

**ARCHIVING SOLUTION FOR TRAVELBOX**

By

**N.A.S. GUNAWARDHANA.**  
**03/AS/005**

**THIS THESES IS SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE  
IN PHYSICAL SCIENCE**


**DEPARTMENT OF PHYSICAL SCIENCE  
FACULTY OF APPLIED SCIENCES  
SABARAGAMUWA UNIVERSITY OF SRI LANKA.  
BELIHULOYA-91100**

**March 2009**

## DECLARATION

The content described in this thesis was practically implemented by me at the Codegen International Pvt.Ltd and the Faculty of Applied Sciences under the supervision of Mr. Yohan welikala and Mr. B.T.G.S.Kumara.

And the report described on this thesis has not been submitted by any one for another degree.

  
N.A.S Gunawardhana.  
(03/AS/005)


19.04.09  
Date

Certified By,  
Mr. Yohan welikala,  
External Supervisor,  
Senior software engineer,  
R & D,  
Codegen International private limited,  
Sri Lanka.


  
.....  
Signature


Date: 19.04.09

Mr.B. T.G.S Kumara,  
Internal Supervisor,  
Lecture/ Department of Physical Sciences,  
Faculty of Applied Sciences,  
Sabaragamuwa University of Sri Lanka,  
Belihuloya.

  
.....  
Signature

Date: 05/06/2009

  
Dr. Udawaththa,  
Head/Department of Physical Science,  
Faculty of Applied Sciences,  
Sabaragamuwa University of Sri Lanka,  
Belihuloya.

  
.....  
Signature

Date: 2009.06.08

*Affectionately Dedicated*  
*To*  
*My Parents*  
*And*  
*Teachers*

## **AKNOWLEDGEMENT**

First I should express my deepest gratitude to my internal supervisor Mr.B.T.G.S. Kumara, Lecture, Department of Physical Science, Faculty of Applied Sciences for his encouragement and guidance through this study.

And also I wish to express my sincere thanks to my external supervisor Mr. Yohan Welikala, Senior software engineer, R & D, Codegen International Pvt Ltd, Sri Lanka for his advice , encouragement and guidance through the study and for sparing his valuable time in bringing this study to a successful completion.

Special thanks to Mr. Prageeth Gunathilaka, , Saga Division, Codegen International Pvt Ltd, Sri Lanka for his advice, encouragement and guidance through the study and for sparing his valuable time in bringing this study to a successful completion.

I express my sincere gratitude to all managers and staff members and of the Codegen International Pvt Ltd for the providing me the opportunity to carry out the study at the Codegen International Pvt Ltd.

Finally, I express my heart-felt gratitude towards the lectures for their Corporation through out my study and my colleagues for their individual help and guidance at all times.



## ABSTRACT

Codegen International (Pvt) Limited is a leading global technology innovation services provider and it has grown beyond being an efficient provider of product and application development services to being the partner of choice in creating competitive advantage for its clients using technology solutions.

TravelBox covers all travel industry related services which optimizes more demanding options with its wide ranged technologies. From a flight/hotel reservation to supplier payments, TravelBox assists throughout the back office infrastructure setups to internet booking engines. For the maximum utilization of the tour reservation software, making way to maintain a closer customer relationship management with clients and suppliers.

TravelBox technology which is one of the software products in Codegen, is the innovation for the global travel and tourism industry and a foremost product that accommodates tomorrow's travel software solutions. But there were some problems relevant to database growth with infrequently access data (Historical Data). Software performance degradation was encountered and had to spend more money for the storages. To overcome these problems, Archiving Tool was created.

First the requirement analysis was carried out by having discussion with team members and problem owners. Then Graphical User Interface (GUI) was created according to the requirements using JBuilder. After that separated database schema was created by generating with existing database tables using Case Studio database modeling tool and utility table's data were pumped to above created schema's tables. Information Lifecycle Management (ILM) was used to implement the system using Java technologies with Generics and JIDEA 6.0 was used as java compiler. The component was tested individually and finally the integrated system was tested.

Accelerating data growth in the databases could be managed by the Archiving Solution for TravelBox by relocating closed business transactions and other infrequently accessed data to an online. So archive database could easily be accessed. And the size of the production system was reduced as well as performance of the application was improved. So at the end of the development process, the main objective of the project was achieved and the Codegen International (Pvt) Limited was satisfied with the functionalities, usability, security and reliability of the system.

