

**ON-LINE CUSTOMER SERVICE FOR CARGO STORES
OF
SRI LANKA PORTS AUTHORITY**

BY

J.J.D. KALUARACHCHI

REG.NO. 00/AS/014

Dissertation submitted in partial fulfillment of the requirement for the degree of bachelor of sciences in physical science, faculty of applied sciences, Sabaragamuwa university of Sri Lanka. Buttala, 2004.

DECLARATION

I certify that this dissertation has not been previously submitted for another degree or diploma in any university. And to the best of my knowledge, it does not contain any material previously published or written or orally communicated by another person except where the references made in the text.

.....*J.J.D.*.....

J.J.D. Kaluarachchi

.....7/5/2004.....

Date

Certified by,

External supervisor,

Mr. G.A. Dharmadasa

Senior System Analyst

Information System Division

Sri Lanka Ports Authority

Colombo

.....*G.A. Dharmadasa*.....

Signature

.....7/5/2004.....

Date

Internal supervisor

Dr. R.G.N. Meegama

Center for computer studies

Sabaragamuwa university of Sri Lanka

P.O.Box 2, Belihuloya.

.....*R.G.N. Meegama*.....

Signature

.....08/05/2004.....

Date

Department head

Dr. Nirmalee Wicramaratane

Head/Department of physical sciences

Faculty of applied sciences

Sabaragamuwa university of Sri Lanka

.....*Nirmalee Wicramaratane*.....

Signature

.....12/05/2004.....

date

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge assistance and contributions of number of person to this effort.

First I would like to thank to the relevant management of SLPA for giving me the opportunity to carry out this project.

Also I would like to express my sincere appreciation to Mr.G.A.Dharmadasa and Dr.R.G.N. Meegama for encouraging me from their guidance by appointed as my external and internal supervisors in this effort.

Also I would like to extend my thanks to all the staff members in information system division of SLPA including Mss. Apsara E. and Mr. V. A. Thilakaratna, who were always willing to give their assistances to me throughout the training period, as well as employees in this specific warehouse, who provided me details through discussing the manual process.

My deepest gratitude is expressed to Dr.D.B.M.Wickramaratne, dean of the faculty and Dr.Nirmalee Wickramaratne, head of the department, for their generous assistance.

My lecturers are the shadows; lie behind this effort I would like to take this opportunity to gratitude them as well.

ABSTRACT

Warehouse management was one of the main services provided by Sri Lanka ports authority, as the main port in Sri Lanka.

Developing a warehouse management system for a selected warehouse, and online relevant data was important to make the working process more efficient.

Therefore the software development process was done, by following the leaner sequential model. According to the model analysis was carried out by consulting relevant employees. And then in designing stage, relevant Data Flow Diagrams were designed relevant to the structured method and level them up to 3rd, also the database was designed using relational model and the normalization was done up to 3rd normal form. Further the user interfaces were designed using VB 6.0. The database was implemented using MS access. And DAO method was used to access the database through user interfaces. Further more for achieving the task of online, DSN was assigned to the database, so then server side scripting language like ASP was run on IIS web-server.

The outcome of the software project was accepted by relevant employees of Sri Lanka Ports Authority, as a method of increasing the efficiency, and reducing corruption, as well as the best solution for the competitive modern world through online service.

CONTENTS

Chapter 1 Introduction

1.1 Sri Lanka Ports Authority	
1.1.1 Background	1
1.1.2 Services	1
1.2 Project	
1.2.1 Background	2
1.2.2 Problem Identification	3
1.2.3 Objectives	4

Chapter 2 Literature Review

2.1 Software Development Methods	5
2.2 Software Development Life Cycle	7
2.2.1 Requirement Analysis	7
2.2.2 System Designing	8
2.3 Database Designing	9
2.3.1 E-R Modeling	12
2.3.2 Normalization	13
2.4 Interface Designing	14
2.5 Database Online Approach	16

Chapter 3 Analysis and Designing

3.1 How Analysis was Carried out	23
3.2 Functional Requirements	23
3.3 Non Functional Requirements	24
3.4 Designing Approach	26
3.5 Data Flow Diagram	27
3.6 Entity Relationship Diagram	27

Chapter 4 Implementation	
4.1 Implementation Environment	28
4.2 User Interface	28
4.3 Codes	28
4.4 Implementation of the process of online	30
Chapter 5 Conclusion	
5.1 Result	32
5.2 Discussion	33
5.3 Further Enhancement	34
References	35
Appendix I (DFD Diagrams)	36
Appendix II (ER-Diagram)	40
Appendix III (Process Flow Diagram)	41
Appendix IV (Tables structures)	42
Appendix V (User Interfaces)	46
Appendix VI (User Manual)	58

LIST OF FIGURES

Chapter 1
 Fig 1.1 3

Chapter 2
 Fig 2.1 5
 Fig 2.2 6
 Fig 2.3 7
 Fig 2.4 10
 Fig 2.5 17

Chapter 3
 Fig 3.1 26

Chapter 5
 Fig 5.1 33

Appendix V
 Fig V-I 57

LIST OF TABLES

Chapter 2

Table 2.1	20
Table 2.2	21

Appendix IV

Table IV-I	42
Table IV-II	42
Table IV-III.....	42
Table IV-IV.....	43
Table IV-V.....	43
Table IV-VI	43
Table IV-VII	44
Table IV-VIII	44
Table IV-IX	44
Table IV-X	45

ABRIVIATION

API	Application Program Interface.
ASP	Active Server Page
ADO	ActiveX Data Object
DAO	Data Access Object
DBMS	Database Management Systems
DFD	Data Flow Diagram
DSN	Data Source Name
ER	Entity Relation
GIS	Geographic Information Systems
GUI	Graphical User Interface
HTML	Hypo Text Markup Language
IIS	Internet Information Services
infor ⁿ	Information
ODBC	Open DataBase Connection
SLPA	Sri Lanka Ports Authority
1NF	1 st Normal Form
2NF	2 nd Normal Form
3NF	3 rd Normal Form
OLAP	On-Line Analytical Processing
PWS	Personal Web Server
RDO	Remote Data Object
SQL	Structured Query Language
VB	Visual Basic
URL	Uniform Resource Locater

INTRODUCTION

1.1 Sri Lanka Ports Authority

1.1.1 Background

The Sri Lanka Ports Authority (SLPA) was set up by an act of parliament in 1979 and administers and operates all specified commercial ports in Sri Lanka. The specified ports now include Colombo, Gall, Trincomalee and point pedru. The port of Colombo being the premier port in the country has naturally become the focal point of the all port development activities in Sri Lanka, since the formation of the SLPA.

The mission was to provide a conducive policy environment to consolidate and further develop port activities to internationally acceptable standards quality for Sri Lanka to be a competitive shipping center in south Asian region.

Location Latitude 5 55'N to 9 50'N
 Longitude 79 42'E to 81 52'E

1.1.2 Services

- Bonded warehousing and entrepot trade
- Duty free shopping
- Fresh water supply
- Bunkering
- Ship repairs
- Supply of ship stores
- Marine surveyors
- Fumigation services
- Pollution and oil spill control
- Health services
- LCL De-stuffing services
- Training institute
- Fire fighting services

1.2 Project

1.2.1 Background

A warehouse is a storehouse of the port that stores de-stuffed goods which are imported from other countries, until they get delivered from the relevant company or person. When goods are loaded to the ship to export, a list called manifest which contain details about goods in a vessel (ship) is sent to the relevant ports, which are occupied to unload those goods.

There are two types of containers according to the goods in them

- LCL – Goods in a container belongs to several owners
- FCL – Goods in a container belongs to only one owner

In this case only LCL containers are considered since others are not stored in warehouses.

After the particular vessel reached to the considered port, remove the seal of relevant containers and unloaded only listed goods, remarking details of them on a sheet called tally sheet. Also put a note as a comment whether the container unloaded completely or if not, to where remaining goods were directed.

In fact the manifest is containing from all the details of goods such as physical details of goods, consignee's details, agent's details as well as shipping line's details except warehouse location details. But a tally contains only some physical details of goods and some details of relevant container and the vessel.

Having stored goods in warehouses the relevant agent is noticed for delivering. Then the consignee or an authorized agent handles the process of delivery. Warehouse is involved this delivery process also up to preparing the gate pass. After verifying the authorization of goods, the relevant charges must be paid. And certification must be done by the rent clerk of the warehouse, and after doing some certifications by authorized employees, goods could be delivered. That is the process of delivery in brief.

THE RELEVANT OPERATION IN THE WAREHOUSE

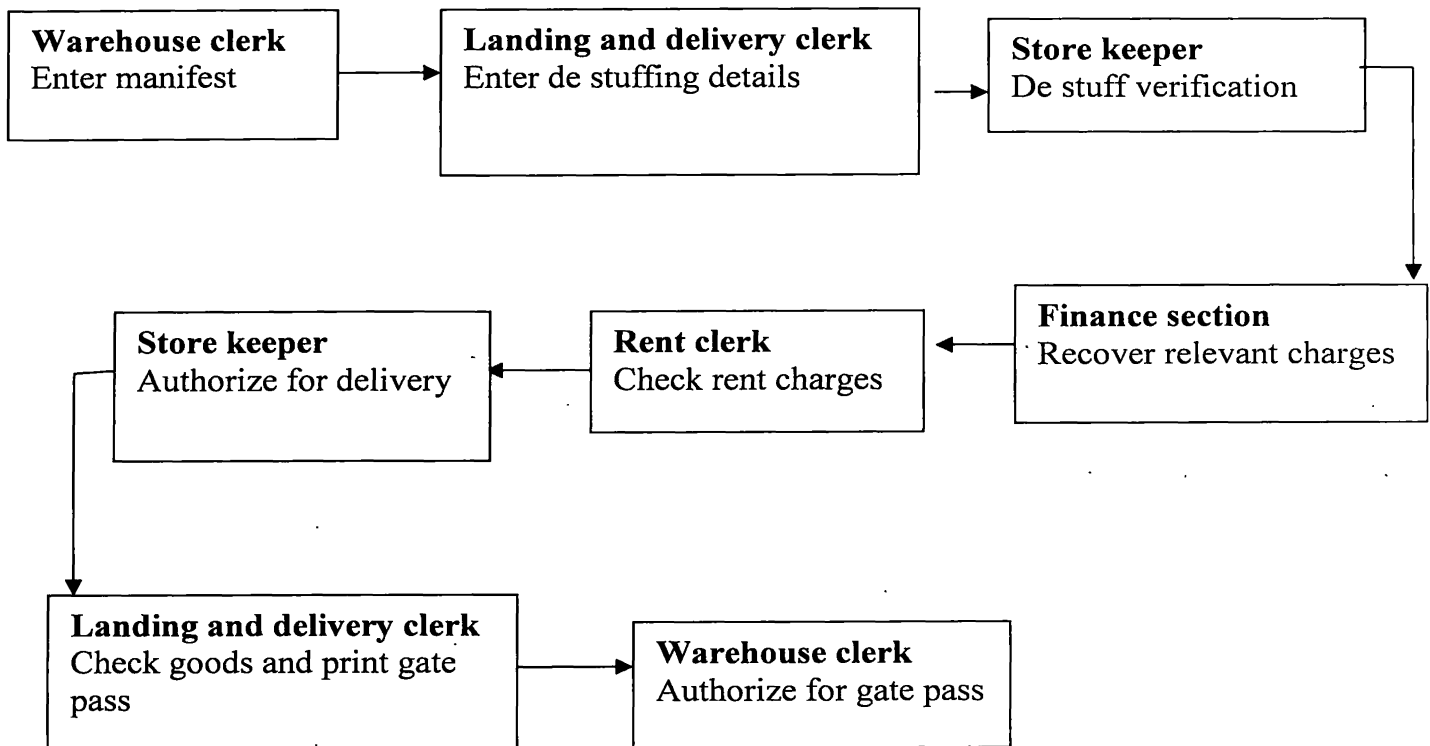


Fig.1.1 relevant operation in the warehouse

Presently all details are kept in files and various forms that are filled manually.

1.2.2 Problems Identification

Currently the owner is not provided any immediate method to know whether their goods have arrived and ready to be delivered, till noticed from the agent. This notification method would take much time, since this notification process goes through several agents before receiving to the consignee.

Except that main problem in consignee's view, this considered warehouse does not have computerized system to store landing and delivery data. The problems bellow could be identified due to that.

- Particular unloaded container details according to the manifest and the tally sheet would be difference. That will not be making in to consideration, since those details on separate documents as separate details.
- Particular unloaded goods details according to the manifest and the tally sheet would be difference. That doesn't notice at once. Although the difference will notice when they are ready to be delivery, several documents have to be considered to verify details.
- The time taken to query for particular goods is very much as well as the method is inefficient.

1.2.3 Objectives

The main objective of this project is to develop an online database for knowing whether goods are arrived to the relevant port, and they are ready for delivering.

Other than that a system is required to store de-stuffed and delivery data of this specific warehouse.

LITERATURE REVIEW

2.1 Software Development Methods

There are several methods, available to develop the system. IEEE proposed the number of standards for this purpose. Before selecting a method developer really concern set of rules that can have the project follows. Those are key features called time availability for project development, Entering releases, clear definition of the problem etc. According to the development method the way to reach end product will be change, model is select by developer according to the environment of customer requirement. But the final product should be well standards and fulfilled the customer requirements. The following models frequently use for software development process.

- (a) Linear sequential model
- (b) Evolutionary model
- (c) Spiral model

Linear Sequential Model

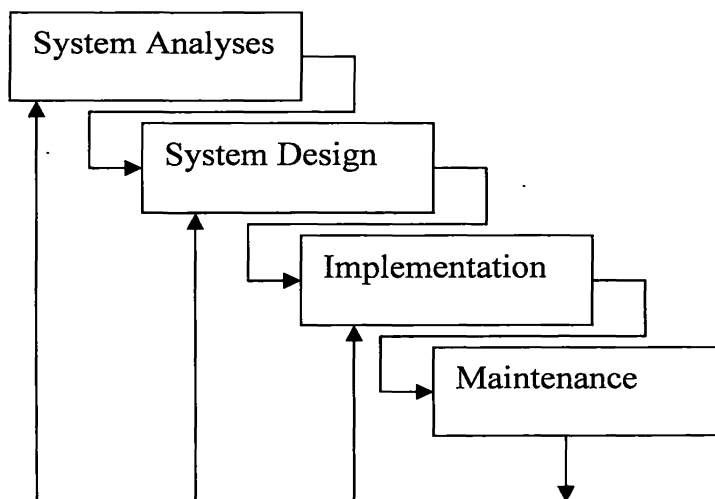


Fig. 2.1 linear sequential model

This model will very clear from the managerial view and easy to model, plan, monitor and understand. However for these models nearly all requirements need to be understood

very clearly. In this process this model is more appropriate when requirements are stable and well understood. Leaner sequential model is more suitable for relatively long term project.

