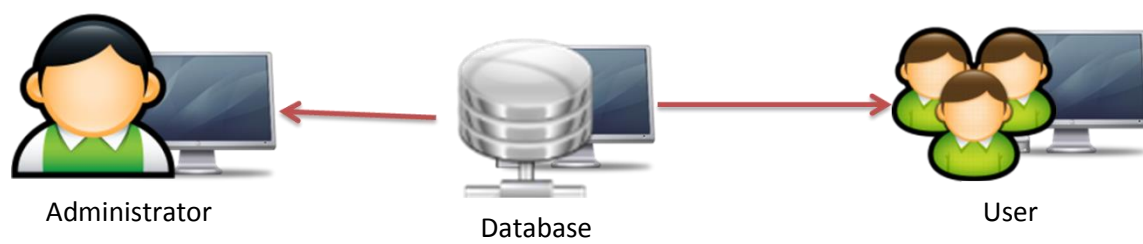


Figure 3.5 E-ECLAB system network block diagram

CHAPTER 4

RESULT AND DISCUSSION

4.1 INTRODUCTION

This chapter will briefly discuss on the analysis data collection from waterfall process models that has shown in chapter 3, result and discussion regarding on the project. The forms are design throughout the project will be discussed.

4.2 REQUIREMENT SPECIFICATION

4.2.1 System Conceptualization

This concept need to be consider of all aspect of the targeted of application function or process, with the objective to create a database system of electronic laboratory and design the application graphical user interface GUI for user application. Include, which aspect will be incorporated into the system. These conceptualizations have been discus in chapter 3, where the concept of this system is show in Figure 3.2. In this system, user only needs to register as user to allow them to make the requisition by submit their requisition by online in certain location in FKP. For the administrator or the coordinator that handle these systems just need to approve the requisition that have been made by user. Besides that, administrator also will alert with the quantities of the inventory component stock where there have three level of inventory stock which is low, medium and high.[14]

4.2.2 System Analysis

By gather the entire requirement that will use in this system, with the objective of this project these requirements will be accommodated in the system. Extensive communication between the user and the administrator is essential. The tables that create in database are required to this application system. The relationships between data are creating to allow the data connect each other, by using primary key as the main data to connect the table. The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute that is guaranteed to be unique such as Student ID in a table with no more than one record per person or it can be generated by the Database Management System or DBMS such as a globally unique identifier, or GUID, in Microsoft Office Access. Primary keys may consist of a single attribute or multiple attributes in combination. Figure 4.1 show the relationship between the tables in database.

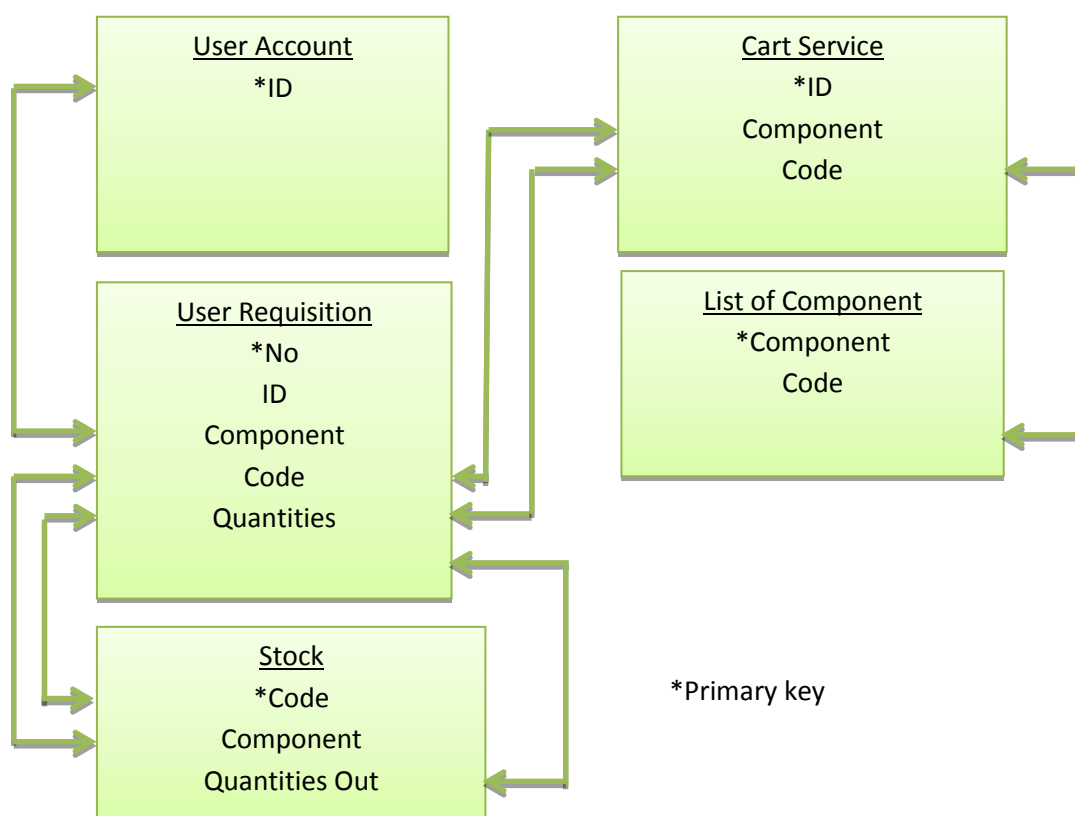


Figure 4.1: Relationship between tables in E-ECLAB

4.3 SOFTWARE DESIGN

Once all the entire requirements have been collected and analyses. By identify the detail of this system; the system will be constructed to perform necessary tasks. More specifically, the software design phase is focused on the data requirement that have been archive in requirement specification, the software construction, and finally the interfaces construction. For requirement specification have been discuses in previously subchapter 4.2. Next will discuss on the software and interface construction

4.3.1 Software construction

Software construction is the collection of programs, processes and information used to perform various tasks within the building or assembling of a structure or infrastructure as a means of increasing productivity, efficiency, and competitiveness. Tasks previously administered by an administrator or coordinator can be performed by construction software applications. For firms considering construction software, it is important to consider how the software integrates and interoperates with the database and existing software applications used by the firm and its subcontractors. Other influential factors are CPU speed, hardware configuration and the amount of memory needed. [15]

This application has divide by two part of application, which is for user and administrator. For the construction, the user and administrator will have three sections for each application. For user application, the sections are search item, application form and tracking item. In section search item, there have filter section which allow user to filter the by insert the keyword of component. In this section also have cart service which allow user to use this service to store catalog and ordering process in application form.

Next the application form, there are divide by two section which is requisition for existed item and requisition for new item. For requisition item, user just need to fill up the form given in application and print it out and scan the barcode given at component vendor machine to pick up the selected item that have made in system

previously. After user has made the requisition for new item, user can track their item in tracking section to check whether the item request is available in system or not. For requisition item that available in laboratory user can obtain the component immediately after submission the requisition while for new item requisition, user need to check up the requisition approval about 3 day work. This because to give time to the administrator to make a reservation to the new component. Figure 4.2 are shown the software construction for user application

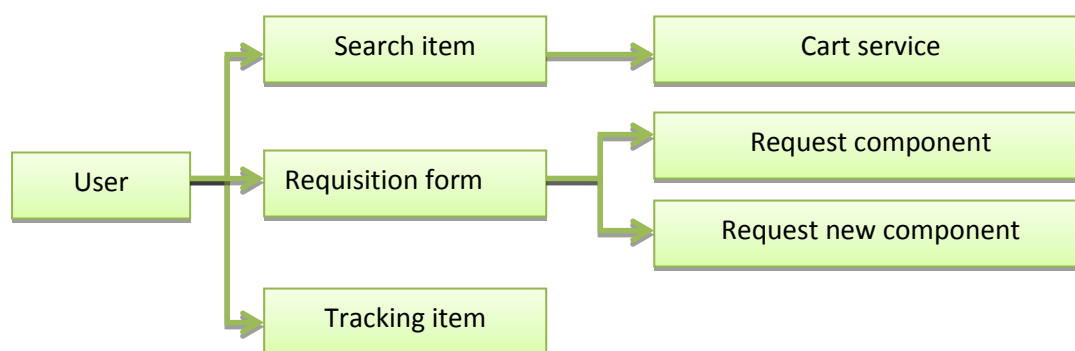


Figure 4.2: Software constructions for user application

For administrator application there are users detail section, where in this section only allow administrator to find, add new and delete the user in system. In find user details, admin just need insert student ID or first name in text box given; admin need to insert by select the checkboxes for filter by student ID or first name of user. Then the result all the user detail that relate on filter keyword will show on table given. Admin also can add new user into system by manually, if the student has problem to sign in or sign up in system. Delete user section is usually use when admin want to delete the certain user that have not available study in FKP.

Next section is inventory stock, these inventory stocks are used to view the stock, add new item, update stock and delete selected item. Admin can view stock by select the checkboxes to check level of component quantities or filter by insert keyword of component detail such like name of component or code component. The result will show in table given in application interface. For add new item function is same like add new user in user details section. When there have new stock purchase, admin can update

the stock by inset data in stock in updated section. Function for delete section in these applications is to delete the selected item that admin want to remove from database.

Last section in administrator application is user requisition, where here admin are able to view users that has made the requisition and also for new item requisition. Admin also are allowed to use manually requisition that request from staff or student that have problem to sign in the system, it can fill by admin. Figure 4.3 are shown the software construction for administrator application

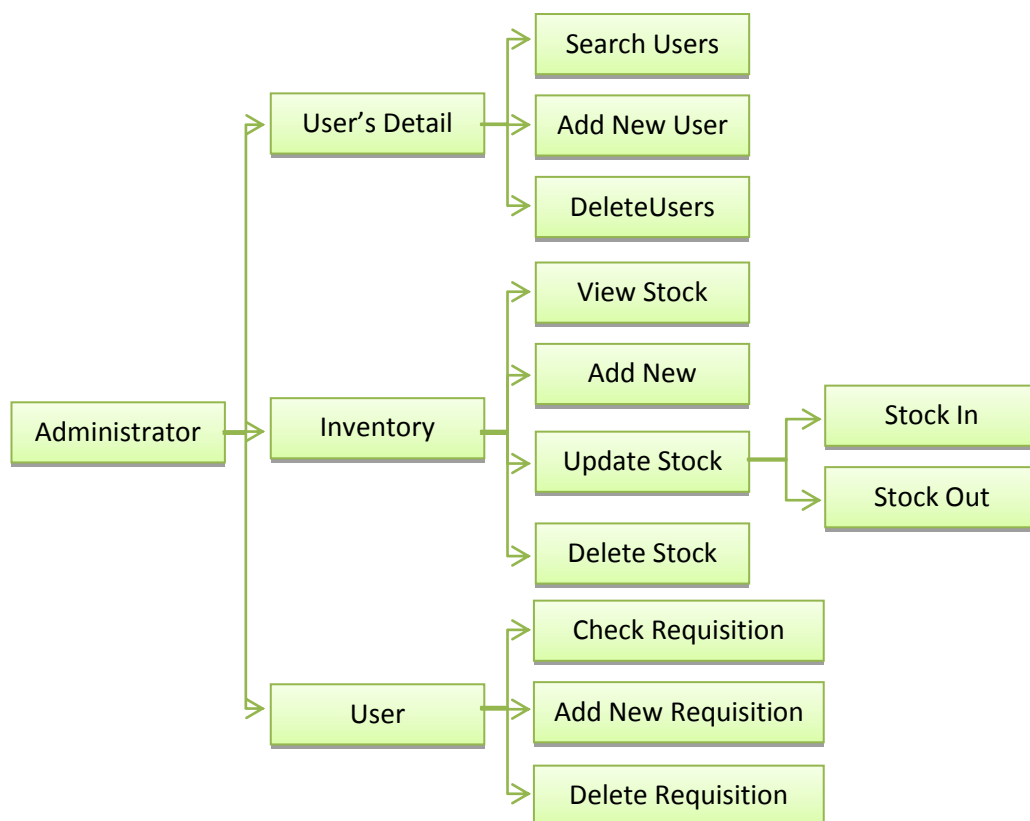


Figure 4.3: Software constructions for administrator application

4.3.2 Interfaces construction

Based on the software construction that have discuss, there are about 30 forms design has been create in this application. There are:-