## CONTENT

ABSTRACT	I
ACKNOWLEDGEMENT	II
LIST OF ABBREVIATIONS	III
LIST OF FIGURES	IV
CONTENTS	V

### Contents

Chapter 1: Introduction .....	1
1.1 Codegen International Pvt Ltd.....	1
1.1.1 Background.....	1
1.2 Project.....	1
1.2.1 Background.....	1
1.2.1.1 About existing system.....	1
1.2.1.2 Archiving solution for Travel Box.....	2
1.2.2 Objectives.....	3
Chapter 2: Literature Review.....	4
2.1 Software development methods.....	4
2.1.1 SDLC Model.....	4
2.1.1.1 System/Information Engineering and Modeling .....	4
2.1.1.2 Software Requirement Analysis.....	5
2.1.1.3 System Analysis and Design.....	6
2.1.1.4 Code Generation .....	6
2.1.1.5 Testing.....	6
2.1.1.6 Maintenance.....	6

2.1.1.7 Maintenance.....	7
2.1.1.7.1 Introduction.....	7
2.1.1.7.2 Oracle database 10g with ILM .....	7
2.1.1.7.3 Implementing ILM using Oracle database 10g.....	8
2.1.1.8 Moving Data using Partitioning.....	11
2.2 JAVA technology.....	11
2.2.1. JAVA Framework.....	12
2.2.1.1 Frameworks and open source.....	13
2.2.1.2 Types of Frameworks.....	13
2.2.2 Advantage of JAVA.....	15
2.3 JAVA Language.....	14
2.3.1. Introduction.....	14
2.3.2. Object oriented features.....	15
2.3.2.1 Introduction.....	15
2.3.2.2 Class.....	15
2.3.2.3 Object.....	17
2.3.2.4 Inheritance.....	19
2.3.2.5 Encapsulation.....	20
2.3.2.6 Interfaces.....	20
2.3.2.7 Abstraction.....	21
2.3.2.8 Polymorphism.....	23
2.4 Database connection with Java.....	23
2.5 Oracle Database.....	24
2.5.1 What is a database?.....	24
2.5.1.1 Database Structure in Oracle.....	25

2.5.1.2 Relational Databases.....	26
2.5.1.3 Primary Keys.....	27
2.5.1.4 Foreign Key.....	27
Chapter 03: Methodology .....	28
3.1. Requirement Analysis.....	28
3.1.1. Functional Requirement Analysis.....	28
3.2. System Analysis and Designing.....	29
3.2.1. Data and Database design.....	29
3.2.2. Interfaces and form Design.....	29
3.3. Coding and Implementation.....	31
3.3.1. Coding.....	31
3.3.2. Implementation.....	32
3.4. Testing.....	32
3.4.1. Unit Testing.....	32
3.4.2. System Testing.....	32
Chapter 04: Result and Discussion .....	33
Chapter 05: Conclusion.....	37
Reference.....	38

## **LIST OF ABBREVIATIONS**

ILM	- Information Lifecycle management
SDLC	- System Development Life Cycle Model
DBMS	- Database Management System
IDE	- Integrated Development Environment
MS	- Microsoft
OOP	- Object Oriented Programming
OS	- Operating System
PMON	- Process Monitor
SDLC	- System Development Life Cycle
SMON	- System Monitor
TB	- Terabyte

## LIST OF FIGURES

Figure 2.1	- General Overview of waterfall Model.....	05
Figure 2.2	- Partition data classes.....	09
Figure 2.3	- illustrates the allocating Data Classes to a Partition.....	09
Figure 2.4	- data usage over time.....	10
Figure 2.5	- A software object.....	17
Figure 2.6	- A bicycle modeled as a software object.....	18
Figure 2.7	- A hierarchy of bicycle classes.....	19
Figure 2.8	- Tables relationship.....	27
Figure 3.1	- Database Modeling Tool.....	29
Figure 3.2	- Booking Search Panel.....	30
Figure 3.2	- Database Table View Panel.....	30
Figure 4.1	- Reservation Booking Search panel.....	34
Figure 4.2	- Archiving Scheduling panel.....	35
Figure 4.3	- Table Data View panel .....	36

# **Chapter 1**

## **Introduction**

### **1.1 About Codegen International Pvt Ltd**

Codegen International Pvt Ltd is a leading global technology innovation services provider that creates competitive advantage for its clients. It was established in 1999, with offices in London, UK and Colombo Sri Lanka, and continues to deliver and maintain solutions around the world. CodeGen designs and builds innovation, cost effective technology solutions for the Travel and Tourism, Finance and Banking, and E-Commerce Industries. CodeGen focuses on making the best use of the latest technologies available in order to satisfy growing demand for fast, efficient and effective solutions. Developing close working relationships with our partners ensure ongoing feasibility studies continue to deliver solutions that maintain a competitive advantage.

#### **1.1.1 Existing system**

The Travel Box, software is a web based application written in Java and developed using the IntelliJIDEA Version 6.0 development environment and associated plug-ins. This product provides a means for the distribution and management of travel and tourism products via a multi channel model that incorporates interfaces such as multiple online booking websites, multiple call centers or retail outlets and data loading teams.