Evolutionary Model

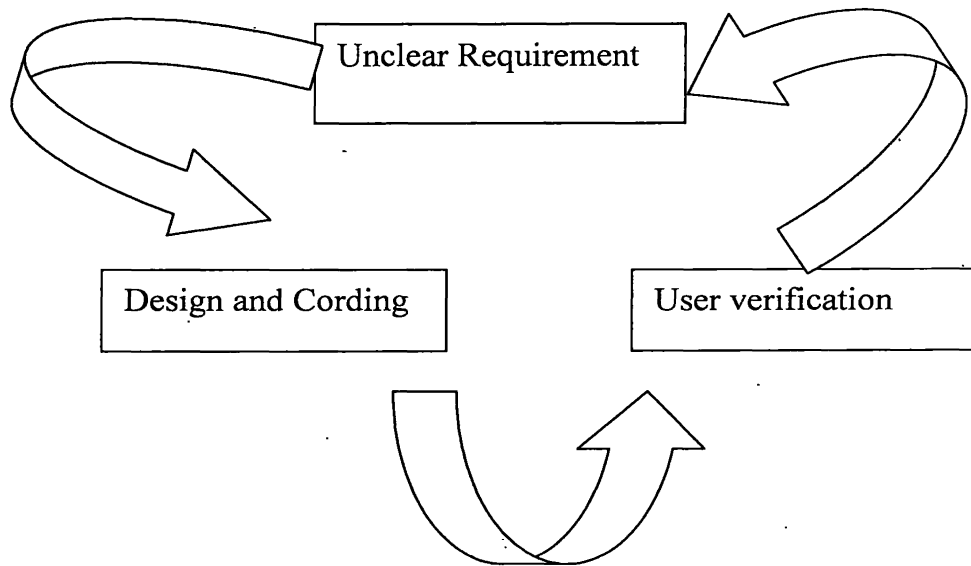


Fig.2.2 evolutionary model

Evolutionary model is more applicable when the requirements are not well defined or understood. Or lightly to undergo significantly changes. In this process system capabilities are evaluated by users. Process cycle is repeated until every stakeholder is satisfied

Spiral Model

This model is most appropriate for system strike requirements in terms of reliability and risk management. This process focuses attention on early error elimination puts quality objectives up run integrate development and maintenance and provide framework for simultaneous. However this process time cost overhead and may not be possible contractual rezones.

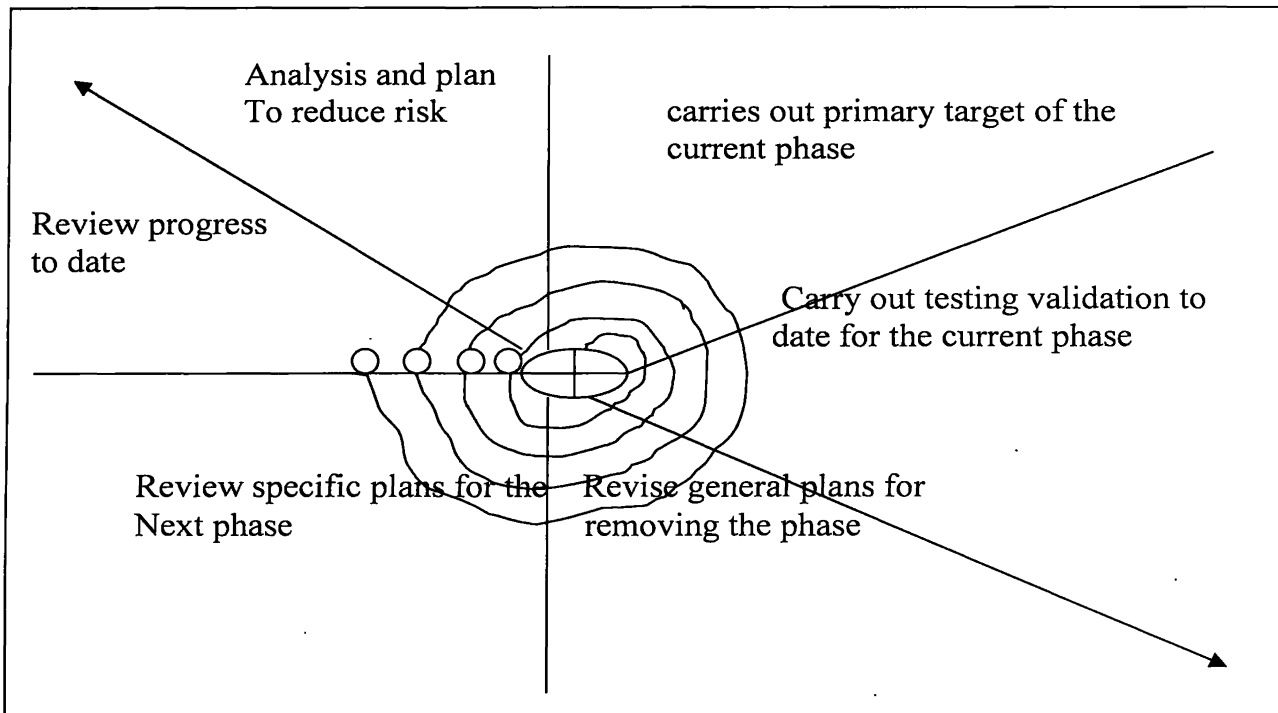


Fig.2.3 spiral model

2.2 Software Development Life Cycle

Systems project involve a sequence of phases called a system life cycle,

There are five stages of this process

- Requirement Analysis
- Software Designing
- Coding
- Testing
- Maintenance

2.2.1 Requirement Analysis

In this stage the nature of the computer program to be developed has to be identified. The System analyst must understand the functional requirements of the software system as well as inputs, outputs and the type of processing that is required.

Eventually, the analyst summarizes the information gleaned, including personal requirements, basic cost, and potential benefits of the new system.

There are several documents used in this stage.

- i. Requirement Definition Document
Define services which are expected from proposed system using natural language. this document must be prepared to easily understand even by non technical person
- ii. Requirement Specification Document
- iii. Software Requirement Specification Document

2.2.2 System designing

System designing is the process of deriving the solutions that satisfies the software. The solution means modeling or describing the problem with sufficient details so that it will be easy to implement. In designing stage there are several facts has to be considered.

- a) Data Structures
- b) Files
- c) Software Architecture
- d) Procedural Details

There are several designing methods as follows

- a) Top down designing
- b) Data flow oriented designing
- c) Jackson structured designing
- d) Object oriented designing

During this stage, designer employ a variety of tools such as data flow diagrams, entity-relationship diagrams, data dictionaries (lists of terms and their definitions), and the Gantt chart (a graph depicting design events, personal assignments, and time schedules). (*Edwards, P. (1997)*)

DFD (Data Flow Diagram)

When information moves through a software system the data entering the system is modified by a series of transformation. A DFD is a graphical method illustrating the flow and the transformations that are applied as data moves from input to output.

Coding

The software, designed as documents are translated into programming languages.

Testing

The program which was coded must be tested for possible logical or syntax errors.

Maintenance

Once the software is delivered to the customer the customer may need some additional functions or modifications. Also it is possible that some errors are identified at the customer's copy of the software.

2.3 Database Designing

Databases and database systems have become an essential component of every day life in modern society. A database is a collection of related data which mean known facts that can be recorded and that have implicit meaning.

In the past years most of the information that is stored and accessed is either textual or numeric that is called traditional database applications. Advances in technology have being leading to exciting new applications of database systems. Multimedia databases can now store pictures, video clips, and sound messages. Geographic Information Systems (GIS) can store and analyze maps, weather data, and satellite images. Data warehouses and on-line Analytical processing (OLAP) systems are used in many companies to extract and analyze useful information from very large databases for decision making. Real time and active database technology is used in controlling industrial and manufacturing process. And database search techniques are being applied to the World Wide Web to improve the search for information that is needed by users browsing through the internet.

A Database Management Systems (DBMS) is a collection of programs that enables user to create and maintain a database. The DBMS is hence a general-purpose of a software system that facilitates the process of defining, constructing, and manipulating databases for various applications. Defining a database involves specifying the data types,

structures, and constrain for the data to be stored in the database. Constructing the database is the process of storing the data itself on some storage medium that is control by the DBMS. Manipulating a database includes such functions as querying the database to retrieve specific data, updating the database to reflect changes and generating reports from the data.

A Simplified Database System Environment

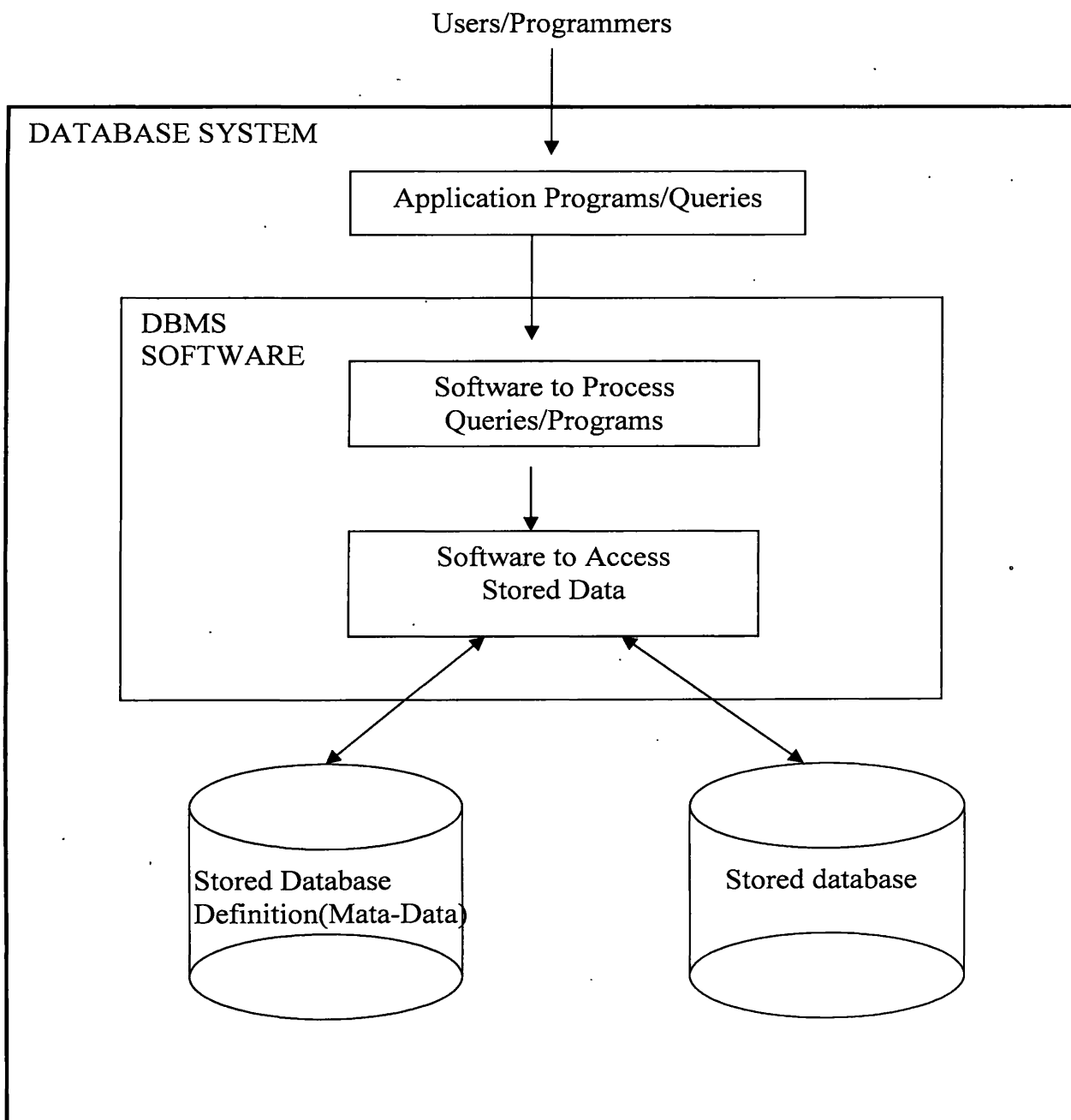


Fig.2.4 a simplified database system environment

Models of Database

- Hierarchical Model
- Network Model
- Relational Model

A database stores and manages both data and metadata. Physically the data within a database do not reside in separate files storing related record; instead data are stored in a single structure. Such an arrangement eliminates most of the file system's inherent limitations. The result of such flexibility however, is a far more complex physical structure. While the system files at least resemble its familiar manual counterpart, the data in a database are seemingly thrown together in an arrangement that is not immediately obvious. In fact database structures required by both the Hierarchical and Network database models often become complicated enough to diminish efficient database designing. The Relational database model changed all that. Using an analogy, the relational database uses an automatic transmission to relieve us of the need to manipulate clutch pedals and gearshifts. So while the relational database physically continues to store all data and metadata within a single structure; Structural and data independence allow designer to ignore the structure's physical complexity. In short, the relational model enables us to view data logically rather than physically. (*Elmasri, R., and Navathe, S. (2000)*)

Entities and Attributes

Relational design begins by defining the required entities. An entity is an object that exist and distinguishable from other objects. Each entity has certain characteristic known as attributes. A grouping of related entities becomes an entity set. Entity can not be assigned particular values, while attributes can be assigned. Entities are represented by rectangles, while attributes by ovals in ER-modeling.

Tables

The logical view of the relational database is facilitated by the creation of data relationships based on a construct known as a table.

Keys

Keys are attributes which uniquely define entities.

2.3.1 ER-Modeling

A model is the relatively simple representation, usually graphic, of complex real world data structures. The model's main function is to help us understand the complexities of the real world environment. Within the database environment a data model represents data structures and their characteristics, relations, constrains and transformations.

The American National standards Institute/ Standards Planning and Requirement committee defines three different data models based on their degree of abstraction; conceptual, external, internal. From the database designer's point-of-view, the relational DBMS tends to be oriented towards the conceptual model; the network and hierarchical DBMSes tend to require activities based on the physical model.

The ER-Model is a tool that commonly used to:

- Translate different views of data among managers, users, and programmers to fit into a common framework.
- Define data processing and constraint requirements to help us meet the different views.
- Help implement the database.

ER-Diagram

An ER-Diagram is used to map the ER-Model. This diagram depicts the ER-Model's three main components: entities, attributes, and relationships.

Relationship

A relationship is an association between entities. Each relationship is identified so that its name is descriptive of the relationship.

2.3.2 Normalization

To have a good database the structure of its contents must be designed carefully. A well designed database facilitates data management and becomes a valuable information generator. A poor design leads to redundant data. A poorly designed database generates errors that lead to bad decisions.

A database contains redundant data when some data are kept in different locations when the same data are kept in different locations for the same entity.