a. Design of Home form

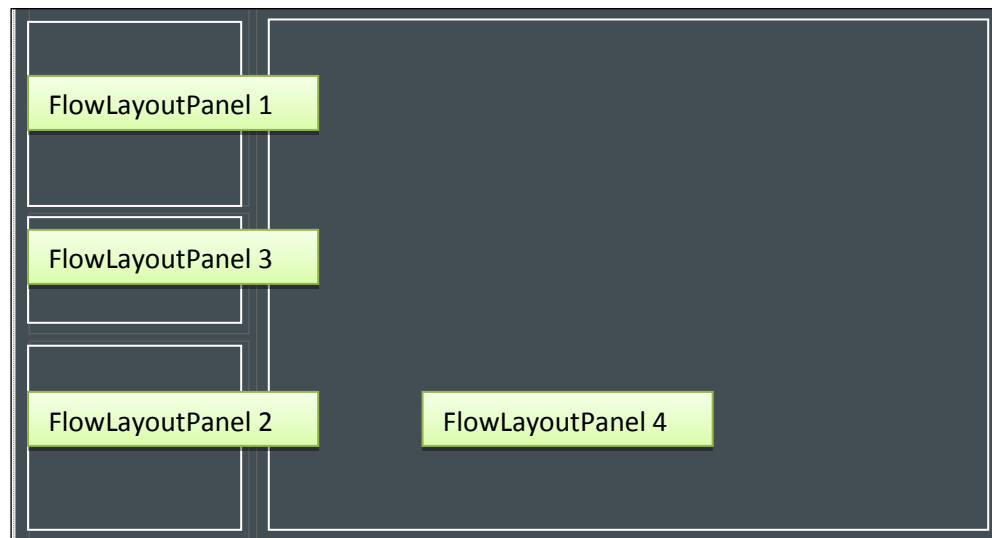


Figure 4.4: Home interface design

Based on Figure 4.4, the form are design using FlowLayoutPanel tool only, where the function for FlowLayoutPanel is allow the another form to popup in home form.

b. Design of Sign In and Sign Up form

**Welcome to FKP Inventory System of
Electronic Component e-CLAB**

Username

Password

[Sign Up Here!](#)

Create new account :

Some basics to get you started

ID:

First Name: Last Name:

Identification Card No. : - -

Contact No:

Email:

(a)
(b)

Figure 4.5: Design of Sign In and Sign Up form. (a) Sign in interface (b) Sign up interface

The forms are design using textboxes, buttons and link button.

c. Design of tools forms

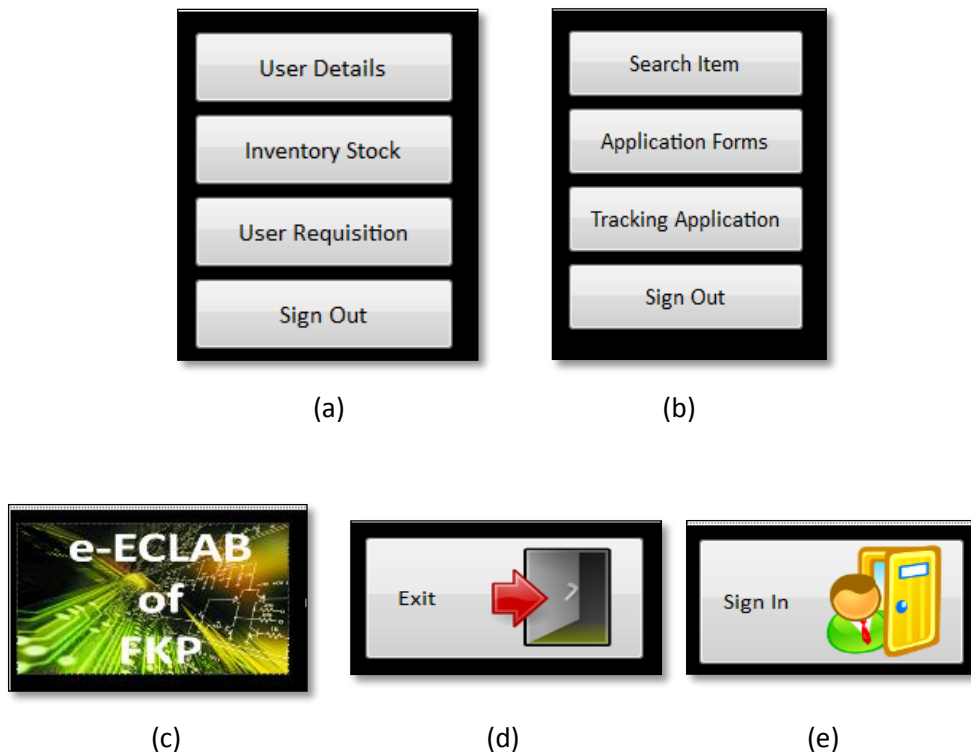


Figure 4.6: Design of tools forms. (a) Admin tool form (b) User tool form (c) Logo of E-ECLAB form (d) Sign Out form (e) Sign In form

Overall of this form are design using button, where each button have functionality. For admin tool form has four buttons that can open User details form, inventory stock form, requisition stock form and lastly sign out button.

d. Design user detail form for administrator application



(a) User details form with buttons for Search, Add New, and Delete.

(b) Search user detail form with a 'How to search user?' section containing four steps: Step 1: Select filter by code or component, Step 2: Insert keyword, Step 3: Check details first before delete, and Step 4: Delete stock. It includes a 'Filter By' section with checkboxes for ID and Name, a search input field, and a Refresh button. Below is a data grid with columns: ID, First_Name, Last_Name, IC_No_1, IC_No_2, IC_No_3.

(c) Add new user form with a 'How to add new user?' section containing two steps: Step 1: fill all form. and Step 2: To save all data click Save button. It includes a 'Please fill the form :-' section with input fields for ID, First Name, Last Name, IC No 1, Contact, and Email, and a Save button. Below is a data grid with columns: ID, First_Name, Last_Name, IC_No_1, IC_No_2, IC_No_3.

(d) Delete user form with a 'How to delete user?' section containing four steps: Step 1: Select filter by code or component, Step 2: Insert keyword, Step 3: Check details first before delete, and Step 4: Delete stock. It includes a 'Delete by' section with checkboxes for ID and Name, a search input field, and a Delete button. Below is a data grid with columns: ID, First_Name, Last_Name, IC_No_1, IC_No_2, IC_No_3.

Figure 4.7: Design user detail form for administrator application. (a) User Detail form (b) Search user detail form (c) Add new user form (d) Delete user form

In user detail form are design using button for search users, add new user and delete users. The form is design using FlowLayoutPanel to allow opening another form. Another form is design using textboxes, buttons, datagridview and checkboxes.

e. Design inventory stock forms for administrator application

Inventory stock

View Update Add New Delete

(a)

View Stock

Filter by: Component Code

Filter by Quantity: Low Medium High

Code	Company	Last_quantity	Last_date	Quantity_in	Date_in
*					

(b)

Add new stock

Please fill the forms for add new stock.

Code:

Component:

Type:

Company:

Last quantity:

Last date:

Current quantity:

Current date:

(c)

Update Stock

(d)

Stock in

Please fill the forms for update stock in.

Code

Component

Code	Component	Type	Company	Last_quantity

Code:

Component:

Type:

Quantity in:

Date in:

(e)

Stock in

Please fill the forms for update stock in. How to update stock ?

Code:

Last quantity:

Last date:

Current quantity:

Current date:

Quantity out:

Date out:

Step 1: Insert code of component.

Step 2: Insert quantity

Step 3: click update button to save data.

(f)

(e)

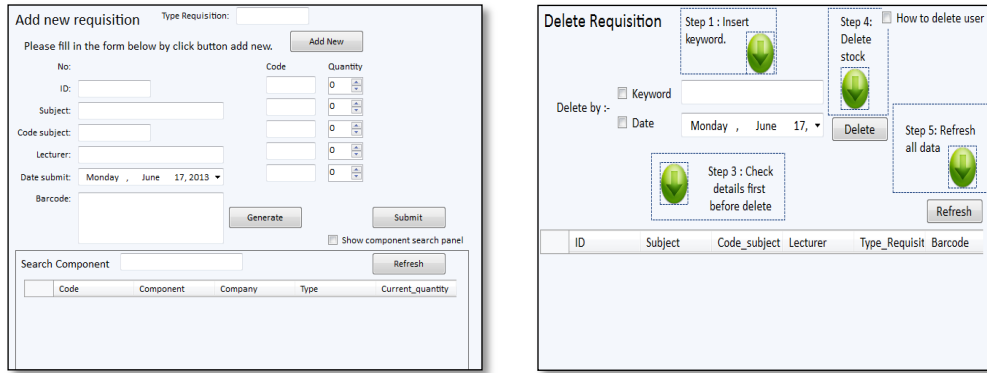
Figure 4.8: Design inventory stock forms for administrator application. (a) Inventory stock form (b) View stock form (c) Add new Stock form (d) Update stock form (e) Stock out form (f) Stock in form (g) Delete Stock form

For inventory stock and update stock are design using button for search users, add new user and delete users. The form is design using FlowLayoutPanel to allow opening another form. Another form is design using textboxes, buttons, datagridview and checkboxes.

f. Design user requisition forms for administrator application

(a)

(b)



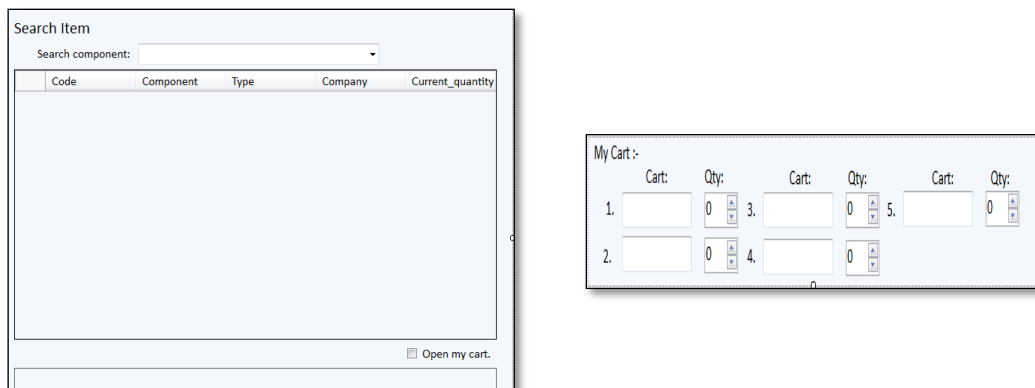
(c)

(d)

Figure 4.9: Design user requisition forms for administrator application (a) User requisition form (b) Check user requisition form (c) Add new Requisition form (d) Delete Requisitions form

For user requisition form is design using button for search users, add new user and delete users. The form is design using FlowLayoutPanel to allow opening another form. Another form is design using textboxes, buttons, datagridview and checkboxes.

g. Design search item form for user application



(a)

(b)

Figure 4.10: Design search item form for user application (a) Search item form (b) Cart service form

Forms are design using textboxes, buttons, datagridview, numericupdown tools.

h. Design Requisition form for user application

(a)

(b)

(c)

Figure 4.11: Design Requisition form for user application (a) user application forms (b) Request item form (c) New item request form

For user application is design using button for search users, add new user and delete users. The form is design using FlowLayoutPanel to allow opening another form. Another form is design using textboxes, buttons, datagridview and checkboxes.