The software itself is built on a growing suite of open sourced modules called the Travel Box Family which are highly flexible web services based modules which can be included or excluded as required, to match any given business model and for any size organization from a single platform. Flights, Accommodation, Insurance, Car Hire, Cruise, Transfers and various other products can be sold as single inventory items, tailor made to specific requirements or creatively packaged in ways such as dynamic, semi dynamic and brochure packaging across all sales channels and brands with varying levels of applied markup.

The Travel Box, software provides an end to end solution covering the four main areas of most business operations; Presales, Sales, Post Sales and Accounts. Each

module is designed for a specific task, combined with flexible user rights and real-time reporting; business processes are optimized to increase productivity and profitability.

Travel Box Technology gives the versatility to support complex product ranges. Own inventory contracts can be combined with Seamless XML Integration to global distribution service providers, and/or providers of other third-party products and/or information to create rich content.

From sourcing product to automated supplier payments, Travel Box Technology provides an End-to End solution covering the four main areas of most Business Operations;

Pre-Sales, Sales, Sales and Accounts. Each module is designed for a specific task, combined with flexible user rights and real-time reporting, business processes are optimized to increase productivity and profitability.

### **1.1.2 Archiving solution for Travel Box**

Archiving Solution for Travel Box manages accelerating data growth by relocating closed business transactions and other infrequently accessed data to an online, easily accessed archive database, significantly reducing the size of the production system and improving application performance.

By reducing the size of the production database, Archiving Solution for Travel Box can boost application performance, improve application stability, and reduce application environment costs. Furthermore, Archiving Solution for Travel Box retains transparent, online, real-time access to both production and archive data, using the source applications interfaces, forms, and reports. In this way, Archiving Solution for Travel Box helps improve application performance, reduce storage costs, and meet regulatory requirements.

To seize control of the situation, today's smart IT organizations take an ILM approach, dedicating high-end resources to production data while relocating historical transactions to online archives stored on more cost-effective servers and storage



classes. Online archiving is an indispensable key to managing the application data lifecycle while ensuring regulatory compliance.

Here actual process is moving the data which is less active, to another storage tiers (less active storage tier or historical tier) using partitioning the existing table to another storage tier. Travel Box system has thousands of table and those tables have Tb size of data. Because growth of this table data, it is problem to Travel Box's performance and its satiability.

## **1.2. Objectives**

### **1.2.1 Overall Objectives:**

Implement the Archiving Solution for TravelBox for Codegen International Pvt Ltd using JAVA and Oracle.

### **1.2.2 Specific Objectives:**

- Literature survey about Java, Oracle.
- Literature survey about RIM for databases system.
- Requirement analysis about archiving solution for TravelBox.
- Retain application transparency
- Implementing Database
- Placement into storage tiers:
- System changes and implementing new setup.
- Existing data preparation and upload.
- System testing
- Live run with maintenance.

## Chapter 2

### Literature Review

#### 2.1 Software development methods

The basic popular models used by many software development firms are as follows:

- A) SDLC Model
- B) Prototyping Model
- C) Rapid Application Development Model
- D) Component Assembly Model

##### 2.1.1 SDLC Model:

This is also called as Classic Life Cycle Model, Linear Sequential Model or Waterfall Method. This model has the following activities.

1. System/Information Engineering and Modeling
2. Software Requirements Analysis
3. Systems Analysis and Design
4. Code Generation
5. Testing
6. Maintenance

##### 2.1.1.1 System/Information Engineering and Modeling

The article in [buzzle.com](http://www.buzzle.com) described System/Information Engineering and Modeling as “As software development is large process so work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. The view of this system is necessary when software must interface with other elements such as hardware, people and other resources. System is the very essential requirement for the existence of software in any entity. In some cases for maximum output, the system should be re-engineered and spruced up. Once the ideal system is designed according to requirement, the development team studies the software requirement for the system”. (<http://www.buzzle.com/editorials/3-13-2005-67039.asp> , 2008.12.15).