It is possible to create poor table structures even in a good database designing, hence the solution is based on normalization. Normalization is a process for assigning attributes to entities. Normalization reduces data redundancies and by extension, helps eliminate the data anomalies that result from those redundancies. Normalization does not redundancies, instead, it produces the control redundancy that lets us link database tables.

Normalization works through a series of stage call normal form. The first three stages are described as first normal form (1NF), Second normal (2NF), and third normal form (3NF). For most business database design purposes, third normal form is as high as we need to go in the normalization process. (*Elmasri, R., and Navathe, S. (2000)*)

1NF

The term 1NF describes the tabular format in which:

1. All the attributes are defined.
2. There are no repeating groups in the table. In other words, each row column intersection can contain one and only one value, rather than a set of values.
3. All attributes are dependent on the primary key.

Even though all relational databases satisfy the 1NF requirements, the table structure could be contain partial dependencies; that is, dependencies based on only a part of the primary key.

2NF

The relational database design can be improved easily by converting the database into a format known as the 2NF. as a result partial dependencies are eliminated. But it is still possible for a table in 2NF to exhibit transitive dependency that is one or more attributes may be functionally dependent on non key attributes.

3NF

The data anomalies created by the database organization are eliminated by breaking off the pieces. When the table is in 2NF and it contains no transitive dependencies, the table is in 3NF.

2.4 Interface Designing**Event driven model**

In event_ driven programming, it allows a user to create windows applications, without using much cords in any programming languages. Programming this way means when event occurs and have to write cord that will make program do something in response to the event.

Procedure defined as any block of cord that can be called from within the application. This procedure cord might be used to move objects around on the form, calculate a value from a formula, or write data to a database.

Object oriented concept

Object-oriented programming is the process of developing one to many lines of programming instructions based on well-defined designed models. Such design models are graphical illustrations that represent different aspects of objects, those objects' classes from which they derive functions, and property variables and their interactions with each other.

Object is an entity that has state and defines set of operation, which operates on that state. The state is represented as a set of objects attributes. The operation associate with the

object provides services to other object, which request the services when some computation is required.

Objects are created according to some object class definition an object class definition serves as a template for object. It includes all attribute and services which should be associated with an object and that class. Objects are abstraction of real world or system entity, which is responsible for managing their own private state and offering services to other objects. Objects are independent entities that may readily be change because state and representation information is held within the objects. Change to the representation may be made without reference to other system objects.

Data access methods in VB6

Data Access Object database Engine (DAO)

DAO is used to access to multiple type of data in one interface. DAO jet engine has set of routines and talks to a set of translation routines, which convert the request in to a format that the target database can understand. DAO has own query engine, search, and order, filter the data. This database engine is standalone in local area network.

In addition to using jet, DAO can also access ODBC databases through its ODBC direct object model

Remote Data Object Model (RDO)

All data work will be against remote database servers over the network connections. RDO is quite similar to DAO. One of the obvious differences is in the size of the object models. The DAO model has over twenty major objects to work with, and the RDO model has less than fifteen. RDO interface was meant for use when connecting to large multi-user database systems as SQL server, oracle. In fact, these two database interfaces sometimes work together. When DAO uses the ODBC direct model, it does it through RDO.

ActiveX Data Object Model (ADO)

Database project is needed to interact with user connected via web services and support the data programming.

Ado model is the smallest database object model. It provides a quick and effective way to build record set from connected database. Even better, it create a record set, disconnect from the server_ even save the record set to a local disk file_ and then, later, connect back to the database and send updated record set. (*Jain, C. (1998)*)

Visual Basic 6.0 application that shows how to use the DAO model in code to create connection and record set, and to run parameter queries against both Microsoft Access and SQL Server databases.

In fact, there only three key objects in the model:

1. Connection represents the actual database connection,
2. Command is used to execute queries against the data connection.
3. Record set represent the set of records colleted from the query issued via the Command object.

2.5 Database online Approach

Today's technology has been moved rapidly from static to dynamic web pages, where content may be in a constant state of flux. The web server uses a standard interface called the common gateway interface (CGI) to act as the middleware-the additional software layer between the user interface front-end and the DBMS back-end that facilitates access to heterogeneous databases. The CGI middleware executes external programs or scripts to obtain the dynamic information, and it returns the information to the server. (*Elmasri, Navathe, 2000*)

Database access on the web using CGI scripts

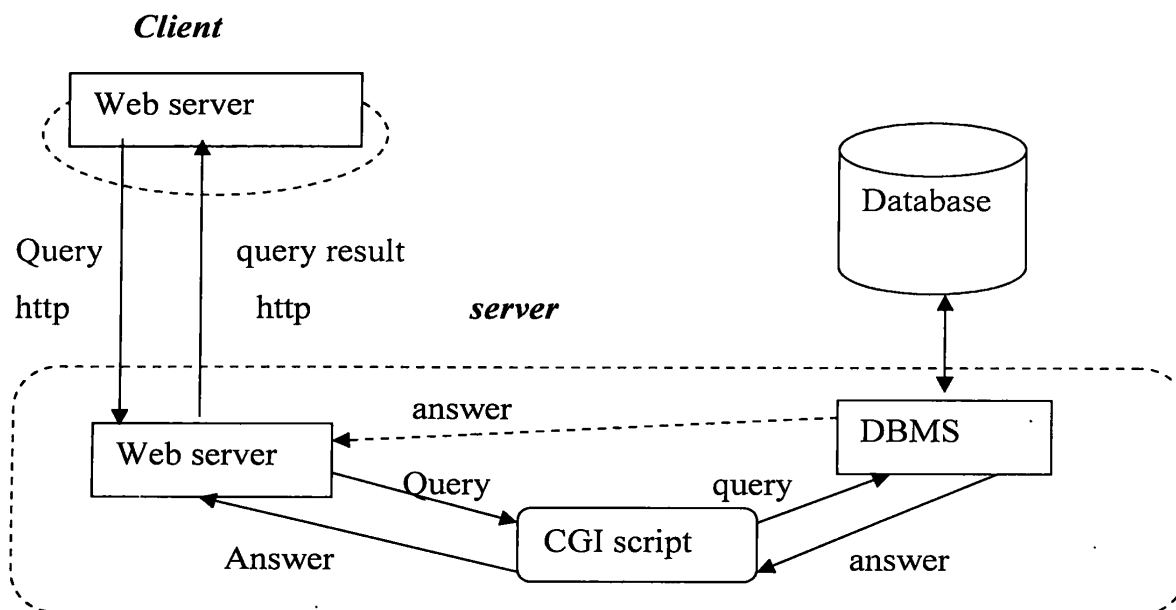


Fig. 2.5 database access on the web using CGI scripts

Asp (Active Server Page)

In the language of Microsoft, Active server page is an open, compile free application environment in which you can combine HTML, scripts, and reusable Active X server components to create dynamic and powerful web-based business solutions. ASP enables server side scripting for IIS with native support for both VBScript and Jscript.

Active Server Pages (ASPs) are web pages that contain server side scripts in addition to the usual mixture of text and HTML tags. Server side scripts are special commands you put in web pages that are processed before the pages are sent from the server to the web-browser of someone who's visiting your website. When you type URL in the address box or click a link on a web page, you're asking a web-server on a computer somewhere to send a file to the web-browser (also called a "client") on your computer. If that file is a normal HTML file, it looks the same when your web-browser receives it as it did before the server sent it. After receiving the file, your web-browser displays its contents as a combination of text, images, and sounds. In the case of an Active Server Page, the process is similar as HTML, except there's an extra processing that takes place just before the server sends the file. Before the server sends the active server page to the

browser, it runs all server side scripts contained in the page. Some of these scripts display the current date, time, and other information. Others process information the user has just typed into a form, such as a page in the website's guest-book. And you can write your own code to put in whatever dynamic information you want. In short ASP provides facility to display information that changes, while HTML does not.

Requirements to run ASP

Since the server must do additional processing on the ASP scripts, it must have the ability to do so, which support this facility, used to be Microsoft internet information services and Microsoft personal web server. Recently ASP support has been included in more popular web servers such as Apache.

- **Internet Information Services**

This is Microsoft's web server designed for the Windows NT platform. It can only run on Microsoft Windows NT 4.0, Windows 2000 professional, and Windows 2000 server. The current version is 5.0, and it ships as a part of the Windows 2000 operating system.

- **Personal Web Server**

This is a stripped-down version of IIS and supports most of the features of ASP. It can run on all Windows platforms, including Windows 95, Windows 98, and Windows Me. Typically, ASP developers use PWS to develop their sites on their own machines and later upload their files to a server running IIS. If your running windows 9X or Me, your only option is to use personal web server 4.0.

The Active server pages Model

A server-side script begins to run when a browser requests an .asp file from the Web server. Web server then calls ASP, which processes the requested file from top to bottom, executes any script commands, and sends a Web page to the browser.

Because the scripts run on the server rather than on the client, Web server does all the work involved in generating the HTML pages sent to browsers. Server-side scripts cannot

be readily copied because only the result of the script is returned to the browser. Users cannot view the script commands that created the page they are viewing. (*Jones, A. (1999)*)

Creating an ASP Page

An Active Server Pages (ASP) file is a text file with the extension .asp that contains any combination of the following:

- Text
- HTML tags
- Server-side scripts

A quick way to create an .asp file is to rename the HTML files by replacing the existing .htm or .html file name extension with an .asp extension. If the file does not contain any ASP functionality, then the server dispenses with the ASP script processing and efficiently sends the file to the client. As a Web developer, this affords you tremendous flexibility because you can assign your files .asp extensions, even if you do not plan on adding ASP functionality until later.

To publish an .asp file on the Web, save the new file in a virtual directory on your Web site (be sure that the directory has Script or Execute permission enabled). Next, request the file with your browser by typing in the file's URL. (Remember, ASP pages must be served, so you cannot request an .asp file by typing in its physical path.) After the file loads in your browser, you will notice that the server has returned an HTML page. This may seem strange at first, but remember that the server parses and executes all ASP server-side scripts prior to sending the file. The user will always receive standard HTML.

You can use any text editor to create .asp files. As you progress, you may find it more productive to use an editor with enhanced support for ASP, such as Microsoft® Visual InterDev™. (*1997-1999 Microsoft corporation*) <http://msdn.microsoft.com/vinterdev/>

Accessing a Data Source

ActiveX Data Objects (ADO) are an easy-to-use yet extensible technology for adding database access to the Web pages. You can use ADO to write compact and scalable scripts for connecting to OLE DB compliant data sources, such as databases, spreadsheets, sequential data files, or e-mail directories. OLE DB is a system-level programming interface that provides standard set of COM interfaces for exposing database management system functionality. With ADO's object model you can easily access these interfaces (using scripting languages, such as VBScript or JScript) to add database functionality to your Web applications. In addition, you can also use ADO to access Open Database Connectivity (ODBC) compliant databases.

DSN (Data Source Name)

Data source can be a database, spreadsheet, text file etc. if you assign DSN to a data source so that irrespective of the data source details and location, you can use that data source; add, modify or delete records, just by knowing its DSN.

Creating a Connection String

The first step in creating a Web data application is to provide a way for ADO to locate and identify your data source. This is accomplished by means of a connection string, a series of semicolon delimited arguments that define parameters such as the data source provider and the location of the data source. ADO uses the connection string to identify the OLE DB provider and to direct the provider to the data source. The provider is a component that represents the data source and exposes information to your application in the form of rowsets.

The following table lists OLE DB connection strings for several common data sources:

Data Source	OLE DB Connection String
Microsoft® Access	Provider=Microsoft.Jet.OLEDB.4.0;Data Source= <i>physical path to .mdb file</i>

Microsoft SQL Server	Provider=SQLOLEDB.1;Data Source= <i>path to database on server</i>
Oracle	Provider=MSDAORA.1;Data Source= <i>path to database on server</i>
Microsoft Indexing Service	Provider=MSIDXS.1;Data Source= <i>path to file</i>

Table 2.1

To provide for backward compatibility, the OLE DB Provider for ODBC supports ODBC connection string syntax. The following table lists commonly used ODBC connection strings:

Data Source Driver	ODBC Connection String
Microsoft Access	Driver={Microsoft Access Driver (*.mdb)};DBQ= <i>physical path to .mdb file</i>
SQL Server	DRIVER={SQL Server};SERVER= <i>path to server</i>
Oracle	DRIVER={Microsoft ODBC for Oracle};SERVER= <i>path to server</i>
Microsoft Excel	Driver={Microsoft Excel Driver (*.xls)};DBQ= <i>physical path to .xls file</i> ; DriverID=278

Microsoft Excel 97	Driver={Microsoft Excel Driver (*.xls)};DBQ= <i>physical path to .xls file</i> ;DriverID=790
Paradox	Driver={Microsoft Paradox Driver (*.db)};DBQ= <i>physical path to .db file</i> ;DriverID=26
Text	Driver={Microsoft Text Driver (*.txt;*.csv)};DefaultDir= <i>physical path to .txt file</i>
Microsoft Visual FoxPro® (with a database container)	Driver={Microsoft Visual FoxPro Driver};SourceType=DBC;SourceDb= <i>physical path to .dbc file</i>
Microsoft Visual FoxPro (without a database container)	Driver={Microsoft Visual FoxPro Driver};SourceType=DBF;SourceDb= <i>physical path to .dbf file</i>

Table 2.2

Accessing Data with ASP

ASP offers the Web developer flexible, easy-to-use, scalable methods to interact with OLE DB database providers for Internet or intranet applications. Using ASP and the Microsoft Data Access Components (MDAC) not only means that you can initially develop a solution using Microsoft Access and scale up to Microsoft SQL Server 6.5, but that you can access any other databases that have OLE DB providers. With the additional power of Component Services, you can create highly scalable multi-tier applications that support Web-based transactions. (1997-1999 Microsoft corporation)

<http://www.microsoft.com/data/>

Chapter 3

ANALYSIS AND DESIGNING

3.1 How Analysis was carried out

Before developing the software the nature of the computer program must be identified. What the program has to do as well as what are the inputs, outputs, and the type of process required.

Hence the analysis was carried out having followed the methods mentioned bellow.

- Discussed with relevant employees to get an idea, what are their functions in this process as well as how the process is going on.
- The process was observed having visited to the relevant workshops in the warehouse to get a clear idea.
- Documents were collected which were used by the warehouse that kept written data.

While designing the system some puzzles were created, for that relevant employees was visited again. As a result the view of the design, extended and in some cases it was slightly changed.

3.2 Functional Requirements

Functional requirements describe functionality or system services. They are statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situations.

1. The software should store information on de-stuff and delivery.
 - 1.1 The user should be provided with facility to enter information on manifest.
 - 1.2 The user should be provided with facility to enter information on de-stuff.
2. The system should provide facility to verify de-stuff details comparing with manifest information.