- i. sign tracking item form for user application

Figure 4.12: Tracking item form

Tracking form is design using textboxes, buttons, datagridview and checkboxes.

4.4 IMPLEMENTATION AND INTEGRATION

In this implementation for the system, an important part that must be considered is the coding of the software design. Coding also known as a system of symbols and rules; which function is to represent instructions to a computer. Requirement and system specification from the software design step, are translate into machine readable computer code. Using VB language to write the coding for each design there are several reference and function that use in this application. There are bar code references for convert the text into font encoder of code 39 and SetProperty function to set the parent window of a pop-up, overlapped, or child window. For declaration function syntax for bar code that use in this interface is shown in Figure 4.13 is function to generate bar codes from a font in application. For declaration syntax function for SetProperty that use in this interface is shown in Figure 4.14 is function to changes the parent windows of the specified child window. For coding for each interface that has design is shown in Appendixes.

```
Dim fontencoder AsNew IDAutomation.NetAssembly.FontEncoder
```

Figure 4.13: Syntax for Bar code function

```
DeclareAutoFunction SetParent Lib"user32"(ByVal hWndChild As  
IntPtr, ByVal hWndParent As IntPtr) As Integer
```

Figure 4.14: Syntax for SetParent function

4.5 TESTING

The software are created to being added in developing system by link it with the database that it can stored the data form the software design, testing is performed to ensure that it is working correctly and efficiently. Testing is generally focused on two areas, which one is internal efficiency and external effectiveness. The objective of external effectiveness testing is to verify that the software is functioning according to system design, and that it is performing all necessary function or sub-functions. Next for internal testing is to make sure that the computer code is efficient, standardized and well documented. Testing can be a labor intensive process, due to its iterative nature.

4.6 DEPLOYMENT

Deployment starts after the code is appropriately tested, approved for release or otherwise distributed into a FKP area. This may involve installation, customization such as by setting parameters to the customer's values, testing, and possibly an extended period of evaluation.

4.7 MAINTENANCE

Maintaining and enhancing software to cope with newly discovered faults or requirements can take substantial time and effort, as missed requirements may force redesign of the software. For administrator, software training and support is important, as software is only effective if it is used correctly.

CHAPTER 5

CONCLUSION AND RECOMMENDATION FOR FUTURE WORKS

5.1 INTRODUCTION

This final chapter represents about conclusion and recommendation for the project. In this chapter will discuss mainly about the conclusion of the project, concluding all the process that involved. Besides that this chapter also contains recommendation about the project. So for this recommendation it can make improvement about the project in the next semester

5.1 CONCLUSION

The project of E-ECLAB inventory management for software development had achieved target only but for overall software development are not complete achieved due to certain problem. There are two main objectives stated in the early stage of the project. The first objective is to create database for inventory management system of electronic laboratory. Databases store information in electronic records that may be searched, retrieved and organized in countless ways. By have own system information in a database, instead of on paper or in spread sheets, not only it's saves times and preserves vital information, it allows administrator to see patterns in system operations that are visible in no other way.

There are many advantages using databases instead on old inventory. Databases save time by retrieving a single record, which may take several minutes of digging in a paper-based filing system, can be done with just a few clicks in a computerized database. Merging database records with documents makes producing paper reports

almost instantaneous, eliminating hours of retyping. Data entry screens are design using Visual basic to let administrator to see related records as administrator create a new one, so administrator don't have to spend time wading through on check spellings, addresses, historical data, such as previous requisition. Second is Databases giving lots of different ways to look at system data; a request for a specific answer or list from your database called a "query" can take just a minute or two to write, as opposed to hand collating data records or sorting and sorting a spread sheet. Plus, administrator can write standing queries to search component in database, so an up-to-the-minute answer is just seconds away.

The second objective is to develop an interface for E-ECLAB that can connect with database. Create interface using Visual basic by link the interface and database using databinding. DataBinding is a powerful feature provided by the .NET framework that enables visual elements in a client to connect to a datasource such as DataSets, DataViews, Arrays etc. Some of the visual elements in the client can be TextBox, Datagrid etc. A two-way connection is established such that any changes made to the datasource are reflected immediately in the visual element and vice versa. The advantage using this concept is databinding in .NET can be used to write data driven applications quickly. For .NET data binding allows to write less code with fast execution but still get the work done in the best way. Second advantage are .NET automatically writes a lot of databinding code in the background (by see it in "Windows Generated Code" section), so the administrator does not have to spend time writing code for basic databinding, but still has the flexibility of modifying any code that he would like to. Lastly get the benefits of bound as well as unbound approach.

As a conclusion, these applications just need to improve from time to time because this application system is really easier and make additions functions for administrator to handle the inventory management in laboratory. Based on result that have been archived the administrator that manages the system are easily identify the quantity and type of component that found in the laboratory. In addition the administrator also easily to obtain a list of student who made the requisition

5.1 RECOMMENDATION

Recommendations for this project:

- i. To use Global System for Mobile Communication or GSM in E-ECLAB application. Which GSM function in this application is to enable the administrator to receive message about the requisition that had made by user through mobile phone. It easier for administrator to approve the requisition in any place, for example, when administrator have at a meeting place or a holiday

REFERENCE

- [1] What is inventory management (n.d.).In *BarcodesInc* website.Retrieved from <http://www.barcodesinc.com/articles/what-is-inventory-management.htm>.
- [2] Geoff Relph, FIOM, Inventory Matters Limited, WitekBrzeski(2003). “The first step to inventory management”.*IBM Business Consulting Services, Gail Bradbear, MIOM, IBM Industrial Practice*
- [3] Bonney, M.C., 1994. Trends in inventory management.Interna-tional Journal of Production Economics 25, 107–114.
- [4] S.O.E. Ng, E.C.Partington and D. Sculli, (1992).“A computer system for inventory management of lighting products: A case study”. , copyright 1993 by Elsevier
- [5] M.R. Kesner (1988), *Information Systems: A Strategic Approach to Planning and Information*, American Library Association, Chicago, IL,
- [6] Margaret Rouse (2006). “Definition databease”.In Techtarger website. Retrieved from <http://searchsqlserver.techtarger.com/definition/database>
- [7] “Difference between MS Access and SQL Server”(2006).In ASPfree website. Retrieved from <http://forums.aspfree.com/microsoft-access-help-18/difference-between-ms-access-sql-server-99300.html>
- [8] “Microsoft Access Versus SQL Server” (n.d). in quackit.com website. Retrieved from http://www.quackit.com/microsoft_access/tutorial/microsoft_access_versus_sql_server.com
- [9] Whitten, Jeffrey L.; Lonnie D. Bentley, Kevin C. Dittman. (2003). *Systems Analysis and Design Methods*.6th edition.ISBN 0-256-19906-X.

- [10] Xiao He (2007). "A metamodel for the notation of graphical modeling languages". In: Computer Software and Applications Conference, 2007. COMPSAC 2007 - Vol. 1. 31st Annual International, Volume 1, Issue , 24–27 July 2007, pp 219-224.
- [11] FlowLayoutPanel Class (n.d.). In *msdn* website. Retrieved from <http://msdn.microsoft.com/en-us/library/system.windows.forms.flowlayoutpanel.aspx>.
- [12] Winkelman, Dr. Roy (2009). "Chapter 6: Software", An Educator's Guide to School Networks, 6.
- [13] How to change the listening port for Remote Desktop. Microsoft. January 31, 2007. Retrieved November 2, 2007. Microsoft KB article 306759, revision 2.2.
- [14] Ashih B. Sasankar; Dr. VinayChavan (2011). SWOT analysis of Software Development Process Models. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, September 2011, pp 391-392
- [15] Margaret Rouse (2008). "Definition Construction Software". In WhatIs.com website. Retrieved from <http://whatis.techtarget.com/definition/construction-software>.

APPENDIX A

GANTT CHART OF THE PROJECT (SEMESTER 01)

Project Activities	Weeks														
	September			October				November				December			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Briefing by project supervisor	Planning	Actual													
FYP title confirmation			Planning	Actual											
Verify Project Scope and Objective				Planning	Actual										
Literature Review				Planning	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual			
Related Topic Study					Planning	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual		
Methodology					Planning	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual		
Designing Interface								Planning	Actual	Actual	Actual	Actual	Actual	Actual	Actual
Analysis Database												Planning	Actual	Actual	Actual
Report Writing											Planning	Actual	Actual	Actual	Actual
Preparation on FYP1 Final Presentation												Planning	Actual	Actual	Actual
Pre-Presentation with Supervisor													Planning	Actual	Actual
Final Presentation for FYP1														Planning	Actual

Planning
 Actual

APPENDIX B

GANTT CHART OF THE PROJECT (SEMESTER 02)

Project Activities	Weeks															
	Feb.		Mach				April				Mei				June	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Briefing by project supervisor	Planning	Actual	Actual													
Literature Review (finalize study)			Actual	Actual	Actual	Actual										
Methodology (Testing)				Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual				
Prepare for project proposal									Actual	Actual	Actual	Actual	Actual	Actual		
Methodology (design experiment setup)												Actual	Actual	Actual		
Result and Discussions														Actual	Actual	
Prepare for Final Presentation PSM2															Actual	Actual
Final Presentation PSM2															Actual	Actual
Report for PSM2 and log book															Actual	Actual

Planning
 Actual

APPENDIX C

CODING FOR HOME FORM DESIGN

```
PublicClass a_home
'call library user 32
DeclareAutoFunction SetParent
Lib"user32" _
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr) _
AsInteger

'set clearpanel as clear form
PrivateSub clearpanel()

Me.FlowLayoutPanel1.Controls.Cl
ear()

EndSub
PrivateSub clearpanel1()

Me.FlowLayoutPanel2.Controls.Cl
ear()

EndSub
PrivateSub clearpanel2()

Me.FlowLayoutPanel3.Controls.Cl
ear()

EndSub
PrivateSub clearpanel3()

Me.FlowLayoutPanel4.Controls.Cl
ear()

EndSub

PrivateSub home_Load(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load

'call private sub clearform()
to clean all flowlayoutpanel
```

```
clearpanel()
clearpanel1()
clearpanel2()
clearpanel3()

'make the flowlayoutpanel3 to
sign in form

SetParent(b_sign_in.Handle,
Me.FlowLayoutPanel4.Handle)
    b_sign_in.Location =
New Point(0, 0)
    b_sign_in.Size =
Me.FlowLayoutPanel4.Size
    b_sign_in.TopLevel =
False
Me.FlowLayoutPanel4.Controls.Ad
d(b_sign_in)

    b_sign_in.Show()

'make the flowlayoutpanel3 to
exit in form

SetParent(f_exit.Handle,
Me.FlowLayoutPanel3.Handle)
    f_exit.Location = New
Point(0, 0)
    f_exit.Size =
Me.FlowLayoutPanel3.Size
    f_exit.TopLevel = False
Me.FlowLayoutPanel3.Controls.Ad
d(f_exit)

    f_exit.Show()

EndSub

EndClass
```