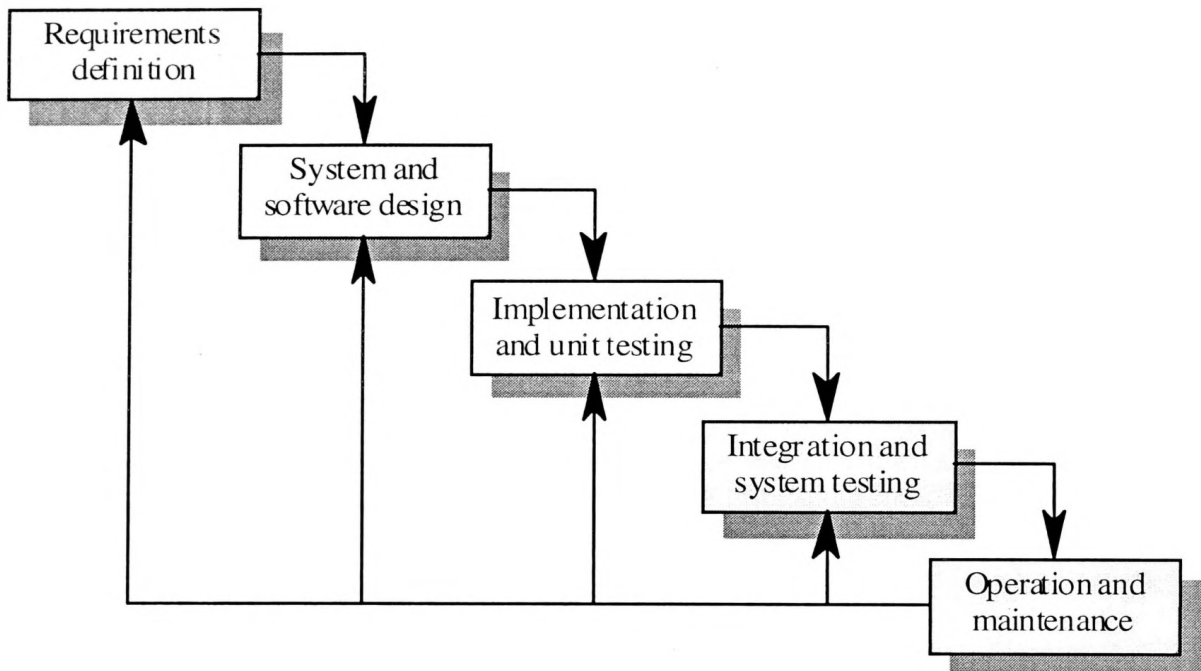


Figure 2.1: General Overview of Water Fall Model

### 2.1.1.2 Software Requirement Analysis

The article in [buzzle.com](http://www.buzzle.com) described Software Requirement Analysis as “Software Requirement Analysis is also known as feasibility study. In this requirement analysis phase, the development team visits the customer and studies their system requirement. They examine the need for possible software automation in the given software system. After feasibility study, the development team provides a document that holds the different specific recommendations for the candidate system. It also consists of personnel assignments, costs of the system, project schedule and target dates.

The requirements analysis and information gathering process is intensified and focused specially on software. To understand what type of the programs to be built, the system analyst must study the information domain for the software as well as understand required function, behavior, performance and interfacing. The main purpose of requirement analysis phase is to find the need and to define the problem that needs to be solved”. (<http://www.buzzle.com/editorials/3-13-2005-67039.asp> , 2008.12.15)

### **2.1.1.3 System Analysis and Design**

The article in [buzzle.com](http://www.buzzle.com/editorials/3-13-2005-67039.asp) described System Analysis and Design as “In System Analysis and Design phase, the whole software development process, the overall software structure and its outlay are defined. In case of the client/server processing technology, the number of tiers required for the package architecture, the database design, the data structure design etc are all defined in this phase. After designing part a software development model is created. Analysis and Design are very important in the whole development cycle process. Any fault in the design phase could be very expensive to solve in the software development process. In this phase, the logical system of the product is developed”. (<http://www.buzzle.com/editorials/3-13-2005-67039.asp> , 2008.12.15).

### **2.1.1.4 Code Generation**

In Code Generation phase, the design must be decoded into a machine-readable form. If the design of software product is done in a detailed manner, code generation can be achieved without much complication. For generation of code, Programming tools like Compilers, Interpreters, and Debuggers are used. For coding purpose different high level programming languages like C, C++, Pascal and Java are used. The right programming language is chosen according to the type of application.

### **2.1.1.5 Testing**

After code generation phase the software program testing begins. Different testing methods are available to detect the bugs that were committed during the previous phases. A number of testing tools and methods are already available for testing purpose.

### **2.1.1.6 Maintenance**

The article in [buzzle.com](http://www.buzzle.com) described Maintenance as “Software will definitely go through change once when it is delivered to the customer. There are large numbers of reasons for the change. Change could happen due to some unpredicted input values into the system. In addition to this the changes in the system directly have an effect on

the software operations. The software should be implemented to accommodate changes that could be happen during the post development period. (<http://www.buzzle.com/editorials/3-13-2005-67039.asp> , 2008.12.15).

### **2.1.1.7 Implementing ILM lifecycle model**

#### **2.1.1.7.1 Introduction**

The article in hp.com described ILM lifecycle model as “The article Although most organizations have long regarded their stores of data as one of their most valuable corporate assets, originally, data was used to help achieve operational goals, run the business and help identify the future direction and success of the company.