3. The system should provide facility to certify authorization for delivery by particular authorized employees.
4. the software should be able to print gate pass for particular delivery
5. The software must provide ways to query for de-stuff and delivery of concern about location and other details.
 - 5.1 The user must be provided with means of entering the details he is going to query about.
 - 5.2 There should also be an area for typing the query.
 - 5.3 When the user has presses the Submit query button the output should appear in a separate window with facilities to either save or print.
6. The software should be able to produce Reports
 - 6.1 The system should be able to produce a daily report on all de-stuff. It should include the de-stuff container as well as vessel.
 - 6.1.1 The inputs will be the selected date.
 - 6.1.2The output will be a text report of vessel details including vessel reference no, vessel name, arrival date and container details including vessel reference no, container no, seal no and hatch no.
 - 6.2 The system should be able to produce a daily report on delivery.
 - 6.2.1 Inputs for this function will be the selected date.
 - 6.2.2 The output will be a text report of delivery information including external details such as consignee, agent.

3.3 Non Functional requirements

Non-functional requirements are constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

Non-functional requirements are weighted according to the following criteria:

- High priority core requirements – requirements that are essential for the system

- Low priority core requirements – requirements that are not needed at the moment but will be needed in the future
- Optional requirements – requirements that are not essential to the system but that are good to have.

Usability - High priority core requirement

The effort required to learn, operate, prepare input and interpret output of the system. Usability is very important to the customer. The end product must provide a clear API for implementing the GUI. The Users should find the system easy to learn and use.

Efficiency - High priority core requirement

The amount of computing resources (time, space) required by the system to perform its function. Efficiency issues should be considered in architectural scale when designing the implementation of required functionality. When a user enters information on a form and submits it or presses the ok button he should get a response from the system before one minute

Reliability - Low priority core requirement

Reliability is defined as the extent to which the system can be expected to perform its intended function with required precision and without failure. Reliability issues should be considered when designing the system. Mean time to fail should be less than two per month.

Extendibility – Low priority core requirement

For all functionality mentioned in this document but not implemented by the project, there should be clear APIs and enough documentation to build functionality in the future. Especially it should be easy to add support for new functionality.

Maintainability – Optional requirement

The effort required when introducing modifications into the system. Both the database and the code produced must be maintainable. Therefore sufficient documentation on the

code and the database should be produced. Coding standards should be followed when programming takes place.

Security - Optional requirement

Give access to the system, only to authorized personnel.

3.4 Designing Approach

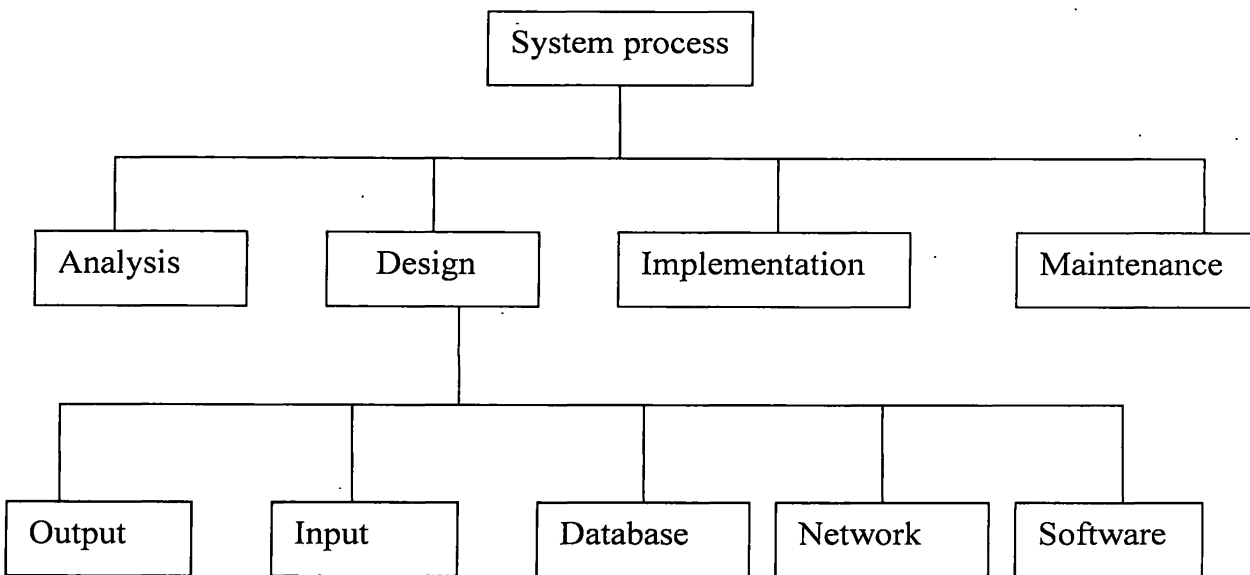


Fig. 3.1 designing approach

Having analyzing, the designing was carried out considering the requirements as well as the system background. The system designing was done using the structured method, by drawing Data Flow Diagrams (DFD). There were four Data flow diagrams including the context diagram, up to the third level.

Main tables relevant to the system were created, and they were decomposed into several tables, due to the normalization. ClearingAgent, consignee, deliveryDataEntry, deliveryPass, employee, goods, login, manifestContainer, manifestVessel, tallyheadingEntry, vehicle, and wharfClerk were referred as tables in the database. Then the ER diagram was designed for the database.

Interfaces were designed considering the functional requirements, as well as nonfunctional requirements.

3.5 Dataflow Diagram

The context diagram which shows all inputs and outputs of the system, and the 1st level DFD which shows all processes involved with the system, were drawn. Then two processes; report generation and prepare gate pass, were extended further according to the requirements as shown in appendix I.

3.6 ER-Diagram

Relevant ER-diagram was drawn as shown in appendix II

Chapter 4

IMPLEMENTATION

4.1 Implementation Environment

A Microsoft Access 2000 database was created to store data, for the convenience of data accessing through Visual Basic Codes, the database was converted to Access 97 file format using the database utilities of the tools option. Further the database was assigned to Data Source Name (DSN) for the purpose of online.

User interfaces were created in standard EXE format of Visual Basic 6.0.

Data Access Object (DAO) method was used to write Codes that used to access and manipulates data programmatically in the database.

IIS web-server was used as the web-server to run ASP codes, as the purpose of online the database.

4.2 User Interfaces

There were eighteen user interfaces. They were frmclearingAgent, frmconsignee, frmdailyDeliveryReport, frmdataEntryLogin, frmdeliveryDataEntry, frmdeliveryPass, frmdeliveryPassLogin, frmdestuffLogin, frmdestuffVessel, frmdisplayGoodsData, frmemployee, frmgoods, frmlogin, frmMain, frmManifestVessel, frmManifestContainer, frmtallyDetailsVhical, frmtalyHeading, frmvhical.

4.3 Codes

To add records to the database through a VB interface

- Relevant command button was click twice
- Code was written to open the database
- Code was written to add data to corresponding fields

Codes which were written to add data to the manifestVessel table

```
Set manifestVessel = cargodb.OpenRecordset("manifestVessel")
```

```

manifestVessel.AddNew
manifestVessel!VesselReferenceNo = txtVesselRefer.Text
manifestVessel!VesselName = txtVesselName.Text
manifestVessel!preplanArrivalDate = txtpreplanAD.Text
'manifestVessel!ActualArrivalDate = txtActualAD.Text
manifestVessel!VoyageNO = txtVoyageNo.Text

```

```

manifestVessel.Update
manifestVessel.Close

```

To delete data from the database

- One or many fields were selected to refer data from the database using a SQL statement
- Then the SQL statement was executed

Codes which were written to add data to the manifestVessel table

```

SQL = " delete * from manifestVessel where VesselReferenceNo = '" &
txtVesselRefer.Text & "'"
cargodb.Execute SQL

```

Codes for enabling a grid

```

If Not txtVesselName.Text = "" Then
cnt = cnt + 1
SQL = "select *from manifestVessel where VesselName=''" & txtVesselName.Text & "'"
Set manifestVessel = cargodb.OpenRecordset(SQL, dbOpenDynaset)
With MS1
.Rows = .Rows + 1
.TextMatrix(cnt, 0) = cnt
.TextMatrix(cnt, 1) = manifestVessel!VesselName
.TextMatrix(cnt, 2) = manifestVessel!preplanArrivalDate
.TextMatrix(cnt, 3) = manifestVessel!VesselReferenceNo

```

End With

4.3 Implementation of the process of online

To assign DSN to the database

- Control panel --->Administrative tool --->Data source (ODBC) --->System DSN tab --->Select Microsoft Access Driver (*.mdb) --->Finish ->type database name --->Select button --->browse to the location where the database was stored --->Ok

Relevant ASP page

- Code which is shown bellow were typed on a notepad
- the page was saved with .asp extension
- the location was C:/Interpub/wwwroot/ where .asp pages could be run

```

<% response.buffer=true%>
<html>
<head>
<title> stardeveloper/database</title>
</head>
<body>
<%
dim rs
set rs =server.createobject("ADODB.recordset")
rs.open"manifestVessel","DSN=cargostors"
while not rs.EOF
response.write "VesselReferenceNo:"& rs("VesselReferenceNo") & "<br>"
response.write "VesselName": "& rs("VesselName") & "<br>"
response.write "PreplanArrivalDate": "& rs("PreplanArrivalDate") & "<br>"
response.write "ActualArrivalData": "& rs("ActualArrivalData") & "<br>"
response.write "VoyageNo": "& rs("VoyageNo") & "<br>"
response.write"<br>"
rs.MoveNext

```

```
Wend  
rs.close  
set rs=nothing  
%>  
</body>  
</html>
```

Finally IIS was started, and the favorite browser was started, then the following was typed in the URL box of the browser.

http://127.0.0.1 (as the localhost)

Chapter 5

CONCLUSION

5.1 Result

As the result, well computerized system was developed for achieving the purposes bellow.

- Manifest data entry
- Tally data entry
- De-stuff conformation
- Delivery conformation
- Authorization of the gate pass
- Print the gate pass
- Database online access

One user interface is shown bellow. Others are shown in appendix V in details.

The screenshot displays a software window titled "Tally Heading" with the following components:

- Input Fields:**
 - Vessel Reference No: 2a
 - Vessel Name: samu
 - Arrival Date: 24/02/2004
 - Container No: 2aa
 - Seal No: (empty)
 - Hatch No: (empty)
 - Landed wearhouse: (empty)
 - Comments: (empty)
 - Remarked Employee ID: (empty)
- Table:**

No	Vessel Name	preplan Arrival Date	Vessel Rete No
3	samu	2/3/2222	2a
- Buttons:** ADD, EDIT, DELETE, CLOSE, OK, NEXT.
- Dialog Box:** "De-stuff Verification" with a "NEXT" button.
- Taskbar:** Shows the Start button and several open applications including APPENDIX, ABSTRA..., APPEND..., USER IN..., Data Ent..., and Tally He... The system clock shows 4:45 PM.

5.1 Discussion

Implementation of this system would be important in such a way.

- This system would enhance the efficiency
- Since the confirmation of particular goods is very reliable, less ability to have corruption.
- This would be a ladder to bring the particular warehouse to a well position among other competitive warehouses, since the computerization system as well as its online services.

Due to processing requirements, denormalization was done for some tables.

Although the creation of normalized relations is an important database design goal, it is only one of many such goals. Good database design also considers processing requirements, as tables are decomposed to conform to normalization requirements, the number of database tables expands, and joining the larger number of tables takes additional disk input/output operations and processing logic, thereby reducing system speed.

The rate of user requirements for assistance after system installation

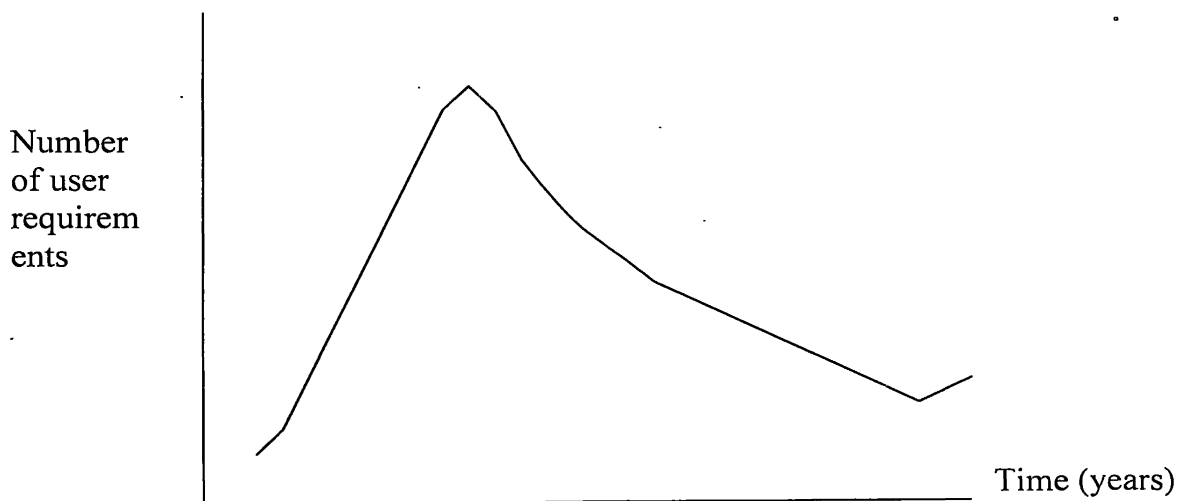


Fig. 5.1 The rate of user requirements for assistance after system installation

As an individual, I got lot of skills and knowledge in the process of software development, Due to this effort.

- The practical aspect of designing the software system
- Designing ER models and table structures
- Programming skill in visual basic 6.0
- Developing a client server system using ODBC connection
- Use of server side Scripting language like ASP

5.3 Further Enhancement

- The database security would be increased by designing it in SQL file format.
- The quality of the system would be increased using error handlers, with referring default error numbers, while coding interfaces.
- Entire system could be further developed, connecting external databases such as finance database, custom database through the network.
- The effort of online approach would be enhanced, by supporting query process.

REFERENCES

Edwards, P. (1997) Systems Analysis & Designing, McGraw-Hill, 1993, pp. 5-46, 390-413.

Elmasri, R., and Navathe, S. (2000) Fundamentals of DataBase Systems, Addison-Wesly, 2000, pp. 3-65, 465-496.