APPENDIX D

CODING FOR SIGN IN FORMS DESIGN

```
PublicClass b_sign_in
'call library user 32
DeclareAutoFunction SetParent
Lib"user32" _
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()

a_home.FlowLayoutPanel1.Controls
s.Clear()

EndSub
PrivateSub clearpanel1()

a_home.FlowLayoutPanel2.Controls
s.Clear()

EndSub
PrivateSub clearpanel2()

a_home.FlowLayoutPanel3.Controls
s.Clear()

EndSub
PrivateSub clearpanel3()

a_home.FlowLayoutPanel4.Controls
s.Clear()

EndSub
PrivateSub b_sign_in_Load(ByVal
sender As System.Object, ByVal
e _
As System.EventArgs)
Handles MyBase.Load

EndSub

PrivateSub
LinkLabel1_LinkClicked(ByVal
sender As System.Object, _
ByVal e As
System.Windows.Forms. _
LinkLabelLinkClickedEventArgs)
Handles LinkLabel1.LinkClicked
```

```
'call private sub clear panel
to clean the flowlayoutpanel3
clearpanel()
clearpanel1()
clearpanel2()
clearpanel3()

'make the flowlayoutpanel3 to
sign up form

SetParent(c_sign_up.Handle,
a_home.FlowLayoutPanel4.Handle)
c_sign_up.Location =
New Point(0, 0)
c_sign_up.Size =
a_home.FlowLayoutPanel4.Size
c_sign_up.TopLevel =
False

a_home.FlowLayoutPanel4.Control
s.Add(c_sign_up)

c_sign_up.Show()

c_sign_up.User_detailBindingSou
rce.AddNew()

'make the flowlayoutpanel3 to
exit in form

SetParent(f_1_sign_in.Handle,
a_home.FlowLayoutPanel3.Handle)
f_1_sign_in.Location =
New Point(0, 0)
f_1_sign_in.Size =
a_home.FlowLayoutPanel3.Size
f_1_sign_in.TopLevel =
False

a_home.FlowLayoutPanel3.Control
s.Add(f_1_sign_in)

f_1_sign_in.Show()

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

Try
'user log in
```

```

IfCType(Me.User_detailTableAdap
ter1.login(Me.txt_username.Text
, _
Me.txt_password.Text), Integer)
> 0 Then

'admin log in
'clean all forms in
flowlayoutpanel

                clearpanel()
                clearpanel1()
                clearpanel2()
                clearpanel3()

'show form user_tool at
flowlayoutpanel2

SetParent(e_user_tool.Handle,
a_home.FlowLayoutPanel2.Handle)

e_user_tool.Location = New
Point(0, 0)

e_user_tool.Size =
a_home.FlowLayoutPanel2.Size

e_user_tool.TopLevel = False

a_home.FlowLayoutPanel2.Control
s.Add(e_user_tool)

e_user_tool.Show()

'show form welcome_form at
flowlayoutpanel1

SetParent(f_2_welcome.Handle,
a_home.FlowLayoutPanel3.Handle)

f_2_welcome.Location = New
Point(0, 0)

f_2_welcome.Size =
a_home.FlowLayoutPanel3.Size

f_2_welcome.TopLevel = False

a_home.FlowLayoutPanel3.Control
s.Add(f_2_welcome)

f_2_welcome.Show()

ForEach ctl In Controls
IfTypeOf ctl Is TextBox Then
ctl.Text = ""
Next ctl

'show message box successfully
login user

```

```

MessageBox.Show("Welcome To e-
ECLAB Inventory System")

'admin log in
ElseIf txt_username.Text =
"admin"Then

txt_password.Text = "admin"

'clean all forms in
flowlayoutpanel

                clearpanel()
                clearpanel1()
                clearpanel2()
                clearpanel3()

'show form admin_tool at
flowlayoutpanel1

SetParent(d_admin_tool.Handle,
a_home.FlowLayoutPanel1.Handle)

d_admin_tool.Location = New
Point(0, 0)

d_admin_tool.Size =
a_home.FlowLayoutPanel1.Size

d_admin_tool.TopLevel = False

a_home.FlowLayoutPanel1.Control
s.Add(d_admin_tool)

d_admin_tool.Show()

'show form welcome_form at
flowlayoutpanel1

SetParent(f_2_welcome.Handle,
a_home.FlowLayoutPanel3.Handle)

f_2_welcome.Location = New
Point(0, 0)

f_2_welcome.Size =
a_home.FlowLayoutPanel3.Size

f_2_welcome.TopLevel = False

a_home.FlowLayoutPanel3.Control
s.Add(f_2_welcome)

f_2_welcome.Show()

ForEach ctl In Controls
IfTypeOf ctl Is TextBox Then
ctl.Text = ""

```

```
Next ctl

'show message box for admin

MessageBox.Show("Welcome
admin!")

Else

                MsgBox("Invalid
username or password.")

EndIf

Catch ex As Exception

                MsgBox(ex.ToString)

EndTry

EndSub

EndClass
```

APPENDIX D1

CODING FOR SIGN IN FORMS DESIGN

```
PublicClass c_sign_up

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Try
Me.Validate()
Me.User_detailBindingSource.EndEdit()
Me.TableAdapterManager.UpdateAll(Me._e_ECLABDataSet)
' if success to register
MsgBox("Welcome "& First_NameTextBox.Text & ". Now you can sign in. " & vbNewLine & "Your account : " & vbNewLine & "Username is " & IDTextBox.Text & vbNewLine & "Password is " & IC_No_3TextBox.Text)

Catch ex As Exception
'show error msgbox
MsgBox("Error 1: Create new account. Please see administrator")

EndTry

' clear all text boxes
ForEach ctl In Controls
IfTypeOf ctl Is TextBox Then
ctl.Text = ""
Next ctl

EndSub

EndClass
```

APPENDIXES E

CODING ADMIN TOOL FORMS DESIGN

```
PublicClass d_admin_tool

'call library user 32
DeclareAutoFunction SetParent
Lib"user32"
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
' set clearpanel to clear all
flowlayoutpanel
PrivateSub clearpanel()

a_home.FlowLayoutPanel1.Controls.Clear()

EndSub
PrivateSub clearpanel1()

a_home.FlowLayoutPanel2.Controls.Clear()

EndSub
PrivateSub clearpanel2()

a_home.FlowLayoutPanel3.Controls.Clear()

EndSub
PrivateSub clearpanel3()

a_home.FlowLayoutPanel4.Controls.Clear()

EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click

'call private sub clearform()
to clean the flowlayoutpanel3
    clearpanel()
    clearpanel1()
    clearpanel2()
    clearpanel3()
```

```
'make the flowlayoutpanel4 to
sign in form

SetParent(b_sign_in.Handle,
a_home.FlowLayoutPanel4.Handle)
    b_sign_in.Location =
New Point(0, 0)
    b_sign_in.Size =
a_home.FlowLayoutPanel4.Size
    b_sign_in.TopLevel =
False

a_home.FlowLayoutPanel4.Controls.Add(b_sign_in)

    b_sign_in.Show()

'make the sign in form in
flowlayoutpanel3

SetParent(f_exit.Handle,
a_home.FlowLayoutPanel3.Handle)
    f_exit.Location = New
Point(0, 0)
    f_exit.Size =
a_home.FlowLayoutPanel3.Size
    f_exit.TopLevel = False

a_home.FlowLayoutPanel3.Controls.Add(f_exit)

    f_exit.Show()

EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

'call private sub clearform()
to clean the flowlayoutpanel4
and 2
    clearpanel1()
    clearpanel3()

'make the flowlayoutpanel4 to
sign in form
```



```

SetParent(g_user.Handle,
a_home.FlowLayoutPanel4.Handle)
    g_user.Location = New
Point(0, 0)
    g_user.Size =
a_home.FlowLayoutPanel4.Size
    g_user.TopLevel = False

a_home.FlowLayoutPanel4.Control
s.Add(g_user)

    g_user.Show()

EndSub

PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click

'call private sub clearform()
to clean the flowlayoutpanel3
    clearpanel1()
    clearpanel3()

'make the flowlayoutpanel4 to
sign in form

SetParent(i_stock.Handle,
a_home.FlowLayoutPanel4.Handle)
    i_stock.Location = New
Point(0, 0)
    i_stock.Size =
a_home.FlowLayoutPanel4.Size
    i_stock.TopLevel =
False

a_home.FlowLayoutPanel4.Control
s.Add(i_stock)

    i_stock.Show()

EndSub

PrivateSub Button4_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button4.Click

'call private sub clearform()
to clean the flowlayoutpanel3
    clearpanel1()
    clearpanel3()

```

```

'make the flowlayoutpanel4 to
sign in form

SetParent(h__requisition.Handle
,
a_home.FlowLayoutPanel4.Handle)
    h__requisition.Location
= New Point(0, 0)
    h__requisition.Size =
a_home.FlowLayoutPanel4.Size
    h__requisition.TopLevel
= False

a_home.FlowLayoutPanel4.Control
s.Add(h__requisition)

    h__requisition.Show()

EndSub
EndClass

```

APPENDIXES E1

CODING USER TOOL FORMS DESIGN

```
PublicClass e_user_tool
'call library user 32
DeclareAutoFunction SetParent
Lib"user32"
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()

a_home.FlowLayoutPanel1.Controls
s.Clear()

EndSub
PrivateSub clearpanel1()

a_home.FlowLayoutPanel2.Controls
s.Clear()

EndSub
PrivateSub clearpanel2()

a_home.FlowLayoutPanel3.Controls
s.Clear()

EndSub
PrivateSub clearpanel3()

a_home.FlowLayoutPanel4.Controls
s.Clear()

EndSub

PrivateSub Button4_Click(ByVal
sender As System.Object, ByVal
e _
As System.EventArgs) Handles
Button4.Click

'call private sub clearform()
to clean the flowlayoutpanel3
    clearpanel()
    clearpanel1()
    clearpanel2()
    clearpanel3()
```

```
'make the flowlayoutpanel4 to
sign in form

SetParent(b_sign_in.Handle,
a_home.FlowLayoutPanel4.Handle)
    b_sign_in.Location =
New Point(0, 0)
    b_sign_in.Size =
a_home.FlowLayoutPanel4.Size
    b_sign_in.TopLevel =
False

a_home.FlowLayoutPanel4.Control
s.Add(b_sign_in)

    b_sign_in.Show()

'make the sign in form in
flowlayoutpanel3

SetParent(f_exit.Handle,
a_home.FlowLayoutPanel3.Handle)
    f_exit.Location = New
Point(0, 0)
    f_exit.Size =
a_home.FlowLayoutPanel3.Size
    f_exit.TopLevel = False

a_home.FlowLayoutPanel3.Control
s.Add(f_exit)

    f_exit.Show()

EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click

'call private sub clearform()
to clean the flowlayoutpanel3
    clearpanel()
    clearpanel3()

'make the sign in form in
flowlayoutpanel3
```

```
SetParent(k_application.Handle,
a_home.FlowLayoutPanel4.Handle)
    k_application.Location
= New Point(0, 0)
    k_application.Size =
a_home.FlowLayoutPanel4.Size
    k_application.TopLevel
= False
```

```
a_home.FlowLayoutPanel4.Control
s.Add(k_application)
```

```
    k_application.Show()
```

```
EndSub
```

```
PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click
```

```
'call private sub clearform()
to clean the flowlayoutpanel3
    clearpanel()
    clearpanel3()
```

```
'make the sign in form in
flowlayoutpanel3
```

```
SetParent(j_search_item.Handle,
a_home.FlowLayoutPanel4.Handle)
    j_search_item.Location
= New Point(0, 0)
    j_search_item.Size =
a_home.FlowLayoutPanel4.Size
    j_search_item.TopLevel
= False
```

```
a_home.FlowLayoutPanel4.Control
s.Add(j_search_item)
```

```
    j_search_item.Show()
```

```
EndSub
```

```
PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click
```

```
'call private sub clearform()
to clean the flowlayoutpanel3
    clearpanel()
    clearpanel3()
```

```
'make the sign in form in
flowlayoutpanel3
```

```
SetParent(l_tracking.Handle,
a_home.FlowLayoutPanel4.Handle)
    l_tracking.Location =
New Point(0, 0)
    l_tracking.Size =
a_home.FlowLayoutPanel4.Size
    l_tracking.TopLevel =
False
```

```
a_home.FlowLayoutPanel4.Control
s.Add(l_tracking)
```

```
    l_tracking.Show()
```

```
EndSub
```

```
EndClass
```