Information Lifecycle Management, no longer have to be concerned about issues such as which database files the storage manager can handle, because, the database manages its own files in an ILM environment.

Information Lifecycle Management (ILM) is designed to address these issues, with a combination of processes, policies, software and hardware so that the appropriate technology can be used for each phase of the lifecycle of the data.

Oracle Database 10g already contains a rich feature set to support and comply with the ever changing demands required by organizations. Oracle Database 10g can be used to build an ILM strategy”. (<http://www.hp.com/storage/storagegrid>, 2008.10.08).

#### **2.1.1.7.2 Oracle database 10g with ILM**

Oracle Database 10g provides the ideal environment for implementing your ILM solution, because it offers a cost-effective solution that is secure, transparent to the application and achieves all of this without compromising performance. It also already has the ability to:

- Manage storage
- Use database features beneficial for ILM
- Protect data

Oracle Database 10g is capable of storing both structured and unstructured data. Therefore, using Oracle means that all your data is now much easier to manage, because it is all in one place, instead of being stored in multiple file systems.

### **2.1.1.7.3 Implementing ILM using Oracle database 10g**

The Oracle white paper Information Lifecycle Management for Business Data presents an overview of implementing ILM on Oracle and it explains how data can be described as being; active, less active or historical. Building an Information Management Lifecycle solution, using Oracle Database 10g is quite straightforward, and it can be completed, by following these three simple steps:

1. Define the Data Classes
2. Create Storage Tiers for the Data Classes
3. Manage Access and Migration of Data by Class

#### **➤ Step 1 – Define the Data Classes**

There are many ways that data can be classified, the most common type of classification is by age or date, but other types are possible, such as by product or privacy, or a hybrid classification could be used such as by privacy and age.

Consider a system where all the orders received by a company are stored. If date is chosen as the class, then when the data is grouped at the row level by their class, which in this example would be the date of the order, all orders for Q1 can be managed as a self contained unit, and where as the orders for Q2 would reside in a different class. In Oracle these classes can be implemented by using partitioning, and since partitions are completely transparent to the application, the data is physically separated, but the application still sees all of the orders.

Partitioning involves physically placing data according to a data value, and a frequently used technique is to store information, for example, by date. Figure 2 illustrates a scenario where all the orders for Q1, Q2, Q3 and Q4 would be stored in individual partition and the orders for previous years in other partitions.

There are a number of benefits to partitioning data, because it provides an easy way to distribute the data across the appropriate storage devices depending on its usage, whilst still keeping the data online and stored on the most cost-effective device. Since partitioning is completely transparent to anyone accessing the data, no applications changes are required. Other benefits of using partitions are that each partition can have its own local index, and when the optimizer uses partition elimination, queries

will only access the relevant partitions instead of all partitions, thus improving query response time.

Oracle offers several different types of partitioning techniques; such as range, hash, composite and list partitioning, but range partitioning is one of the most frequently used types for ILM.

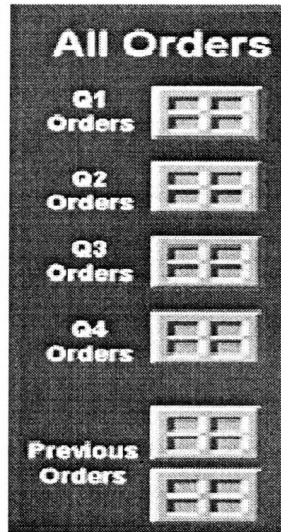


Figure 2.2 Partition data classes

Below figure illustrates the allocating Data Classes to a Partition:

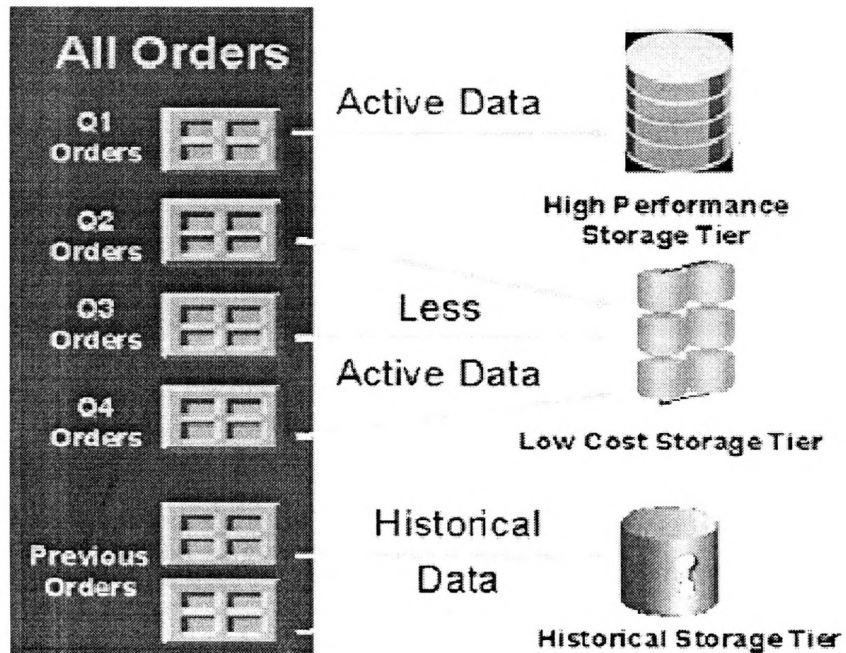


Figure 3.3 Allocating Data classes to partition.

➤ **Step 2 – Create Storage Tiers for the Data Classes**

Figure 4 illustrates how data is used over a period of time. Using this information, it can then be determined that to retain all this information, three storage tiers will be required to hold all of the data:

- High Performance
- Low Cost
- Historical

The high performance storage tier is where all the important and frequently accessed data would be stored, such as the partition holding our most recent Q1 orders. This would utilize the smaller, faster disks on high performance storage devices.

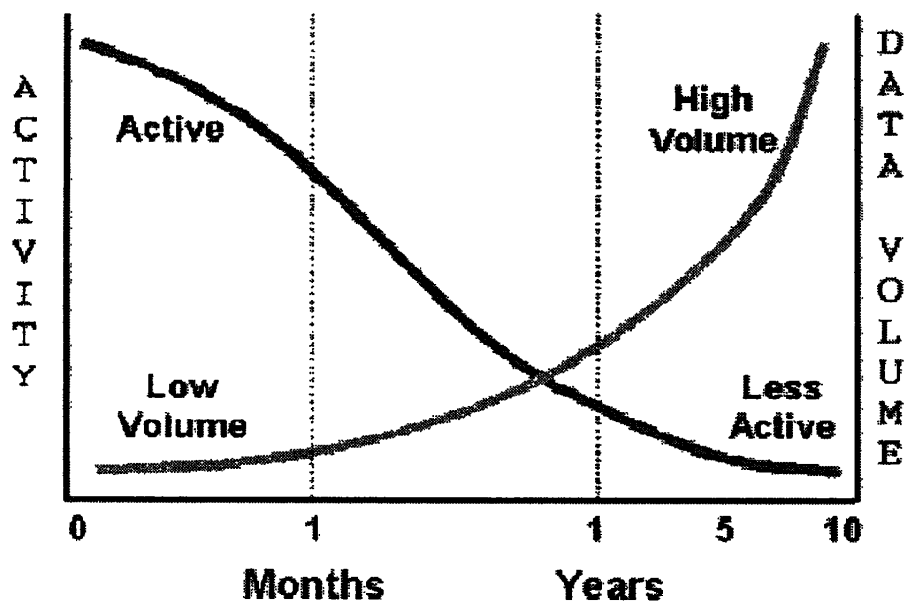


Figure 2.4 Data usage over time



### ➤ **Step 3 – Manage Access and Migration of Data by Class**

The final step is to ensure that only authorized Controlling Access to Data. The security of your data is another very important part of ILM because the access rights to the data may change during its lifetime. In addition there may also be legal and regulatory requirements that are imposed upon data when it is held in electronic format which must be complied with. For example organizations such as the tax authorities may demand that you show that electronic data is secure from unauthorized access and changes, and can report when changes were made and by whom. Securing the data is considerably easier when it is held inside an Oracle Database, because it can be secured using database security features such as:

- Database Security
- Virtual Private Database
- Audit Trails
- Encryption

Implementing ILM users may access the data and as the data ages, there are a number of techniques that can be used to migrate the data between the storage tiers.

#### **2.1.1.8 Moving Data using Partitioning**

Data not only has to be secure, but it will also have to be moved during its lifetime. Incorporating partitioning into the database design can facilitate very easy movement of data as its usage changes during its lifetime. When the information in that partition is no longer being regularly accessed, the partition can be moved with the single MOVE PARTITION command, to the inexpensive ATA disks.

#### **2.2 JAVA technology**

Java technology is an object-oriented, platform-independent, multithreaded programming environment. It is the foundation for Web and networked services, applications, platform-independent desktops, robotics, and other embedded devices. Java technology has become a complete software ecosystem that represents different values to different types of consumer and business users. It offers developers a choice of three Java platform editions depending on the need:

- Java technology in small and mobile devices
- Java technology in PC desktops
- Java technology in medium to large businesses

### **2.2.1. JAVA Framework**

The JAVA Framework is managed, type-safe environment for application development and execution. The framework manages all aspects of the execution of program: it allocates memory for the storage of data and instructions, grants or denies the appropriate permissions to your application, initiates and manages application execution, and manages the reallocation of memory for resources that are no longer needed.