Jain, C. (1998) Visual Basic 6, Sams, 1998, pp. 260-399

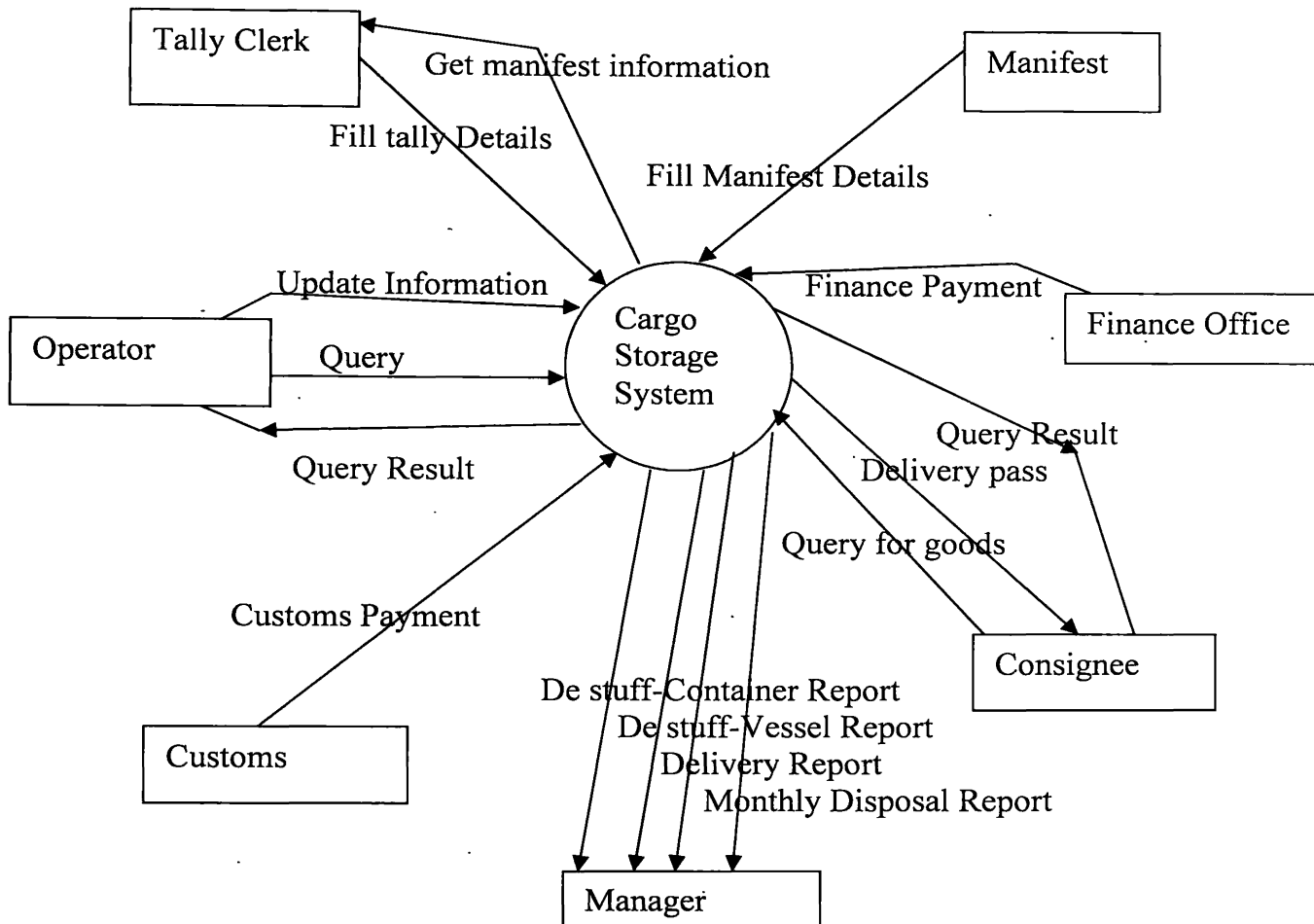
Jones, A. (1999) Visual Basic Developer's guide to ASP and IIS, BPB, 1999, pp. 9-52.

Microsoft Corporation, 1997-1999, <http://msdn.microsoft.com>

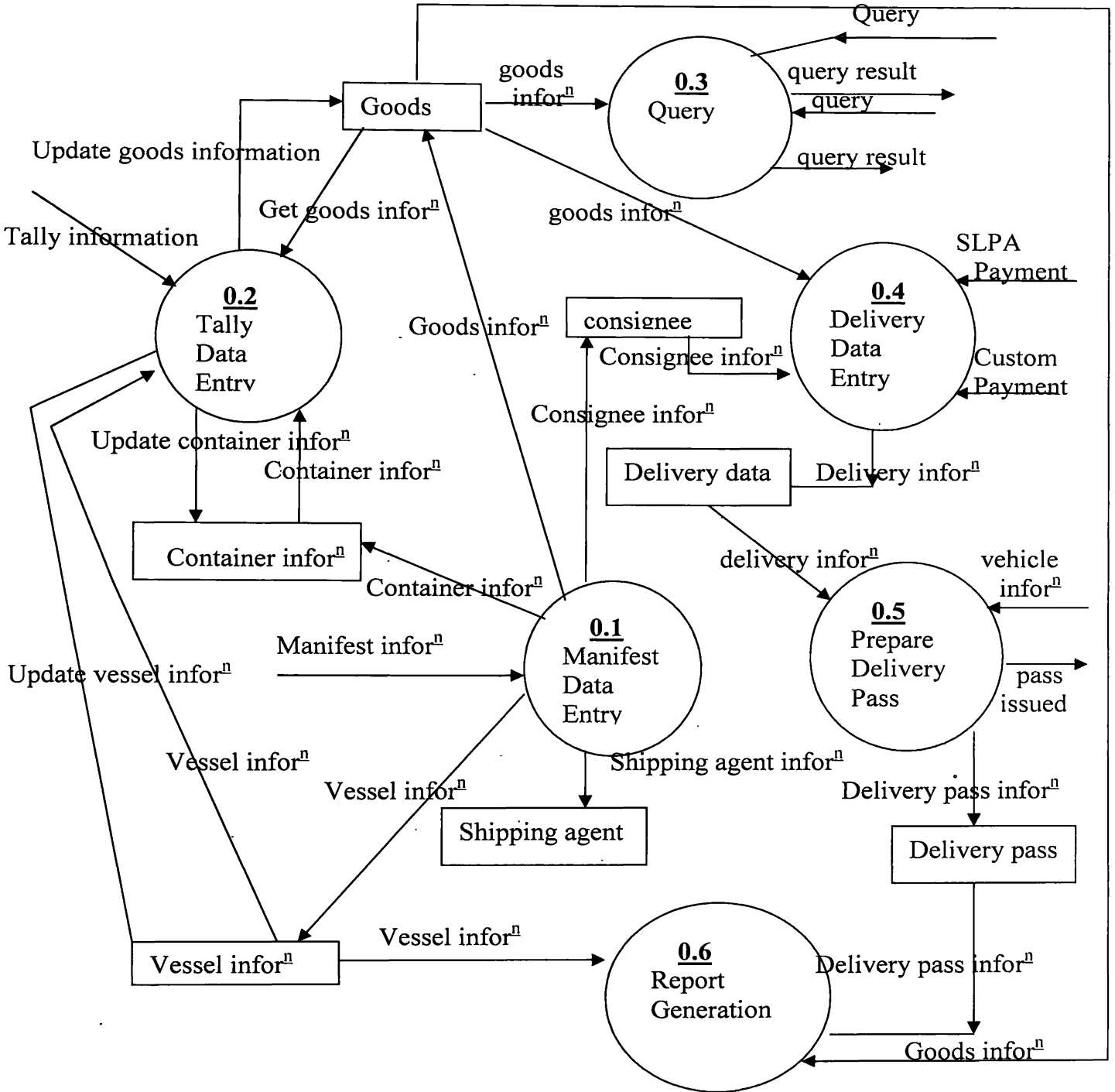
Microsoft Corporation, 1996, <http://www.w3schools.com> ,