APPENDIXES E2

CODING SIGN IN BUTTON FORMS DESIGN

```
PublicClass f_1_sign_in
'call library user 32
DeclareAutoFunction SetParent
Lib"user32"
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()

a_home.FlowLayoutPanel1.Controls
s.Clear()

EndSub
PrivateSub clearpanel1()

a_home.FlowLayoutPanel2.Controls
s.Clear()

EndSub
PrivateSub clearpanel2()

a_home.FlowLayoutPanel3.Controls
s.Clear()

EndSub
PrivateSub clearpanel3()

a_home.FlowLayoutPanel4.Controls
s.Clear()

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, ByVal
e _
As System.EventArgs) Handles
Button1.Click
'call private sub clearform()
to clean the flowlayoutpanel3
    clearpanel()
    clearpanel1()
    clearpanel2()
    clearpanel3()
```

```
'make the flowlayoutpanel4 to
sign in form

SetParent(b_sign_in.Handle,
a_home.FlowLayoutPanel4.Handle)
    b_sign_in.Location =
New Point(0, 0)
    b_sign_in.Size =
a_home.FlowLayoutPanel4.Size
    b_sign_in.TopLevel =
False

a_home.FlowLayoutPanel4.Control
s.Add(b_sign_in)

    b_sign_in.Show()

'make the sign in form in
flowlayoutpanel3

SetParent(f_exit.Handle,
a_home.FlowLayoutPanel3.Handle)
    f_exit.Location = New
Point(0, 0)
    f_exit.Size =
a_home.FlowLayoutPanel3.Size
    f_exit.TopLevel = False

a_home.FlowLayoutPanel3.Control
s.Add(f_exit)

    f_exit.Show()

EndSub
EndClass
```

APPENDIXES E3

CODING EXIT BUTTON FORMS DESIGN

```
PublicClass f_exit  
  
PrivateSub Button1_Click(ByVal sender As System.Object, _  
ByVal e As System.EventArgs) _  
Handles Button1.Click  
    a_home.Dispose()  
EndSub  
EndClass
```

APPENDIXES F

CODING FOR USER DETAIL FORM FOR ADMINISTRATOR

APPLICATION DESIGN

```
PublicClass g_user
'call library user 32 to used
flowlayout
DeclareAutoFunction SetParent
Lib"user32" _
(ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()

Me.FlowLayoutPanel1.Controls.Cl
ear()

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

clearpanel()
'make the flowlayoutpanell1 to
search user
SetParent(g_1_user_search.Handle
e, Me.FlowLayoutPanel1.Handle)
g_1_user_search.Location = New
Point(0, 0)
g_1_user_search.Size =
Me.FlowLayoutPanel1.Size
g_1_user_search.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(g_1_user_search)
g_1_user_search.Show()

EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click

clearpanel()

'make the flowlayoutpanell1 to
add new user
SetParent(g_1_user_add.Handle,
Me.FlowLayoutPanel1.Handle)
```

```
g_1_user_add.Location = New
Point(0, 0)
g_1_user_add.Size =
Me.FlowLayoutPanel1.Size
g_1_user_add.TopLevel = False
Me.FlowLayoutPanel1.Controls.Ad
d(g_1_user_add)

g_1_user_add.Show()
g_1_user_add.User_detailBinding
Source.AddNew()

EndSub

PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click
clearpanel()

'make the flowlayoutpanell1 to
delete user
SetParent(g_1_user_delete.Handle
e, Me.FlowLayoutPanel1.Handle)
g_1_user_delete.Location = New
Point(0, 0)
g_1_user_delete.Size =
Me.FlowLayoutPanel1.Size
g_1_user_delete.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(g_1_user_delete)
g_1_user_delete.Show()

EndSub
EndClass
```

APPENDIXES F1

CODING FOR SEARCH USER DETAIL FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass g_1_user_search

PrivateSub
g_1_user_search_Load(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load

'TODO: This line of code loads
data into the
'_e_ECLABDataSet.user_detail'
table.

Me.User_detailTableAdapter.Fill
(Me._e_ECLABDataSet.user_detail
)

        Panel1.Visible = False
        Panel2.Visible = False
        Panel3.Visible = False
        Panel4.Visible = False

EndSub

PrivateSub
CheckBox1_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox1.CheckedChanged

If CheckBox1.CheckState =
CheckState.Checked Then

CheckBox2.CheckState =
CheckState.Unchecked
EndIf

EndSub

PrivateSub
CheckBox2_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox2.CheckedChanged

If CheckBox2.CheckState =
CheckState.Checked Then
```

```
CheckBox1.CheckState =
CheckState.Unchecked
EndIf

EndSub

PrivateSub
TextBox1_TextChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles TextBox1.TextChanged

If CheckBox1.CheckState =
CheckState.Checked Then
Me.User_detailTableAdapter.Stud
ent_id _

(Me._e_ECLABDataSet.user_detail
, TextBox1.Text)

ElseIf CheckBox2.CheckState =
CheckState.Checked Then
Me.User_detailTableAdapter.Stud
ent_name _

(Me._e_ECLABDataSet.user_detail
, TextBox1.Text)

Else

EndIf

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

Me.User_detailTableAdapter.Fill
(Me._e_ECLABDataSet.user_detail
)

EndSub

PrivateSub
CheckBox3_CheckedChanged _
```

```
(ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
CheckBox3.CheckedChanged

If CheckBox3.CheckState =
CheckState.Checked Then

    Panel1.Visible =
True
    Panel2.Visible =
True
    Panel3.Visible =
True
    Panel4.Visible =
True

ElseIf CheckBox3.CheckState =
CheckState.Unchecked Then

    Panel1.Visible =
False
    Panel2.Visible =
False
    Panel3.Visible =
False
    Panel4.Visible =
False

EndIf
EndSub
EndClass
```


APPENDIXES F2

CODING FOR ADD NEW USER DETAIL FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass g_1_user_add

PrivateSub Button1_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

' add new user
Try
'save all data
Me.Validate()
Me.User_detailBindingSource.End
Edit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

MsgBox("New account
:"& vbNewLine _
&"Username is "& IDTextBox.Text
& vbNewLine _
&"Password is "&
IC_No_3TextBox.Text &
vbNewLine)

Catch ex As Exception
'error for save data
MsgBox("Error 1:
Create new account. Try
again!")

'clear all textbox
ForEach ctl In Controls
IfTypeOf ctl Is TextBox Then
ctl.Text = ""
Next ctl

EndTry

EndSub

PrivateSub
g_1_user_add_Load(ByVal sender
As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load

'TODO: This line of code loads
data into the
```

```
'_e_ECLABDataSet.user_detail'
table.

Me.User_detailTableAdapter.Fill
(Me._e_ECLABDataSet.user_detail
)

Panell.Visible = False

Panel3.Visible = False

User_detailDataGridView.Visible
= False

'clear all textbox
ForEach ctl In Controls
IfTypeOf ctl Is TextBox Then
ctl.Text = ""
Next ctl

EndSub

PrivateSub
CheckBox1_CheckedChanged _
(ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
CheckBox1.CheckedChanged

If CheckBox1.CheckState =
CheckState.Checked Then

Panell.Visible =
True

Panel3.Visible =
True

User_detailDataGridView.Visible
= True

ElseIf CheckBox1.CheckState =
CheckState.Unchecked Then

Panell.Visible =
False
```

```
Panel3.Visible =  
False  
  
User_detailDataGridView.Visible  
= False  
  
EndIf  
EndSub  
  
EndClass
```

APPENDIXES F3

CODING FOR USER DETAIL FORM FOR ADMINISTRATOR

APPLICATION DESIGN

```
PublicClass g_1_user_delete

PrivateSub
g_1_user_delete_Load(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load

        Panel1.Visible = False
        Panel2.Visible = False
        Panel3.Visible = False
        Panel4.Visible = False

EndSub

PrivateSub
CheckBox1_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox1.CheckedChanged

If CheckBox1.CheckState =
CheckState.Checked Then

CheckBox2.CheckState =
CheckState.Unchecked
EndIf

EndSub

PrivateSub
CheckBox2_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox2.CheckedChanged

If CheckBox2.CheckState =
CheckState.Checked Then

CheckBox1.CheckState =
CheckState.Unchecked
EndIf

EndSub
```

```
PrivateSub
TextBox1_TextChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs)
Handles TextBox1.TextChanged

If CheckBox1.CheckState =
CheckState.Checked Then
Me.User_detailTableAdapter.Stud
ent_id _

(Me._e_ECLABDataSet.user_detai
l, TextBox1.Text)

ElseIf CheckBox2.CheckState =
CheckState.Checked Then
Me.User_detailTableAdapter.Stud
ent_name _

(Me._e_ECLABDataSet.user_detai
l, TextBox1.Text)

Else

EndIf

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

If
User_detailDataGridView.Current
Row IsNotNothingThen
If MessageBox.Show("Are you
sure to delete this record?", _
"", MessageBoxButtons.OKCancel)
= _

DialogResult.OK Then

User_detailBindingSource1.Remov
eCurrent()
Me.Validate()
Me.User_detailBindingSource1.En
dEdit()
```

```
Me.TableAdapterManager.UpdateAll(Me._e_ECLABDataSet)

EndIf

EndIf
EndSub

PrivateSub
CheckBox3_CheckedChanged _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
CheckBox3.CheckedChanged

If CheckBox3.CheckState =
CheckState.Checked Then

    Panel1.Visible =
True
    Panel2.Visible =
True
    Panel3.Visible =
True
    Panel4.Visible =
True

ElseIf CheckBox3.CheckState =
CheckState.Unchecked Then

    Panel1.Visible =
False
    Panel2.Visible =
False
    Panel3.Visible =
False
    Panel4.Visible =
False

EndIf
EndSub
EndClass
```

APPENDIXES G

CODING FOR INVENTORY STOCK FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass i_stock
'call library user 32 to used
flowlayout
DeclareAutoFunction SetParent
Lib"user32" _
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()

Me.FlowLayoutPanel1.Controls.Cl
ear()

EndSub
PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

        clearpanel()

'make the flowlayoutpanell to
search user

SetParent(i_1_stock_view.Handle
, Me.FlowLayoutPanel1.Handle)
        i_1_stock_view.Location
= New Point(0, 0)
        i_1_stock_view.Size =
Me.FlowLayoutPanel1.Size
        i_1_stock_view.TopLevel
= False
Me.FlowLayoutPanel1.Controls.Ad
d(i_1_stock_view)

        i_1_stock_view.Show()

EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click

        clearpanel()