Java frameworks span everything from the simplistic to the entirely comprehensive. Some frameworks are concentrating on user interface development and web applications. An application framework is a tool that provides both structure and services to application. Unlike an IDE or compiler, a framework usually will be deployed along with application. Exactly what the structure looks like, or what the services are, depends on the particular framework. Some framework provides the some services ranging from user interface connectivity through to persistence and database access.

A good application framework provides proven and reliable design patterns and code architecture to an application, increasing development productivity many fold. Frameworks grew out of a realization that many common design patterns could have a single implementation, and that implementation could be used on multiple projects.

As open source Java grew in popularity, many application problems turned out to have common solutions. These solutions began to be abstracted into design patterns ways of doing certain things that had been tried and found to be successful. As implementations of these patterns were saved and used again, code re-use began to grow between projects, and the most common solutions were soon grouped together into the first frameworks.

### **2.2.1.1 Frameworks and open source**

Frameworks are a good fit for the open source model in that the more they are used, the more flexible they tend to become, assuming the initial design was good. Often, a framework is built as the result of an iterative process: As projects that have some common elements are developed, reusable components are created to reduce the need to continually re-create typical functionality. As these components are re-used and made more generic and flexible, they become the backbone of a framework.

Many frameworks evolve to encompass services for the entire range of needs of an application. Open source means that a framework of this type will be extended and new services added by a number of different developers, creating services and capabilities that a single group might not ever have time to build for themselves, and increasing the utility of the framework for everyone.

### **2.2.1.2 Types of Frameworks**

Frameworks designed for general application development often provide more capability in particular areas of the application development process. For example, the Apache Struts framework provides a lot of functionality in the area of user-interface building, whereas a framework like spring doesn't touch the user-interface area much at all.

Some frameworks take a more pattern-oriented approach, supplying little of their own code; instead, they concentrate on the structure and overall architecture of an application, providing in essence a "backbone" into which other components can be connected. Often, a framework is broken into several sub-projects, each provides functionality for a particular area, such as user interface, persistence, scheduling, and so forth. Ideally, these sub-projects should be both optional and replaceable, giving the maximum flexibility to the developer.

### **2.2.2 Advantage of JAVA**

Java™ has significant advantages over other languages and environments that make it suitable for just about any programming task.

The advantages of Java are as follows:

- Java is easy to learn.

Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.

- Java is object-oriented.

This allows you to create modular programs and reusable code.

- Java is platform-independent.

One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.

Because of Java's robustness, ease of use, cross-platform capabilities and security features, it has become a language of choice for providing worldwide Internet solutions.

## **2.3 JAVA Language**

### **2.3.1. Introduction**

Java is a high-level, third generation programming language, like C, Smalltalk, Perl, and many others. Java can be used to write computer applications that crunch numbers, process words, play games, store data or do any of the thousands of other things computer software can do.

What's most special about Java in relation to other programming languages is that it lets you write special programs called applets that can be downloaded from the Internet and played safely within a web browser.

Traditional computer programs have far too much access to your system to be downloaded and executed willy-nilly. Although you generally trust the maintainers of various ftp archives and bulletin boards to do basic virus checking and not to post destructive software, a lot still slips through the cracks. Even more dangerous software would be promulgated if any web page you visited could run programs on your system. You have no way of checking these programs for bugs or for out-and-out malicious behaviour before downloading and running them. Java is an object-

oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and that can be used to create applications or applets. Because of its rich set of API's, similar to Macintosh and Windows, and its platform independence, Java can also be thought of as a platform in itself. Java also has standard libraries for doing mathematics.

Much of the syntax of Java is the same as C and C++. One major difference is that Java does not have pointers. However, the biggest difference is that you must write object oriented code in Java. Procedural pieces of code can only be embedded in objects.

### **2.3.2. Object oriented features**

#### **2.3.2.1 Introduction**

Object-Oriented Programming (OOP) is an approach to program organization and development that attempts to eliminate some of the pitfalls of conventional programming methods by incorporating the best of structured programming features with several powerful new concepts. It is a new way of organizing and developing programs and has nothing to do with any particular language. However, not all languages are suitable to implement the OOP concepts easily.

#### **2.3.2.2 Class**

The article in [java.sun.com](http://java.sun.com) described class as "A class is a blueprint or prototype from which objects are created. This section defines a class that models the state and behavior of a real-world object. It intentionally focuses on the basics, showing how even a simple class can cleanly model state and behavior. For example, the 'class of Bicycle' might be a set which includes the various kind of bicycles. A class is the basis of modularity and structure in an object-oriented computer program. A class should typically be recognizable to a non-programmer familiar with the problem domain, and the code for a class should be (relatively) self-contained and independent (as should the code for any good non-OOP function). With such modularity, the structure of a program will correspond to the aspects of the problem that the program is intended to solve. This simplifies the mapping to and from the problem and program". (<http://java.sun.com/docs/books/tutorial/java/concepts/object.html>, 2008.11.23).