APPENDIX I
Data Flow Diagrams

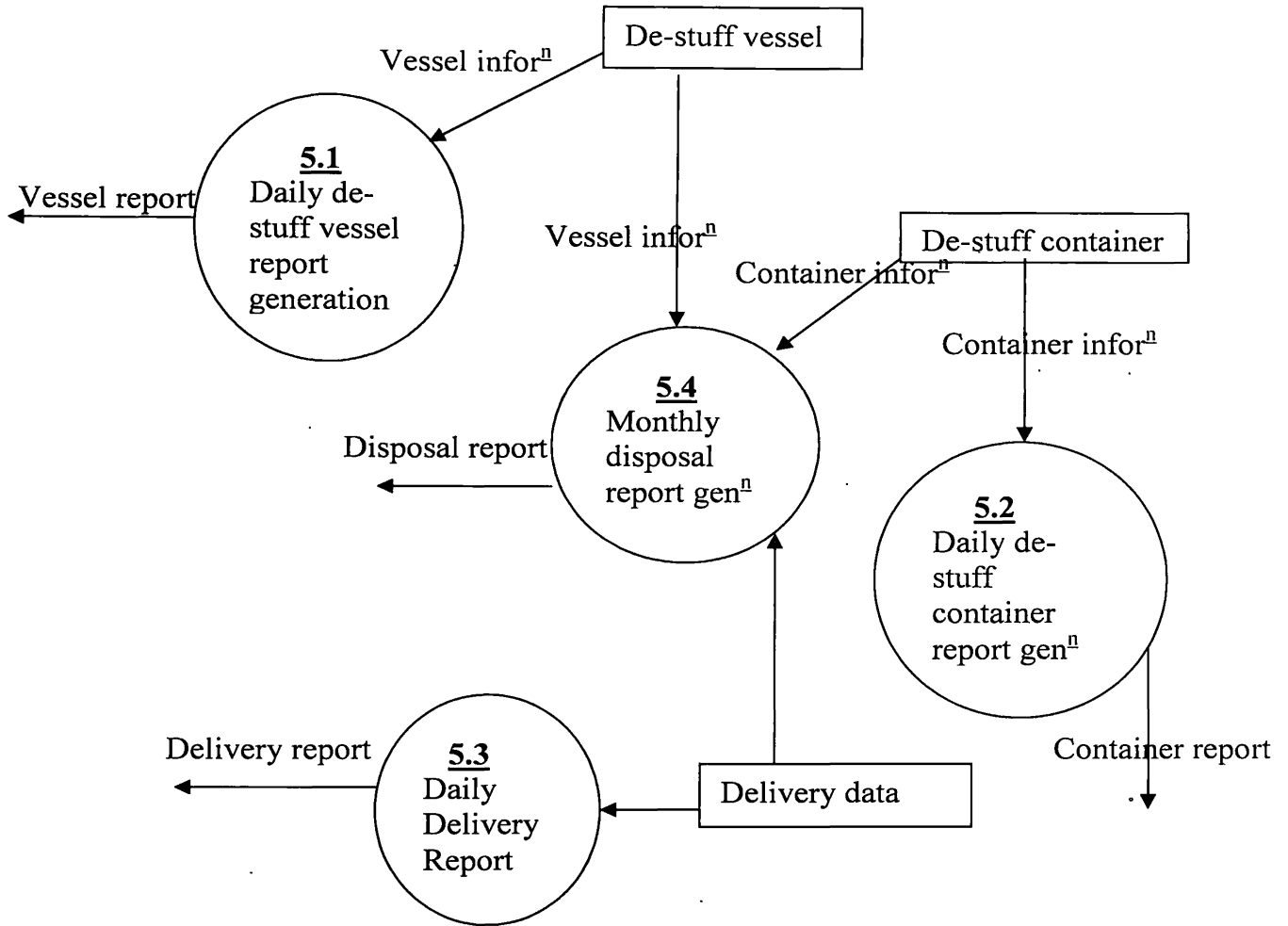
DFD: 0: Context Diagram



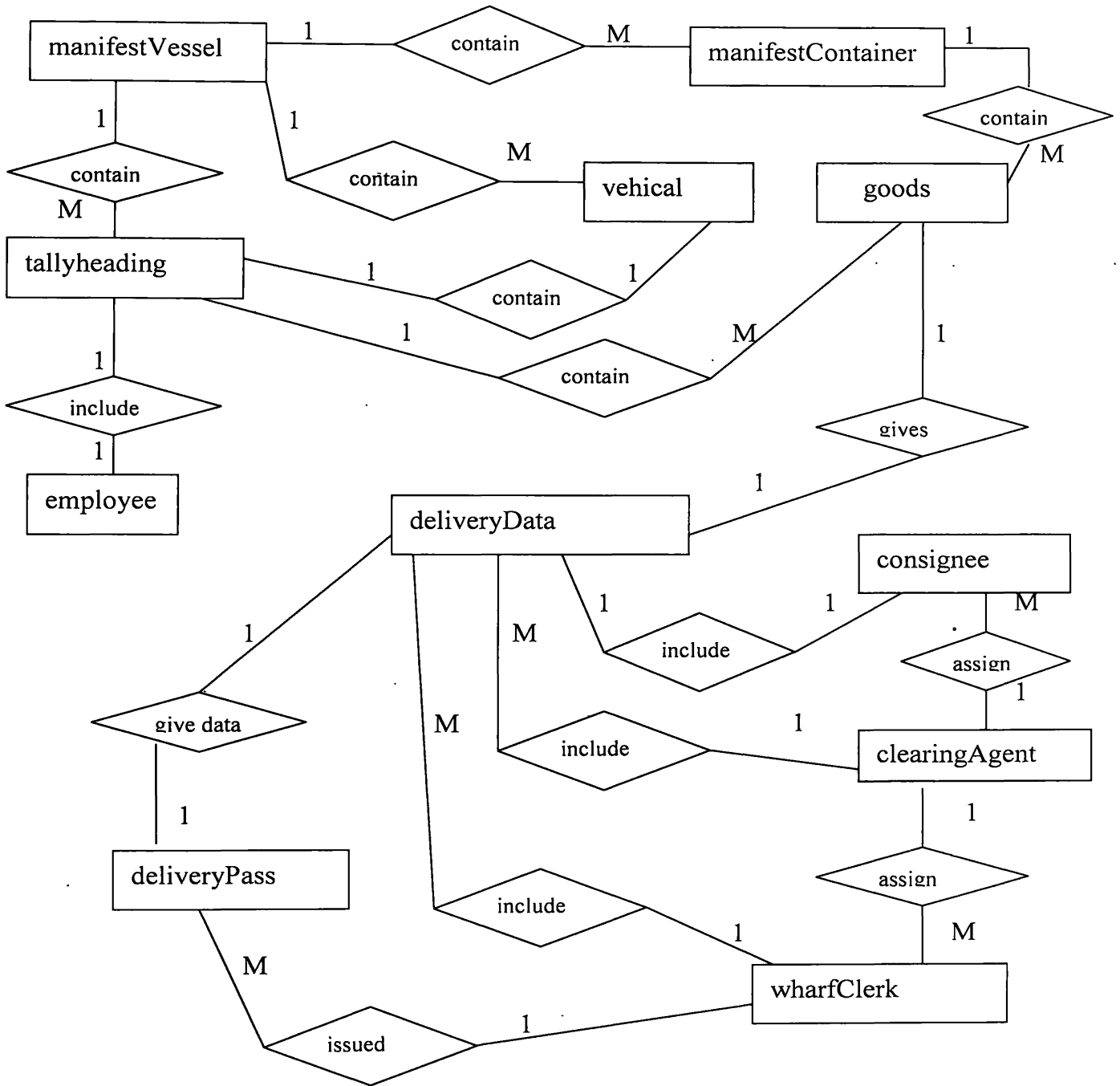
DFD: 1: Level 1



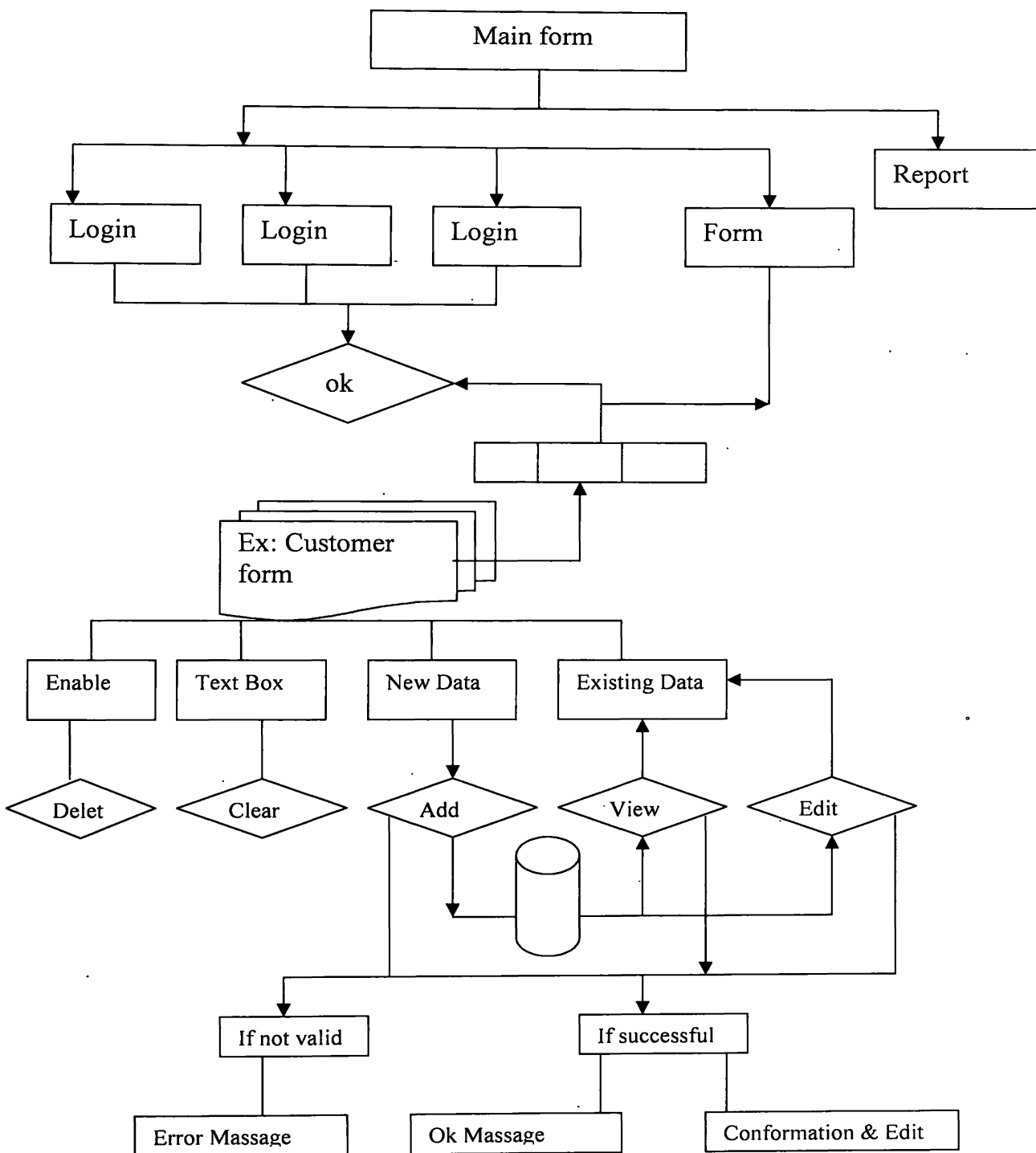
DFD: 3: Report Generation



APENDIX II
ER-Diagram



APPENDIX III Process Flow Diagram



APPENDIX IV
Tables Structures

ClearingAgent

Field Name	Data Type
*ClearingAgentID	Text
Name	Text
Address	Text
TelNO	Text

Table IV-I

Consignee

Field Name	Data Type
*ConsigneeNID	Text
WharfClerkID	Text
Name	Text
Address	Text
TelNo	Text

Table IV-II

DeliveryDataEntry

Field Name	Data Type
*VesselReferenceNo	Text
*ContainerNo	Text
*BillLadingNo	Text
ManifestMarkandNO	Text
SLPANO	Text
CustumNo	Number
SLPAPayment	Text
CustumPayment	Text
DeliveryDate	Date/Time
DeliveryQuantity	Text
ClearingAgent	Text
WharfClerkID	Text
ConsigneeNID	Text

Table IV-III

DeliveryPass

Field Name	Data Type
*GatePassNo	Text
BillLadingNo	Text
ConsigneeNID	Text
VhicalNo	Text
Warehouse	Text
Quantity	Number

Table IV-IV

Goods

Field Name	Data Type
*VesselReferencNo	Text
*ContainerNo	Text
*BillLadingNo	Text
MarkandNumber	Text
ConsigneeNID	Text
Agent	Text
PackageDiscription	Text
Weight	Number
Volume	Number
WearhouseLocation	Text
Received	Text

Table IV-V

manifestContainer

Field Name	Data Type
*VesselReferenceNo	Text
*ContainerNo	Text
SealNo	Number
Size	Number

Table IV-VI

manifestVessel

Field Name	Data Type
*VesselReferenceNo	Text
VesselName	Text
preplanArrivalDate	Date/Time
ActualArrivalDate	Date/Time
VoyageNo	Number

Table IV-VII

TallyheadingEntry

Field Name	Data Type
*VesselName	Text
*ArrivalDate	Data/Timm
*ContainerorTagNo	Text
HatchNo	Number
LandedWearhouse	Text
Comments	Text
RemarkedEmployeeID	Text

Table IV-VIII

Vehicle

Field Name	Data Type
*VesselReferenceNo	Text
*TagNO	Text
MarkandNO	Text
HatchNo	Text
Agent	Text
ConsigneeNID	Text
Discription	Text
Acblower	Text
Clock	Text Text
CarCassette	Text
Wipers	Text
SpareWheels	Text
Jack	Text
Aerial	Text
DieselCup	Text
PersonalPkgs	Text

Radio	Text
Lighter	Text
ToolKit	Text
Hubcaps	Text
SideMirros	Numbers
Badges	Numbers
Keys	Numbers
Fuse	Numbers
OtherAccessories	Text
Received	Text
WarehouseLocation	Text

Table IV-IX

Employee

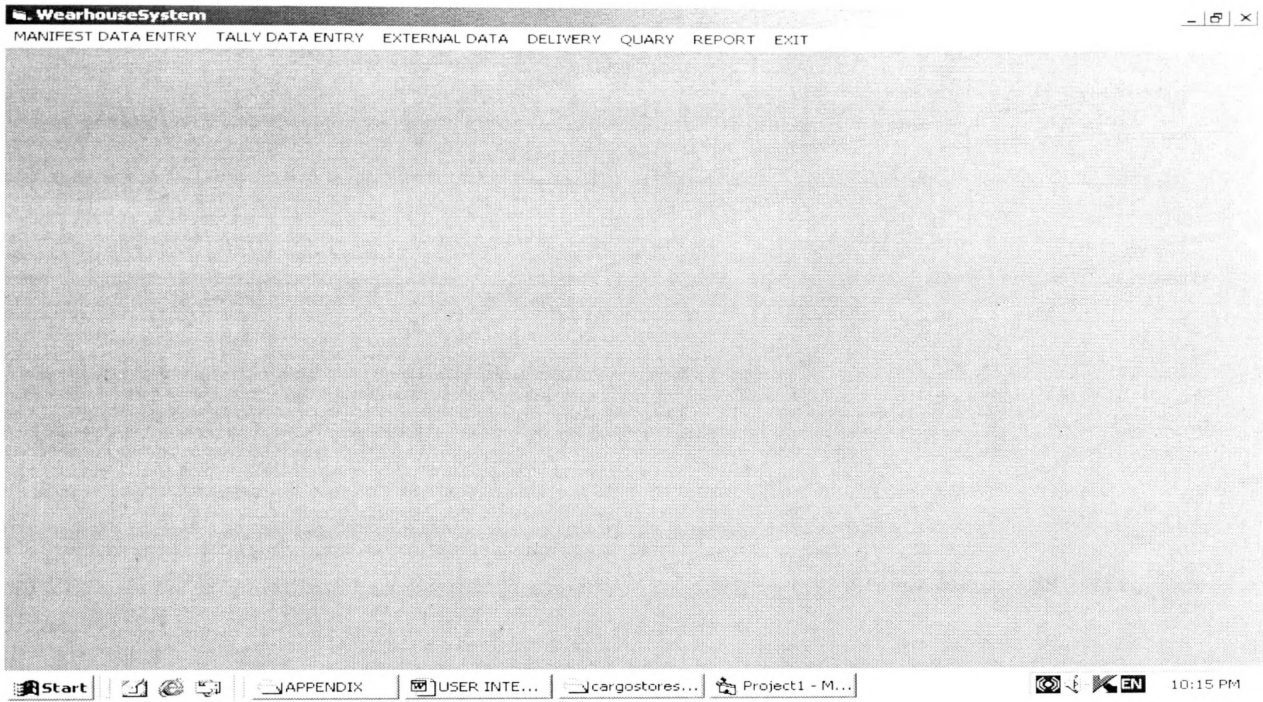
Field Name	Data Type
*EmployeeID	Text
Name	Text
Address	Text

Table IV-X

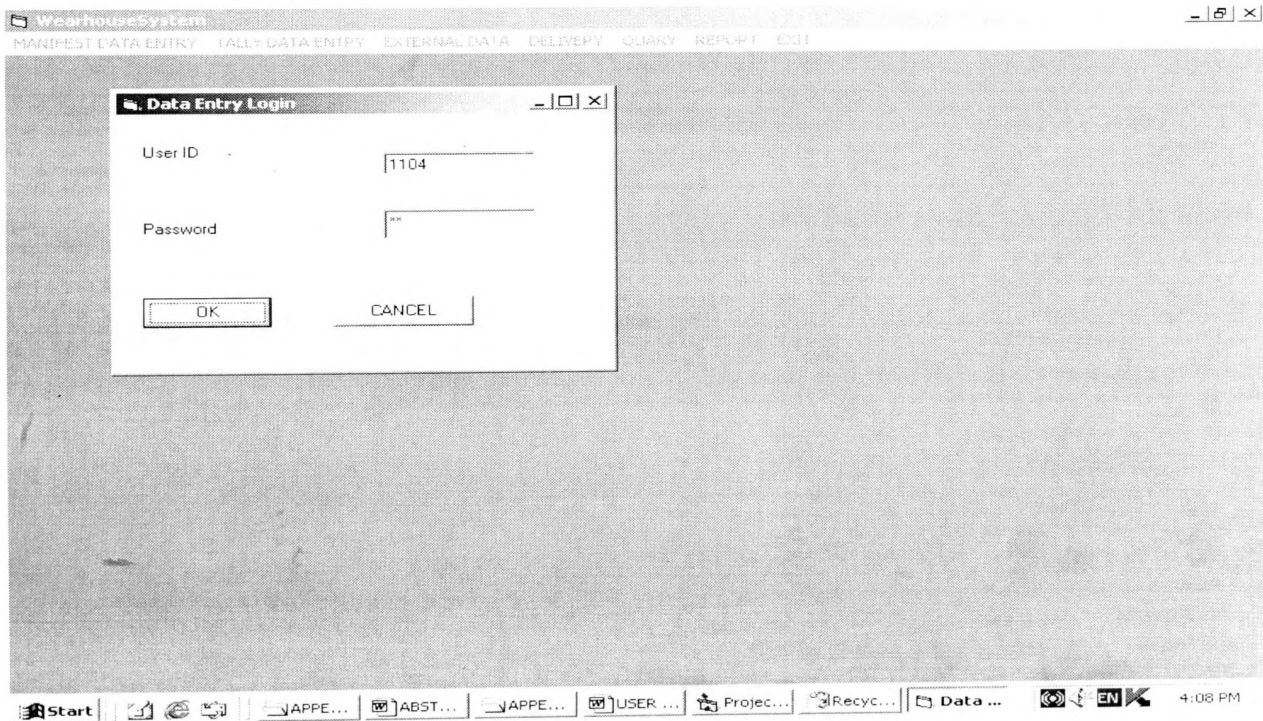
* - Primary Key

APPENDIX V User Interfaces

Main form



Manifest Data Entry Login



Vessel Data Entry

Manifest Data Entry for Vessel

Vessel Reference No: 2a
Vessel Name: samu
Preplan Arrival Date: 2/3/2222
Actual Arrival Date: 1/31/2004
Voyage No: 321

ADD EDIT DELETE CLOSE
VIEW NEXT

Manifest Data Entry for Vessel

Vessel Reference No: 245
Vessel Name:
Preplan Arrival Date:
Actual Arrival Date:
Voyage No:

Project 1
not existing data
OK

ADD EDIT DELETE CLOSE
VIEW NEXT

Container Data Entry

Manifest Data Entry for Container _ | ☒ | ×

Vessel Reference No	<input type="text" value="2a"/>
Container No	<input type="text" value="2aa"/>
Seal No	<input type="text" value="6534"/>
Size	<input type="text" value="423"/>

ADD	EDIT	DELETE	CLOSE
VIEW	NEXT	EXIT	

Goods Data Entry

Goods Data Entry _ | ☒ | ×

Vessel Reference No	<input type="text" value="245"/>
Container No	<input type="text"/>
Bill/Lading No	<input type="text"/>
Mark and no	<input type="text"/>
Consignee NID	<input type="text"/>
Agent	<input type="text"/>
Pkage Discription	<input type="text"/>
Weight	<input type="text"/>
Volume	<input type="text"/>
Received	<input type="text"/>
Wearhouse Location	<input type="text"/>

ADD	EDIT	DELETE	CLOSE
-----	------	--------	-------

Vehicle Data Entry

Tally Details for Vhical [- | [] | X]

Vessel Reference No []
Tag No []

Mark and No [] Received []
Hatch No [] Warehouse []
Agent [] Warehouse Location []
Consignee NID [] Discription []

AC/Blower [] Radio []
Clock [] Lighter []
Car Cassette [] Tool Kit []
Wipers [] Hubcaps []
Spare Wheels [] Side Mirrors []
Jack [] Badges []
Aerial [] Keys []
Diesel Cup [] Fuse []
Personal Pkgs [] Other Accessories []