'make the flowlayoutpanell to
search user
```

```
SetParent(i_1_stock_update.Hand
le, Me.FlowLayoutPanel1.Handle)

i_1_stock_update.Location = New
Point(0, 0)
        i_1_stock_update.Size =
Me.FlowLayoutPanel1.Size

i_1_stock_update.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(i_1_stock_update)

        i_1_stock_update.Show()

EndSub

PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click

        clearpanel()

'make the flowlayoutpanell to
search user

SetParent(i_1_stock_add_new.Han
dle,
Me.FlowLayoutPanel1.Handle)

i_1_stock_add_new.Location =
New Point(0, 0)
        i_1_stock_add_new.Size
= Me.FlowLayoutPanel1.Size
i_1_stock_add_new.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(i_1_stock_add_new)

i_1_stock_add_new.Show()

i_1_stock_add_new.StockBindingS
ource.AddNew()

EndSub
```

```
PrivateSub Button4_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles Button4.Click

        clearpanel()

        'make the flowlayoutpanell1 to
        search user

        SetParent(i_1_stock_delete.Handle, Me.FlowLayoutPanel1.Handle)

        i_1_stock_delete.Location = New Point(0, 0)
            i_1_stock_delete.Size = Me.FlowLayoutPanel1.Size

        i_1_stock_delete.TopLevel = False
        Me.FlowLayoutPanel1.Controls.Add(i_1_stock_delete)

            i_1_stock_delete.Show()

EndSub
EndClass
```

APPENDIXES G1

CODING FOR VIEW INVENTORY STOCK FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass i_1_stock_view
Dim Quantity AsInteger

PrivateSub
i_1_stock_view_Load(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load
'TODO: This line of code loads
data into the
'_e_ECLABDataSet.Stock' table.
'You can move, or remove it, as
needed.
Me.StockTableAdapter.Fill(Me._e
_ECLABDataSet.Stock)
        TextBox1.Visible =
False
        ComboBox1.Visible =
False

EndSub

PrivateSub
CheckBox1_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox1.CheckedChanged

If CheckBox1.CheckState =
CheckState.Checked Then

CheckBox2.CheckState =
CheckState.Unchecked

CheckBox3.CheckState =
CheckState.Unchecked

CheckBox4.CheckState =
CheckState.Unchecked

CheckBox5.CheckState =
CheckState.Unchecked
        TextBox1.Visible =
True

ElseIf CheckBox1.CheckState =
CheckState.Unchecked Then
```

```
        TextBox1.Visible =
False

Else
EndIf

EndSub
PrivateSub
CheckBox2_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox2.CheckedChanged

If CheckBox2.CheckState =
CheckState.Checked Then

CheckBox1.CheckState =
CheckState.Unchecked

CheckBox3.CheckState =
CheckState.Unchecked

CheckBox4.CheckState =
CheckState.Unchecked

CheckBox5.CheckState =
CheckState.Unchecked

        ComboBox1.Visible =
True

ElseIf CheckBox2.CheckState =
CheckState.Unchecked Then
        ComboBox1.Visible =
False
EndIf

EndSub

PrivateSub
TextBox1_TextChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles TextBox1.TextChanged

If CheckBox1.CheckState =
CheckState.Checked Then
```

```

Me.StockTableAdapter.Code (Me._e
_eCLABDataSet.Stock,
TextBox1.Text)

Else

EndIf

EndSub

PrivateSub Button1_Click (ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

'TODO: This line of code loads
data into the
'_e_eCLABDataSet.Stock' table.
'You can move, or remove it, as
needed.
Me.StockTableAdapter.Fill (Me._e
_eCLABDataSet.Stock)

EndSub

PrivateSub
CheckBox3_CheckedChanged (ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox3.CheckedChanged

If CheckBox3.CheckState =
CheckState.Checked Then

CheckBox2.CheckState =
CheckState.Unchecked

CheckBox1.CheckState =
CheckState.Unchecked

CheckBox4.CheckState =
CheckState.Unchecked

CheckBox5.CheckState =
CheckState.Unchecked

If Quantity << 50 Then
Me.StockTableAdapter.Quantity _
(Me._e_eCLABDataSet.Stock, New
System.Nullable _
(OfInteger) (CType (Quantity,
Integer)))
EndIf
EndIf
EndSub

```

```

PrivateSub
CheckBox4_CheckedChanged (ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox4.CheckedChanged

If CheckBox4.CheckState =
CheckState.Checked Then

CheckBox2.CheckState =
CheckState.Unchecked

CheckBox3.CheckState =
CheckState.Unchecked

CheckBox1.CheckState =
CheckState.Unchecked

CheckBox5.CheckState =
CheckState.Unchecked

If Quantity << 100 Then
Me.StockTableAdapter.Quantity _
(Me._e_eCLABDataSet.Stock, New
System.Nullable _
(OfInteger) (CType (Quantity,
Integer)))
EndIf
EndIf
EndSub

PrivateSub
CheckBox5_CheckedChanged (ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox5.CheckedChanged

If CheckBox5.CheckState =
CheckState.Checked Then

CheckBox2.CheckState =
CheckState.Unchecked

CheckBox3.CheckState =
CheckState.Unchecked

CheckBox4.CheckState =
CheckState.Unchecked

CheckBox1.CheckState =
CheckState.Unchecked
EndIf

If Quantity >> 101 Then
Me.StockTableAdapter.Quantity _

```



```
(Me._e_ECLABDataSet.Stock, New
System.Nullable _

(OfInteger) (CType (Quantity,
Integer)))
EndIf

EndSub

PrivateSub
ComboBox1_SelectedIndexChanged
-
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
ComboBox1.SelectedIndexChanged

If CheckBox2.CheckState =
CheckState.Checked Then
Me.StockTableAdapter.Company (Me
._e_ECLABDataSet.Stock, _

ComboBox1.Text)
EndIf
EndSub
EndClass
```

APPENDIXES G2

CODING FOR ADD NEW INVENTORY STOCK FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass i_1_stock_add_new

PrivateSub
i_1_stock_add_new_Load_1(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load
'TODO: This line of code loads
data into the
'_e_ECLABDataSet.component'
table.
Me.ComponentTableAdapter.Fill(M
e._e_ECLABDataSet.component)
'TODO: This line of code loads
data into the
'_e_ECLABDataSet.Stock' table.
Me.StockTableAdapter.Fill(Me._e
_ECLABDataSet.Stock)

EndSub

PrivateSub Button2_Click _
(ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles Button2.Click

Try
Me.Validate()
Me.StockBindingSource.EndEdit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

MsgBox("Data
save!")

Catch ex As Exception

MsgBox("Error: for
save data")
EndTry
EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click
```

```
Me.StockBindingSource.AddNew()

EndSub

PrivateSub
Last_quantityNumericUpDown_Valu
eChanged _
(ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
Last_quantityNumericUpDown.Valu
eChanged

Current_quantityNumericUpDown.V
alue =
Last_quantityNumericUpDown.Valu
e

EndSub

PrivateSub
Last_dateDateTimePicker_ValueCh
anged _
(ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
Last_dateDateTimePicker.ValueCh
anged

Current_dateDateTimePicker.Valu
e =
Last_dateDateTimePicker.Value

EndSub

EndClass
```

APPENDIXES G3

CODING FOR UPDATE INVENTORY STOCK FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass i_1_stock_update
'call library user 32 to used
flowlayout
DeclareAutoFunction SetParent
Lib"user32"
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()

Me.FlowLayoutPanel1.Controls.Cl
ear()

EndSub
PrivateSub Button1_Click(ByVal
sender As System.Object,
ByVal e As System.EventArgs)
Handles Button1.Click

        clearpanel()

'make the flowlayoutpanell1 to
search user

SetParent(i_1_stock_update_in.H
andle,
Me.FlowLayoutPanel1.Handle)

i_1_stock_update_in.Location =
New Point(0, 0)

i_1_stock_update_in.Size =
Me.FlowLayoutPanel1.Size

i_1_stock_update_in.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(i_1_stock_update_in)

i_1_stock_update_in.Show()

EndSub

PrivateSub Button2_Click(ByVal
sender AsObject,
ByVal e As System.EventArgs)
Handles Button2.Click
```

```
        clearpanel()

'make the flowlayoutpanell1 to
search user

SetParent(i_1_stock_update_out.
Handle,
Me.FlowLayoutPanel1.Handle)

i_1_stock_update_out.Location =
New Point(0, 0)

i_1_stock_update_out.Size =
Me.FlowLayoutPanel1.Size

i_1_stock_update_out.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(i_1_stock_update_out)

i_1_stock_update_out.Show()

EndSub

EndClass
```

APPENDIXES G4

CODING FOR UPDATE STOCK IN FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass i_1_stock_update_in

PrivateSub
CheckBox1_CheckedChanged (ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
CheckBox1.CheckedChanged
If CheckBox1.CheckState =
CheckState.Checked Then

CheckBox2.CheckState =
CheckState.Unchecked
TextBox1.Visible =
True

ElseIf CheckBox1.CheckState =
CheckState.Unchecked Then
TextBox1.Visible =
False

EndIf
EndSub

PrivateSub
CheckBox2_CheckedChanged (ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
CheckBox2.CheckedChanged
If CheckBox2.CheckState =
CheckState.Checked Then

CheckBox1.CheckState =
CheckState.Unchecked
ComboBox1.Visible =
True

ElseIf CheckBox2.CheckState =
CheckState.Unchecked Then
ComboBox1.Visible =
False

EndIf
EndSub

PrivateSub
TextBox1_TextChanged (ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
TextBox1.TextChanged
```

```
Me.StockTableAdapter.Code (Me._e
_ECLABDataSet.Stock,
TextBox1.Text)

EndSub

PrivateSub
ComboBox1_SelectedIndexChanged (
ByVal sender As System.Object,
ByVal e As System.EventArgs)
Handles
ComboBox1.SelectedIndexChanged

Me.StockTableAdapter.component (
Me._e_ECLABDataSet.Stock,
ComboBox1.Text)

EndSub

PrivateSub Button2_Click (ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
Button2.Click

Me.StockTableAdapter.Fill (Me._e
_ECLABDataSet.Stock)

EndSub

PrivateSub Button1_Click (ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
Button1.Click

Try
Me.Validate ()
Me.StockBindingSource.EndEdit ()
Me.TableAdapterManager.UpdateAll
(Me._e_ECLABDataSet)

MsgBox ("data has
been updated")
Catch ex As Exception
MsgBox ("error for
updating the selected item.")
EndTry

EndSub
```

```

PrivateSub
i_1_stock_update_in_Load_1 _
  (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles MyBase.Load

'TODO: This line of code loads
data into the
'_e_ECLABDataSet.Stock' table.
You can move, or remove it, as
needed.
Me.StockTableAdapter.Fill(Me._e
_ECLABDataSet.Stock)

        TextBox1.Visible =
False
        ComboBox1.Visible =
False

Last_quantityNumericUpDown.Valu
e =
Current_quantityNumericUpDown.V
alue

Last_dateDateTimePicker.Value =
Current_dateDateTimePicker.Valu
e

EndSub

PrivateSub
Quantity_inNumericUpDown_ValueC
hanged _
  (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
Quantity_inNumericUpDown.ValueC
hanged

Current_quantityNumericUpDown.V
alue = _

Last_quantityNumericUpDown.Valu
e +
Quantity_inNumericUpDown.Value

EndSub

PrivateSub
Date_inDateTimePicker_ValueChan
ged _

```

```

  (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
Date_inDateTimePicker.ValueChan
ged