ADD EDIT DELETE CLOSE
VIEW

Start [] APPENDIX [] ABSTRACT ... [] APPENDIX c... [] USER INTE... [] Tally Detai... [] 4:54 PM

Tally Data Entry Login

WarehouseSystem [- | [] | X]

MANIFEST DATA ENTRY TALLY DATA ENTRY EXTERNAL DATA DELIVERY QUARY REPORT EDIT

Data Entry Login [- | [] | X]

User ID [1103]
Password []

OK CANCEL

Start [] APPENDIX [] ABSTRACT ... [] APPENDIX c... [] USER INTE... [] Data Entry... [] 4:44 PM

Container Details

Display Goods Data

Vessel Reference No: 2a

Container No: 2aa

No	Vessel Refe No	Container No	Bill Lading No
1	2a	2aa	2aa1

OK

Goods Details

Goods Data Entry

Vessel Reference No: 2a

Container No: 2aa

Bill/Lading No: 2aa1

Mark and no: fgsfg

Consignee NID: joiq

Agent: joiiko

Package Discription: kjhk

Weight: 90

Volume: 898

Received: yes

Warehouse Location: rtrdy

de-stuff Verification

Store Keeper

ADD EDIT DELETE CLOSE

Select Option for Vehicle

Option

CONTAINER

VEHICLE

Start | APPENDIX | ABSTRA... | APPEND... | USER IN... | Option

4:59 PM

Vehicle Details

Tally Details for Vhical

Vessel Reference No: 2a

Tag No: 2aav

Mark and No: gdg Received: no

Hatch No: gdgh Warehouse:

Agent: gg Warehouse Location:

Consignee NID: gdfg Discription:

AC/Blower	YES	Radio	NO
Clock	yes	Lighter	YES
Car Cassette	YES	Tool Kit	NO
Wipers	YES	Hubcaps	wqe
Spare Wheels	YES	Side Mirrors	
Jack	YES	Badges	42
Aerial	YES	Keys	2
Diesel Cup	YES	Fuse	31
Personal Pkgs	YES	Other Accessories	64

ADD EDIT DELETE CLOSE

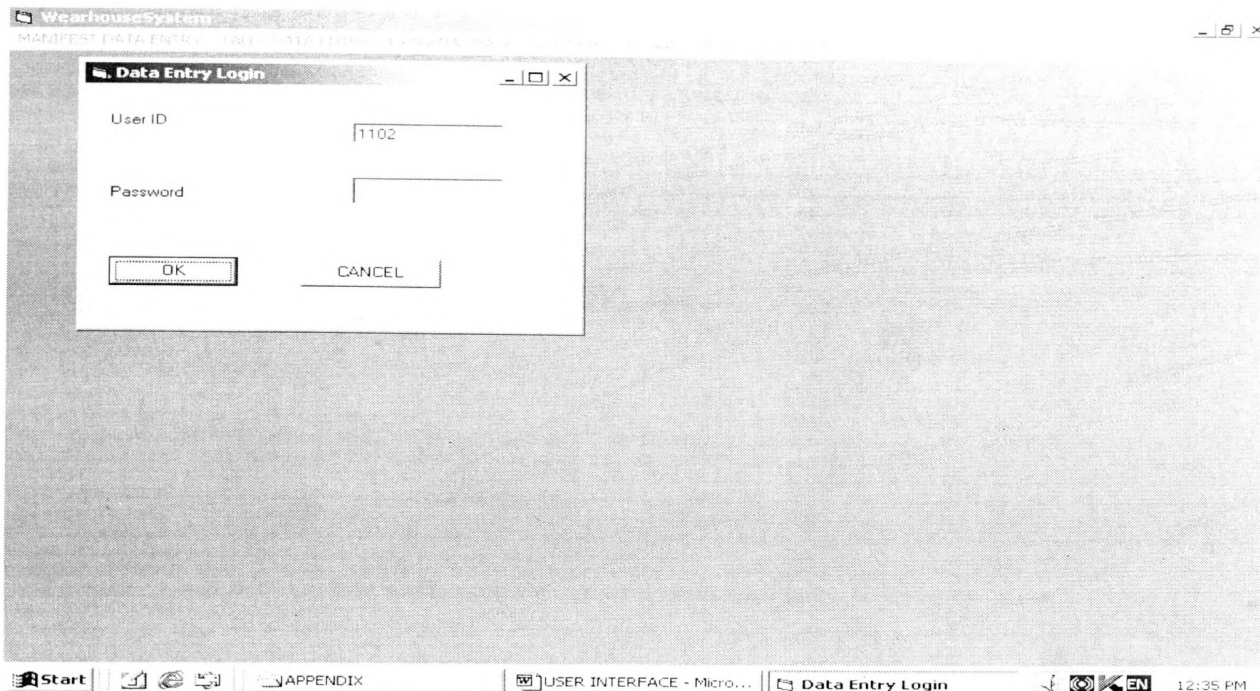
VIEW

de-stuff verification

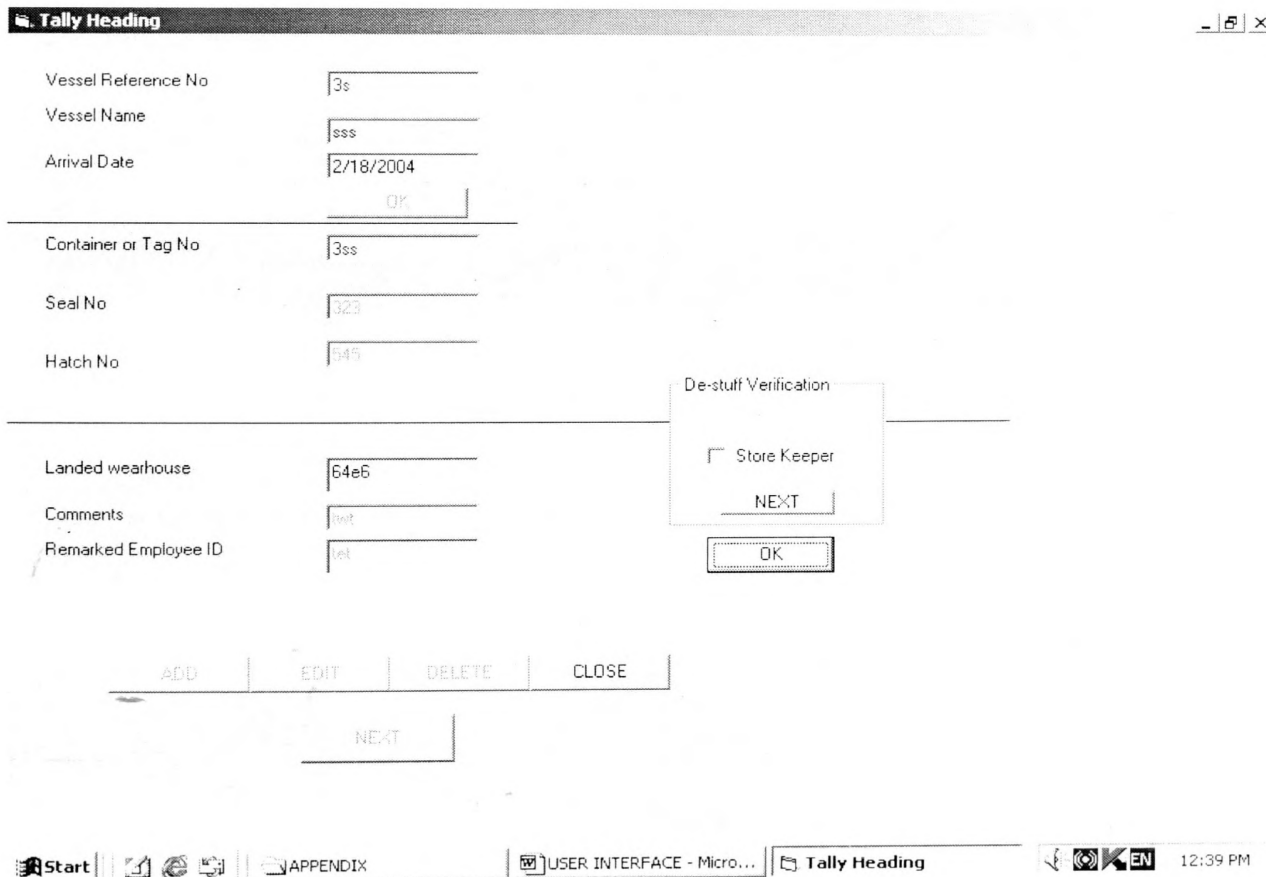
Store Keeper

Start | APPENDIX | ABSTRA... | APPEND... | USER IN... | Option | Tally De... | 4:58 PM

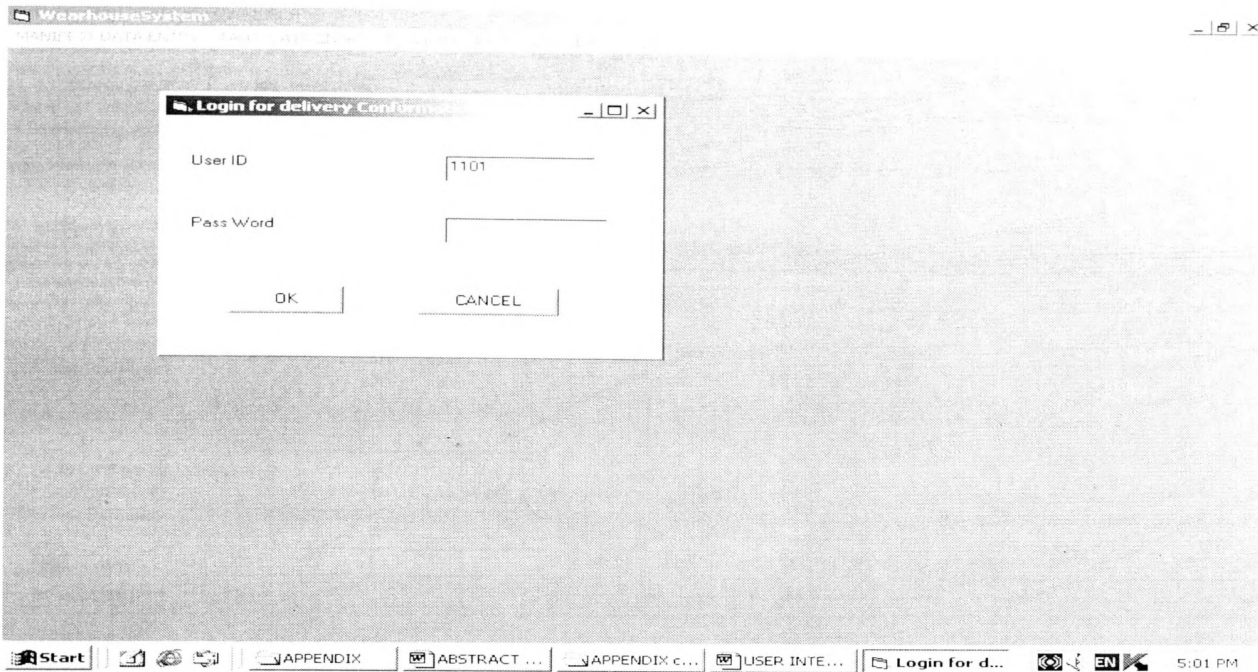
De-stuff Conformation Login



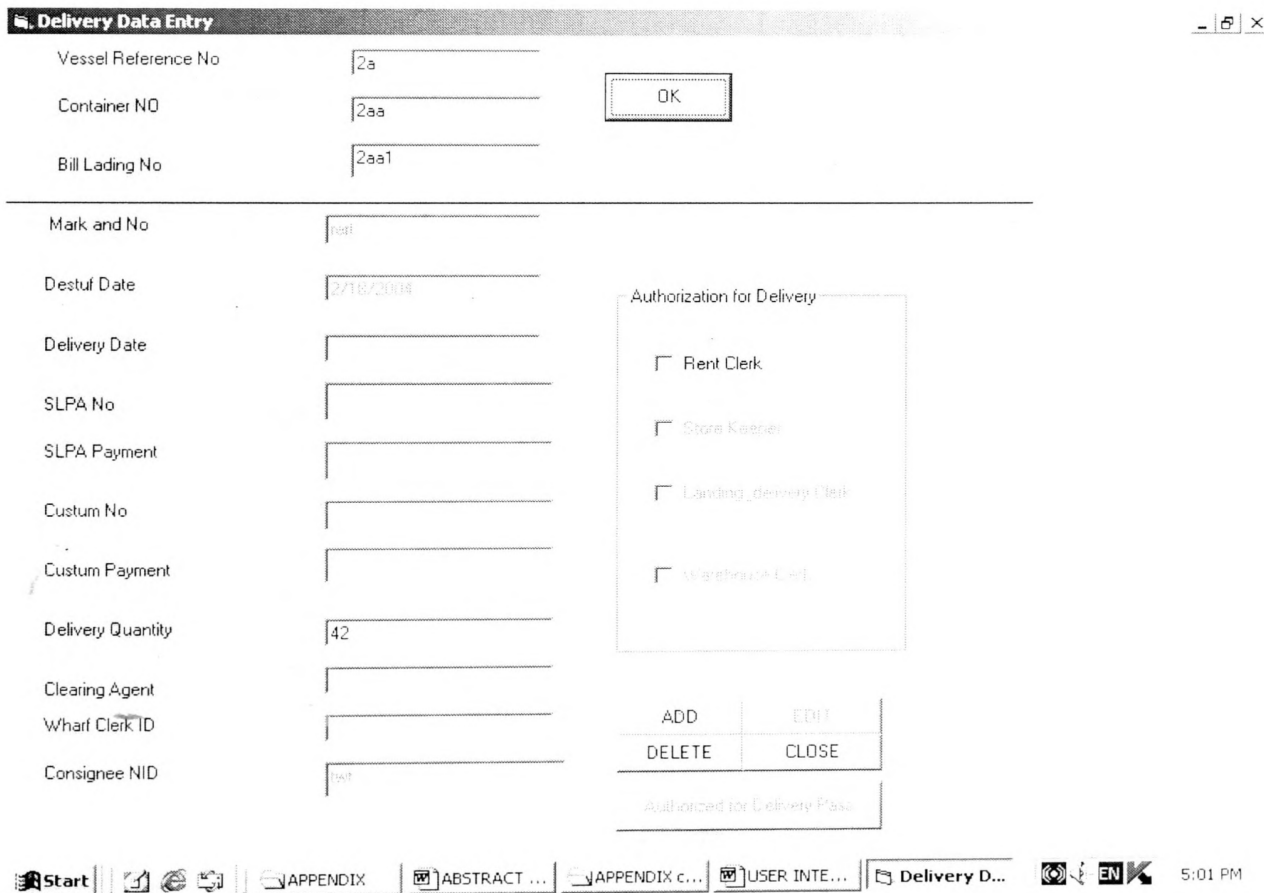
De-stuff conformation



Delivery Conformation Login



Rent Clerk's Operation



Store Keeper's Operation

Delivery Data Entry _ | [] | X

Vessel Reference No

Container NO

Bill Lading No

Mark and No

Destuf Date

Delivery Date

SLPA No

SLPA Payment

Custom No

Custom Payment

Delivery Quantity

Clearing Agent

Wharf Clerk ID

Consignee NID

Authorization for Delivery

Rent Clerk

Store Keeper

Landing_delivery Clerk

Warehouse Clerk

ADD	EDIT
DELETE	CLOSE

Authorized for Delivery Pass

Start | APPENDIX | ABSTRACT ... | APPENDIX c... | USER INTE... | Delivery D... | 5:37 PM

Landing & Delivery Clerk's Operation

Delivery Data Entry _ | [] | X

Vessel Reference No

Container NO

Bill Lading No

Mark and No

Destuf Date

Delivery Date

SLPA No

SLPA Payment

Custom No

Custom Payment

Delivery Quantity

Clearing Agent

Wharf Clerk ID

Consignee NID

Authorization for Delivery

Rent Clerk

Store Keeper

Landing_delivery Clerk

Warehouse Clerk

ADD	EDIT
DELETE	CLOSE

Authorized for Delivery Pass

Start | APPENDIX | ABSTRACT ... | APPENDIX c... | USER INTE... | Delivery D... | 5:36 PM

Warehouse Clerk's Operation

Delivery Data Entry _ | 5 | X

Vessel Reference No
 Container NO
 Bill Lading No

Mark and No
 Destul Date
 Delivery Date
 SLPA No
 SLPA Payment
 Custum No
 Custum Payment
 Delivery Quantity
 Clearing Agent
 Wharf Clerk ID
 Consignee NID

Authorization for Delivery

Prev. Del.