Date_inDateTimePicker.Value =
Current_dateDateTimePicker.Valu
e

EndSub
EndClass

```

APPENDIXES G5

CODING FOR UPDATE STOCK OUT FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass
i_1_stock_update_out

PrivateSub
StockBindingNavigatorSaveItem_C
lick _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs)

Me.Validate()
Me.StockBindingSource.EndEdit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

EndSub

PrivateSub
i_1_stock_update_out_Load(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load

Quantity_outNumericUpDown.Reset
Text()

Date_outDateTimePicker.Update()

    Panel1.Visible = False
    Panel2.Visible = False
    Panel3.Visible = False

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

Last_quantityTextBox.Text =
Current_quantityTextBox.Text

Last_dateDateTimePicker.Value =
Current_dateDateTimePicker.Valu
e
```

```
Current_dateDateTimePicker.Valu
e =
Date_outDateTimePicker.Value()

Try

Me.Validate()
Me.StockBindingSource.EndEdit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

        MsgBox("Stock has
been updated!")
Catch ex As Exception

'error for save data
        MsgBox("Error 1:
Create new account. Try
again!")

'clear all textbox
ForEach ctl In Controls
IfTypeOf ctl Is TextBox Then
ctl.Text = ""
Next ctl

EndTry
EndSub

PrivateSub
TextBox1_TextChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles TextBox1.TextChanged

Me.StockTableAdapter.Code(Me._e
_ECLABDataSet.Stock,
TextBox1.Text)

EndSub

PrivateSub
CheckBox1_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox1.CheckedChanged
```

```
If CheckBox1.CheckState =  
CheckState.Checked Then  
  
    Panel1.Visible =  
True  
  
    Panel2.Visible =  
True  
  
    Panel3.Visible =  
True  
  
ElseIf CheckBox1.CheckState =  
CheckState.Unchecked Then  
  
    Panel1.Visible =  
False  
  
    Panel2.Visible =  
False  
  
    Panel3.Visible =  
False  
  
EndIf  
  
EndSub  
EndClass
```

APPENDIXES G6

CODING FOR DELETE INVENTORY STOCK FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass i_1_stock_delete

PrivateSub
StockBindingNavigatorSaveItem_C
lick_
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs)

Me.Validate()
Me.StockBindingSource.EndEdit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

EndSub

PrivateSub
StockBindingNavigatorSaveItem_C
lick_1_
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs)

Me.Validate()
Me.StockBindingSource.EndEdit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

EndSub

PrivateSub
i_1_stock_delete_Load(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load

'TODO: This line of code loads
data into the
'_e_ECLABDataSet.Stock' table.
'You can move, or remove it, as
needed.
Me.StockTableAdapter.Fill(Me._e
_ECLABDataSet.Stock)

    Panel1.Visible = False
    Panel2.Visible = False
    Panel3.Visible = False
    Panel4.Visible = False
```

```
EndSub

PrivateSub
CodeToolStripButton_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs)
Try
Me.StockTableAdapter.Code(Me._e
_ECLABDataSet.Stock,
TextBox1.Text)
Catch ex As System.Exception

System.Windows.Forms.MessageBox
.Show(ex.Message)
EndTry

EndSub

PrivateSub
ComponentToolStripButton_Click(
ByVal sender As System.Object,
ByVal e As System.EventArgs)
Try
Me.StockTableAdapter.component(
Me._e_ECLABDataSet.Stock, _
TextBox1.Text)
Catch ex As System.Exception

System.Windows.Forms.MessageBox
.Show(ex.Message)
EndTry

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

If StockDataGridView.CurrentRow
IsNotNothingThen
If MessageBox.Show("Are you
sure to delete this record?", _
"", MessageBoxButtons.OKCancel)
= _
DialogResult.OK Then
```



```
StockBindingSource.RemoveCurrent()  
Me.Validate()  
Me.StockBindingSource.EndEdit()  
Me.TableAdapterManager.UpdateAll(Me._e_ECLABDataSet)  
  
EndIf  
EndIf  
  
EndSub  
  
PrivateSub  
CheckBox3_CheckedChanged(ByVal sender As System.Object, _  
ByVal e As System.EventArgs) _  
Handles  
CheckBox3.CheckedChanged  
  
If CheckBox3.CheckState =  
CheckState.Checked Then  
  
        Panel1.Visible =  
True  
  
        Panel2.Visible =  
True  
  
        Panel3.Visible =  
True  
  
        Panel4.Visible =  
True  
  
ElseIf CheckBox3.CheckState =  
CheckState.Unchecked Then  
  
        Panel1.Visible =  
False  
  
        Panel2.Visible =  
False  
  
        Panel3.Visible =  
False  
  
        Panel4.Visible =  
False  
  
EndIf  
  
EndSub  
EndClass
```

APPENDIXES H

CODING FOR USER REQUISITION FORM FOR ADMINISTRATOR

APPLICATION DESIGN

```
PublicClass h__requisition
'call library user 32 to used
flowlayout
DeclareAutoFunction SetParent
Lib"user32" _
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()
Me.FlowLayoutPanel1.Controls.Cl
ear()
EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click
    clearpanel()
'make the flowlayoutpanell1 to
search user

SetParent(h_requisition_check.H
andle,
Me.FlowLayoutPanel1.Handle)
h_requisition_check.Location =
New Point(0, 0)

h_requisition_check.Size =
Me.FlowLayoutPanel1.Size
h_requisition_check.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(h_requisition_check)

h_requisition_check.Show()
EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click
clearpanel()
'make the flowlayoutpanell1 to
search user

SetParent(h_1_requisition_add.H
andle,
Me.FlowLayoutPanel1.Handle)
h_1_requisition_add.Location =
New Point(0, 0)
```

```
h_1_requisition_add.Size =
Me.FlowLayoutPanel1.Size
h_1_requisition_add.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(h_1_requisition_add)

h_1_requisition_add.Show()

h_1_requisition_add.Requisition
BindingSource.AddNew()
EndSub

PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click

    clearpanel()
'make the flowlayoutpanell1 to
search user
SetParent(h_1_requisition_delet
e.Handle,
Me.FlowLayoutPanel1.Handle)
h_1_requisition_delete.Location
= New Point(0, 0)
h_1_requisition_delete.Size =
Me.FlowLayoutPanel1.Size
h_1_requisition_delete.TopLevel
= False
Me.FlowLayoutPanel1.Controls.Ad
d(h_1_requisition_delete)

h_1_requisition_delete.Show()

EndSub
EndClass
```

APPENDIXES H1

CODING FOR CHECK USER REQUISITION FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass h_requisition_check

PrivateSub
h_requisition_check_Load _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles MyBase.Load

'TODO: This line of code loads
data into the
'_e_ECLABDataSet.Requisition'
table.
'You can move, or remove it, as
needed.
Me.RequisitionTableAdapter.Fill
(Me._e_ECLABDataSet.Requisition
)

EndSub

PrivateSub
StatusCheckBox_CheckedChanged _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
StatusCheckBox.CheckedChanged

If StatusCheckBox.CheckState =
CheckState.Checked Then
Me.PictureBox1.Image =
My.Resources.okey___Copy
Else
Me.PictureBox1.Image =
My.Resources.pending

EndIf

EndSub

PrivateSub TextBox1_TextChanged
_
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles TextBox1.TextChanged
```

```
Me.StockTableAdapter.Code(Me._e
_ECLABDataSet.Stock,
TextBox1.Text)

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

Me.RequisitionBindingSource.Mov
eNext()

EndSub

PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click

Me.RequisitionBindingSource.Mov
ePrevious()

EndSub

PrivateSub Button4_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button4.Click

Me.RequisitionBindingSource.Mov
eFirst()

EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click

Me.RequisitionBindingSource.Mov
eLast()

EndSub

PrivateSub
TextBox2_TextChanged (ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
```

```
Handles TextBox2.TextChanged

Try
Me.RequisitionBindingSource.Position = _
Me.RequisitionBindingSource.Find("ID", TextBox2.Text)
Catch ex As Exception
    MsgBox("Error!
Please select No. of the
requisition.")
EndTry
EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button5.Click

'TODO: This line of code loads
data into the
'_e_ECLABDataSet.Stock' table.
'You can move, or remove it, as
needed.
Me.StockTableAdapter.Fill(Me._e
_ECLABDataSet.Stock)

EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button6.Click

Me.RequisitionTableAdapter.Fill
(Me._e_ECLABDataSet.Requisition
)

EndSub
EndClass
```

APPENDIXES H2

CODING FOR ADD NEW USER REQUISITION FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass h_1_requisition_add
Dim fontencoder AsNew
IDAutomation.NetAssembly.FontEncoder
encoder

PrivateSub
h_1_requisition_add_Load(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load

        Panell1.Visible = False
EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

        ' add new stock
Try
'save all data
Me.Validate()
Me.RequisitionBindingSource.End
Edit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

        MsgBox("New Stock
Save!")

Catch ex As Exception
'error for save data
        MsgBox("Error 1:
Create new account. Try
again!")

'clear all textbox
ForEach ctl In Controls
IfTypeOf ctl Is TextBox Then
ctl.Text = ""
Next ctl

EndTry
EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click
```

```
Me.RequisitionBindingSource.Add
New()

Type_RequisitionTextBox.Text =
"Request"

EndSub

PrivateSub
TextBox1_TextChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles TextBox1.TextChanged

Me.StockTableAdapter.Code(Me._e
_ECLABDataSet.Stock,
TextBox1.Text)
Me.StockTableAdapter.component(
Me._e_ECLABDataSet.Stock,
TextBox1.Text)

EndSub

PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click

Me.RequisitionTableAdapter.Fill
(Me._e_ECLABDataSet.Requisition
)

EndSub

PrivateSub
CheckBox1_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox1.CheckedChanged
```

```
If CheckBox1.CheckState =  
CheckState.Checked Then  
  
    Panell1.Visible =  
True  
  
ElseIf CheckBox1.CheckState =  
CheckState.Unchecked Then  
  
    Panell1.Visible =  
False  
  
EndIf  
  
EndSub  
  
PrivateSub Button4_Click(ByVal  
sender As System.Object, _  
ByVal e As System.EventArgs) _  
Handles Button4.Click  
  
Dim a AsString  
  
    a = (IDTextBox.Text)  
  
    BarcodeTextBox.Text =  
fontencoder.Code39(a)  
  
    BarcodeTextBox.Font =  
New System.Drawing.Font _  
("IDAutomationSC39M",  
18, FontStyle.Regular)  
  
EndSub  
EndClass
```

APPENDIXES H3

CODING FOR DELETE USER REQUISITION FORM FOR ADMINISTRATOR APPLICATION DESIGN

```
PublicClass
h_1_requisition_delete

PrivateSub
RequisitionBindingNavigatorSave
Item_Click _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs)
Me.Validate()
Me.RequisitionBindingSource.End
Edit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

EndSub

PrivateSub
h_1_requisition_delete_Load _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles MyBase.Load

        Panel3.Visible = False
        Panel2.Visible = False
        Panel4.Visible = False
        Panel5.Visible = False

        TextBox1.Visible =
False
        DateTimePicker1.Visible
= False

EndSub

PrivateSub
CheckBox3_CheckedChanged _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
CheckBox3.CheckedChanged

If CheckBox3.CheckState =
CheckState.Checked Then

        Panel2.Visible =
True
```

```
        Panel3.Visible =
True
        Panel4.Visible =
True
        Panel5.Visible =
True
ElseIf CheckBox3.CheckState =
CheckState.Unchecked Then

        Panel3.Visible =
False
        Panel2.Visible =
False
        Panel4.Visible =
False
        Panel5.Visible =
False

EndIf
EndSub

PrivateSub TextBox1_TextChanged
_
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles TextBox1.TextChanged

If CheckBox2.CheckState =
CheckState.Checked Then

Me.RequisitionTableAdapter.Barcode _
    (Me._e_ECLABDataSet.Requisition
, TextBox1.Text)

Me.RequisitionTableAdapter.Code
_subject _
    (Me._e_ECLABDataSet.Requisition
, TextBox1.Text)

Me.RequisitionTableAdapter.ID _
    (Me._e_ECLABDataSet.Requisition
, TextBox1.Text)
```

```

Me.RequisitionTableAdapter.Subject _
    (Me._e_ECLABDataSet.Requisition
, TextBox1.Text)

Me.RequisitionTableAdapter.Type _
    (Me._e_ECLABDataSet.Requisition
, TextBox1.Text)

Else

EndIf

EndSub

PrivateSub
CheckBox1_CheckedChanged _
    (ByVal sender As _
System.Object, ByVal e As
System.EventArgs) _
Handles
CheckBox1.CheckedChanged

If CheckBox1.CheckState =
CheckState.Checked Then

CheckBox2.CheckState =
CheckState.Unchecked
    TextBox1.Visible =
True

DateTimePicker1.Visible = False

ElseIf CheckBox1.CheckState =
CheckState.Unchecked Then
    TextBox1.Visible =
False

Else

EndIf

EndSub

PrivateSub
CheckBox2_CheckedChanged _
    (ByVal sender As _
System.Object, ByVal e As
System.EventArgs) _
Handles
CheckBox2.CheckedChanged

If CheckBox2.CheckState =
CheckState.Checked Then

```

```

CheckBox1.CheckState =
CheckState.Unchecked
    TextBox1.Visible =
False

DateTimePicker1.Visible = True

ElseIf CheckBox2.CheckState =
CheckState.Unchecked Then

DateTimePicker1.Visible = False

Else

EndIf

EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click

'TODO: This line of code loads
data into the
'_e_ECLABDataSet.Requisition'
table.

Me.RequisitionTableAdapter.Fill
(Me._e_ECLABDataSet.Requisition
)

EndSub

PrivateSub
DateTimePicker1_ValueChanged _
    (ByVal sender As _
System.Object, ByVal e As
System.EventArgs) _
Handles
DateTimePicker1.ValueChanged

If CheckBox3.CheckState =
CheckState.Checked Then

Me.RequisitionTableAdapter._dat
e _
    (Me._e_ECLABDataSet.Requisition
, New System.Nullable _
(OfDate) (CType(DateTimePicker1.
Value, Date)))

Else

EndIf

EndSub

```



```
PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

    If RequisitionDataGridView.CurrentRow IsNot Nothing Then
        If MessageBox.Show("Are you sure to delete this record?", "", MessageBoxButtons.OKCancel) = DialogResult.OK Then

            RequisitionBindingSource.RemoveCurrent()
            Me.Validate()
            Me.RequisitionBindingSource.EndEdit()
            Me.TableAdapterManager.UpdateAll(Me._e_ECLABDataSet)

        EndIf
    EndIf
EndSub
EndClass
```

APPENDIXES I

CODING FOR SEARCH ITEM FORM FOR USER APPLICATION DESIGN

```
PublicClass j_search_item
'call library user 32 to used
flowlayout
DeclareAutoFunction SetParent
Lib"user32" _
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()

Me.FlowLayoutPanel1.Controls.Clear()
EndSub

PrivateSub
j_search_item_Load(ByVal sender
As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load

'TODO: This line of code loads
data into the
'_e_ECLABDataSet.Stock' table.
Me.StockTableAdapter.Fill(Me._e
_ECLABDataSet.Stock)

        clearpanel()

EndSub

PrivateSub
ComboBox1_SelectedIndexChanged
_
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
ComboBox1.SelectedIndexChanged

Me.StockTableAdapter.component(
Me._e_ECLABDataSet.Stock, _

ComboBox1.SelectedValue)

EndSub
```

```
PrivateSub
CheckBox1_CheckedChanged(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles
CheckBox1.CheckedChanged

If CheckBox1.CheckState =
CheckState.Checked Then

'make the flowlayoutpanell to
search user

SetParent(j_search_cart.Handle,
Me.FlowLayoutPanel1.Handle)

j_search_cart.Location = New
Point(0, 0)
        j_search_cart.Size
= Me.FlowLayoutPanel1.Size

j_search_cart.TopLevel = False
Me.FlowLayoutPanel1.Controls.Add(j_search_cart)

j_search_cart.Show()

j_search_cart.CartBindingSource
.AddNew()

ElseIf CheckBox1.CheckState =
CheckState.Unchecked Then

        clearpanel()

Else

EndIf
EndSub

EndClass
```

APPENDIXES J

CODING FOR REQUISITION FORM FOR USER APPLICATION DESIGN

```
PublicClass k_application
'call library user 32 to used
flowlayout
DeclareAutoFunction SetParent
Lib"user32"
    (ByVal hWndChild As IntPtr,
    ByVal hWndParent As IntPtr)
AsInteger
PrivateSub clearpanel()

Me.FlowLayoutPanel1.Controls.Cl
ear()

EndSub
PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

        clearpanel()

'make the flowlayoutpanell1 to
search user

SetParent(k_1_application_reque
st.Handle,
Me.FlowLayoutPanel1.Handle)

k_1_application_request.Locatio
n = New Point(0, 0)

k_1_application_request.Size =
Me.FlowLayoutPanel1.Size

k_1_application_request.TopLeve
l = False
Me.FlowLayoutPanel1.Controls.Ad
d(k_1_application_request)

k_1_application_request.Show()

k_1_application_request.Requisi
tionBindingSource.AddNew()

EndSub
```

```
PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click

        clearpanel()

'make the flowlayoutpanell1 to
search user

SetParent(k_1_application_new.H
andle,
Me.FlowLayoutPanel1.Handle)

k_1_application_new.Location =
New Point(0, 0)

k_1_application_new.Size =
Me.FlowLayoutPanel1.Size

k_1_application_new.TopLevel =
False
Me.FlowLayoutPanel1.Controls.Ad
d(k_1_application_new)

k_1_application_new.Show()

k_1_application_new.Requisition
BindingSource.AddNew()

EndSub
EndClass
```

APPENDIXES J1

CODING FOR REQUEST APPLICATION FOR USER APPLICATION

DESIGN

```
PublicClass
k_1_application_request
'call library user 32 to used
flowlayout
DeclareAutoFunction SetParent
Lib"user32" _
    (ByVal hWndChild As IntPtr,
ByVal hWndParent As IntPtr)
AsInteger
Dim fontencoder AsNew
IDAutomation.NetAssembly.FontEn
coder
PrivateSub clearpanel()

Me.FlowLayoutPanel1.Controls.Cl
ear()

EndSub

PrivateSub
RequisitionBindingNavigatorSave
Item_Click _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs)

Me.Validate()
Me.RequisitionBindingSource.End
Edit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

EndSub

PrivateSub
k_1_application_request_Load _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles MyBase.Load

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs)
```

```
RequisitionBindingSource.AddNew
()

Type_RequisitionTextBox.Text =
"Request"

EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click

Try
Me.Validate()
Me.RequisitionBindingSource.End
Edit()
Me.TableAdapterManager.UpdateAl
l(Me._e_ECLABDataSet)

    MsgBox("Your
requisitions form has been
stored in our system." _
& vbNewLine & _
"Please check your email to
print out the copy in PDF" _
& vbNewLine & vbTab &"Thank
you")

Catch ex As Exception

    MsgBox("Error!.
Your forms not submitted.")

EndTry
EndSub

PrivateSub
CheckBox1_CheckedChanged _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs) _
Handles
CheckBox1.CheckedChanged

If CheckBox1.CheckState =
CheckState.Checked Then
```

```
'make the flowlayoutpanell1 to
search user

SetParent(j_search_cart.Handle,
Me.FlowLayoutPanell1.Handle)

j_search_cart.Location = New
Point(0, 0)
        j_search_cart.Size
= Me.FlowLayoutPanell1.Size

j_search_cart.TopLevel = False
Me.FlowLayoutPanell1.Controls.Ad
d(j_search_cart)

j_search_cart.Show()

ElseIf CheckBox1.CheckState =
CheckState.Unchecked Then

        clearpanel()

Else

EndIf
EndSub

PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click

Dim a AsString

        a = (IDTextBox.Text)

        BarcodeTextBox.Text =
fontencoder.Code39(a)

        BarcodeTextBox.Font =
New System.Drawing.Font _
("IDAutomationSC39S",
18, FontStyle.Regular)

EndSub
EndClass
```

APPENDIXES J2

CODING FOR REQUEST NEW ITEM FORM FOR USER APPLICATION

DESIGN

```
PublicClass k_1_application_new
Dim fontencoder AsNew
IDAutomation.NetAssembly.FontEncoder

PrivateSub
RequisitionBindingNavigatorSave
Item_Click _
    (ByVal sender As
System.Object, ByVal e As
System.EventArgs)

Me.Validate()
Me.RequisitionBindingSource.End
Edit()
Me.TableAdapterManager.UpdateAll
(Me._e_ECLABDataSet)

EndSub

PrivateSub Button1_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button1.Click

Me.RequisitionBindingSource.Add
New()

Type_RequisitionTextBox.Text =
"New Item"
EndSub

PrivateSub Button2_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button2.Click

    PrintForm1.PrintAction
=
Printing.PrintAction.PrintToPre
view

    PrintForm1.Print(Me,
PowerPacks.Printing.PrintForm.
_
PrintOption.Scrollable)

Try
```

```
Me.Validate()
Me.RequisitionBindingSource.End
Edit()
Me.TableAdapterManager.UpdateAll
(Me._e_ECLABDataSet)

        MsgBox("Your
requisitions form has been
stored in our system." _
& vbNewLine & _
"Please check your email to
print out the copy in PDF" _
& vbNewLine & vbTab &"Thank
you")

Catch ex As Exception

        MsgBox("Error!.
Your forms not submitted.")
EndTry
EndSub

PrivateSub Button3_Click(ByVal
sender As System.Object, _
ByVal e As System.EventArgs) _
Handles Button3.Click

Dim a AsString

        a = (IDTextBox.Text)

        BarcodeTextBox.Text =
fontencoder.Code39(a)

        BarcodeTextBox.Font =
New System.Drawing.Font _
("IDAutomationSC39M",
18, FontStyle.Regular)

EndSub
EndClass
```

APPENDIXES K

CODING FOR TRACKING ITEM FORM FOR USER APPLICATION DESIGN

```
PublicClass l_tracking

PrivateSub l_tracking_Load(ByVal sender As System.Object, _
ByVal e As System.EventArgs) _
Handles MyBase.Load
Me.RequisitionTableAdapter.Fill(Me._e_ECLABDataSet.Requisition)
EndSub

PrivateSub TextBox1_TextChanged(ByVal sender As System.Object, _
ByVal e As System.EventArgs) _
Handles TextBox1.TextChanged

Me.RequisitionTableAdapter.ID(Me._e_ECLABDataSet.Requisition, _
TextBox1.Text)

EndSub

PrivateSub StatusCheckBox_CheckedChanged _
(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles StatusCheckBox.CheckedChanged

If StatusCheckBox.CheckState = CheckState.Checked Then
Me.PictureBox1.Image = My.Resources.okey___Copy
Else
Me.PictureBox1.Image = My.Resources.pending
EndIf

EndSub

EndClass
```