Store Passes

Landing Delivery Clerk

Warehouse Clerk

ADD EDIT
 DELETE CLOSE

Authorized for Delivery Pass

Taskbar: Start | APPENDIX | ABSTRACT ... | APPENDIX c... | USER INTE... | Delivery D... | 5:40 PM

Prepare Gate Pass

Delivery Pass Data Entry _ | 5 | X

Gate Pass NO
 Bill Lading No
 Consignee NID
 Warehouse
 Vhical No
 Quantity

Confirmation of Printing

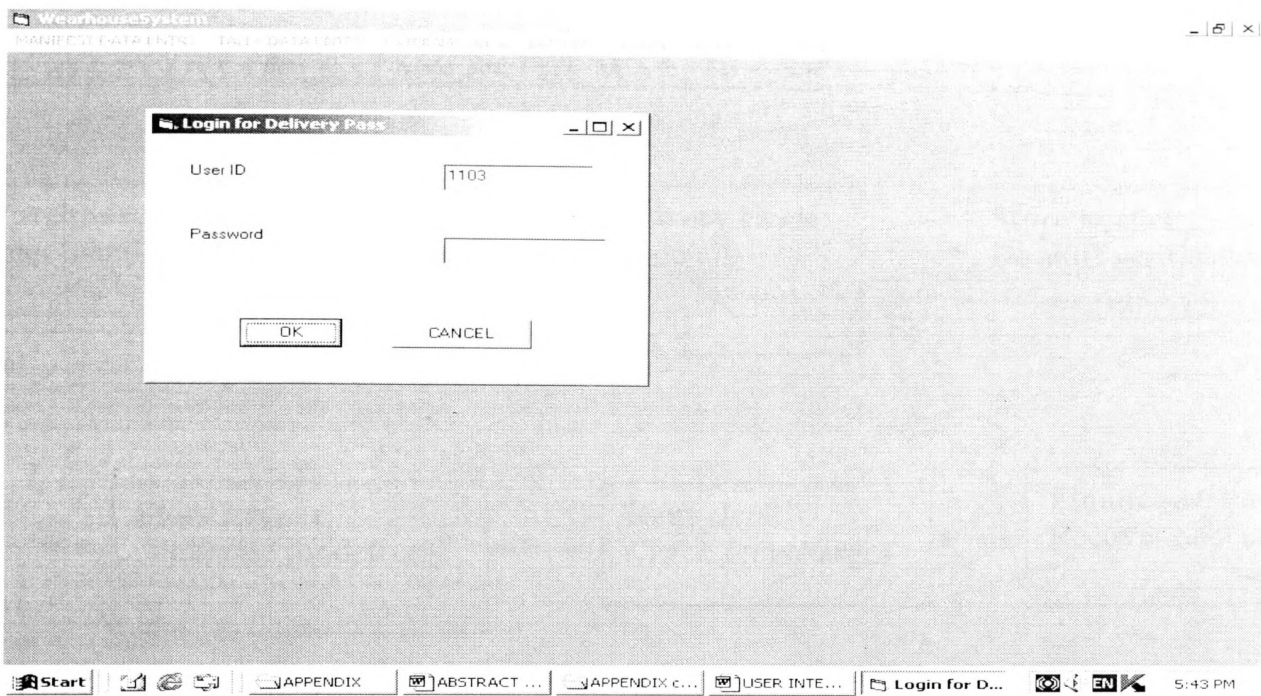
Warehouse Clerk

OK
 PRINT

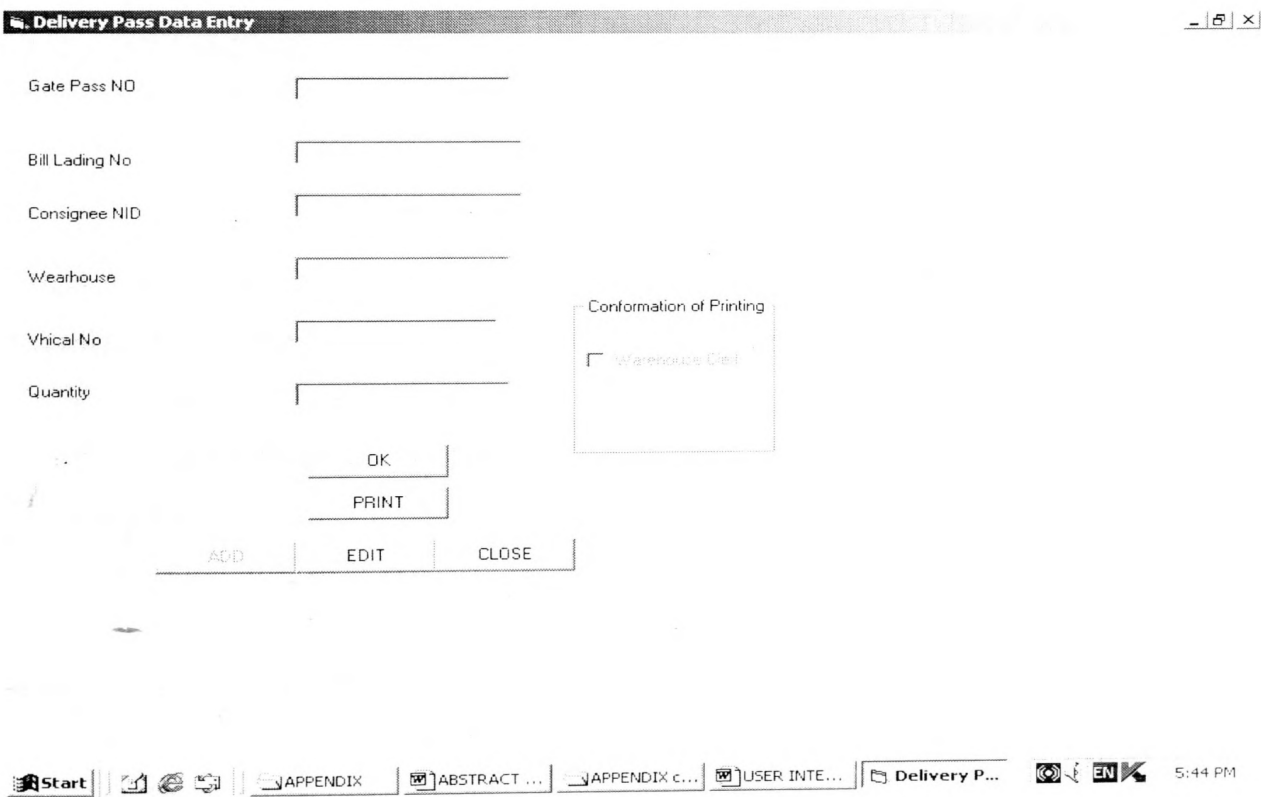
ADD EDIT CLOSE

Taskbar: Start | APPENDIX | ABSTRACT ... | APPENDIX c... | USER INTE... | Delivery P... | 5:41 PM

Delivery Pass Login



Print Delivery Pass



APPENDIX V
User Manual

THE RELEVANT OPERATION IN THE WAREHOUSE

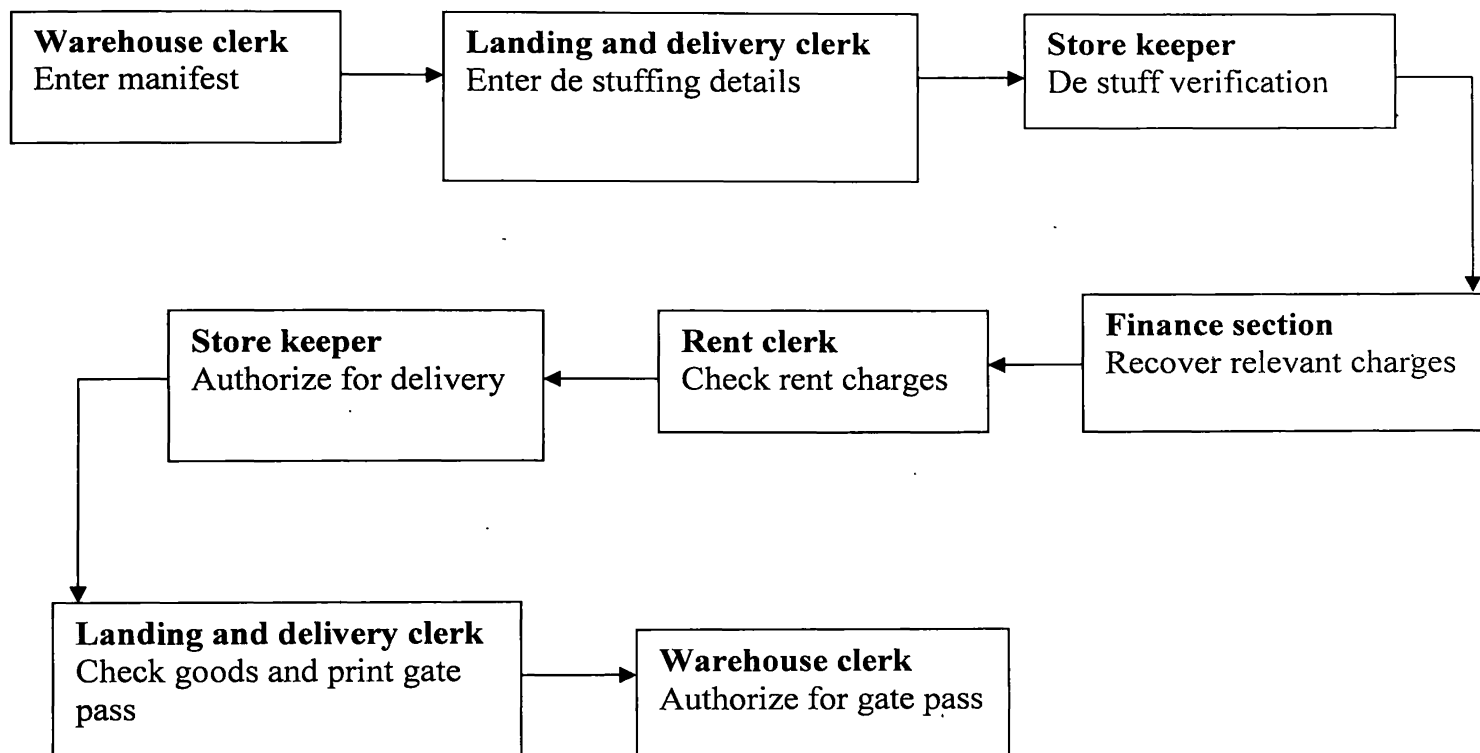


Fig.V-1

Manifest & de-stuffing operation

Warehouse clerk enter manifest date before arrived vessel to the relevant port

- Main form ---> Manifest data entry --->Vessel --->type user ID & password ---> .ok ---> add 4 fields except actual arrival date ---> next
- If a container
 - Add container details ---> next ---> add particular goods details ---> close
- If a vehicle
 - Exit from the form container ---> add particular vehicle details ---> close
- Operations edit, view, and delete can be done using these forms by the authorized employees.

Landing and delivery clerk enter landing details when de-stuffing

- Main form --->Tally data entry ---> tally header ---> type user ID & password ---> ok ---> type vessel name ---> click on the data grid ---> select relevant data row from the grid ---> click ok ---> type container no ---> click ok ---> type remaining fields ---> click add ---> click next
- Click on the grid ---> click ok ---> edit remaining data on the goods form ---> close

Store keeper verify de-stuffing data

- Main form ---> tally header --->type user ID & password ---> ok ---> type vessel Reference No---> click ok ---> type container No ---> click ok ---> verify de-stuffed data and
- if satisfied
 - click on the check box ---> next ---> click on the data grit ---> next ---> verify de-stuffed data
- if satisfied
 - click on the check box ---> close

delivery operation & Generating the gate pass

Rent clerk enter rent charges and authorized for delivery

- Main form ---> delivery ---> delivery data entry --->type user ID & password ---> ok ---> type three key fields, vessel reference no, container no, and Bill Lading no ---> click ok
- add finance and custom details ---> change the quantity if different
- Verify details and click on the check box if satisfied ---> close

Store keeper edit delivery details & authorized for deliver

- Main form ---> delivery ---> delivery data entry ---> type user ID & password ---> ok ---> type three key fields ---> click ok
- Edit authorized delivery details
- Verify details and click on the check box if satisfied ---> close

Landing and delivery clerk authorized for delivery

- Main form ---> delivery ---> delivery data entry ---> type user ID & password ---> ok ---> type three key fields ---> click ok
- Verify details and click on the check box if satisfied ---> close

Warehouse clerk authorized, for delivery & the gate pass

- Main form ---> delivery ---> delivery data entry ---> type user ID & password ---> ok ---> type three key fields ---> click ok
- Verify details and click on the check box if satisfied
- Click on the command button, “authorized for delivery pass”
- add the gate pass no ---> click close

Print the gate pass**Landing and delivery clerk print the gate pass**

- Main form ---> delivery ---> delivery pass ---> type user ID & password ---> ok --> type gate pass no ---> click ok
- Edit relevant data
- Print the gate pass