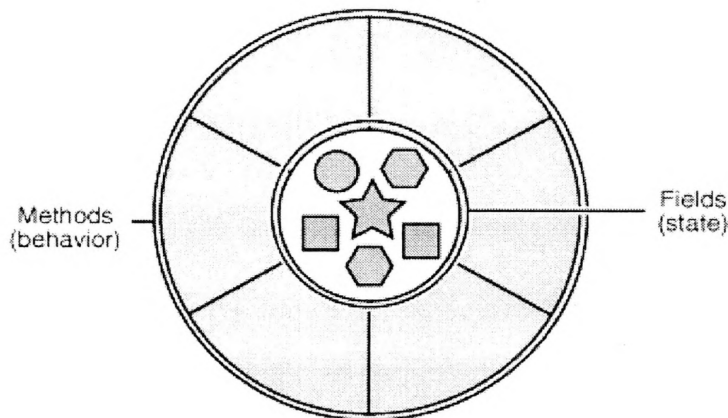
The syntax of the Java programming language for bicycle class.

```
class Bicycle {  
    int cadence = 0;  
    int speed = 0;  
    int gear = 1;  
  
    void changeCadence(int newValue) {  
        cadence = newValue;  
    }  
  
    void changeGear(int newValue) {  
        gear = newValue;  
    }  
  
    void speedUp(int increment) {  
        speed = speed + increment;  
    }  
  
    void applyBrakes(int decrement) {  
        speed = speed - decrement;  
    }  
  
    void printStates() {  
        System.out.println("status");  
    }  
}
```

### 2.3.2.3 Object

The article in [java.sun.com](http://java.sun.com) described Object as “An object is a software bundle of related state and behavior. Software objects are often used to model the real-world objects that you find in everyday life. State and behavior are represented within an object, introduces the concept of data encapsulation.

Real-world objects share two characteristics: They all have state and behavior. Bicycles have state (current gear, current pedal cadence, and current speed) and behavior (changing gear, changing pedal cadence, applying brakes). Identifying the state and behavior for real-world objects is a great way to begin thinking in terms of object-oriented programming.



**Figure 2.5 A software object.**

Software objects are conceptually similar to real-world objects: they too consist of state and related behavior. An object stores its state in fields (variables in some programming languages) and exposes its behavior through methods (functions in some programming languages). Methods operate on an object's internal state and serve as the primary mechanism for object-to-object communication. Hiding internal state and requiring all interaction to be performed through an object's methods is known as data encapsulation a fundamental principle of object-oriented programming.

Consider a bicycle, for example:

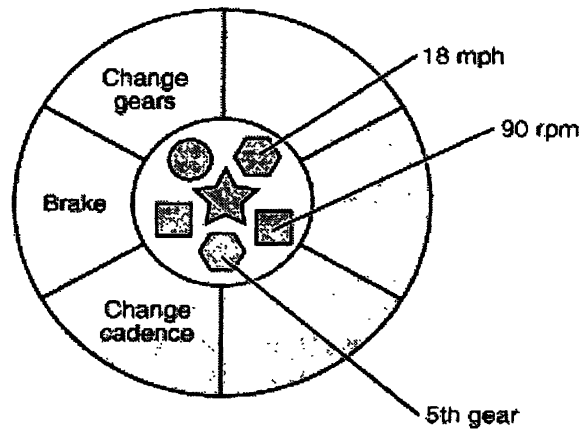


Figure 2.6 A bicycle modeled as a software object.

By attributing state (current speed, current pedal cadence, and current gear) and providing methods for changing that state, the object remains in control of how the outside world is allowed to use it. For example, if the bicycle only has 6 gears, a method to change gears could reject any value that is less than 1 or greater than 6.

Bundling code into individual software objects provides a number of benefits, including:

- 1 Modularity: The source code for an object can be written and maintained independently of the source code for other objects. Once created, an object can be easily passed around inside the system.
- 2 Information-hiding: By interacting only with an object's methods, the details of its internal implementation remain hidden from the outside world.
- 3 Code re-use: If an object already exists (perhaps written by another software developer), you can use that object in your program. This allows specialists to implement/test/debug complex, task-specific objects, which you can then trust to run in your own code.
- 4 Pluggability and debugging ease: If a particular object turns out to be problematic, you can simply remove it from your application and plug in a different object as its replacement. This is analogous to fixing mechanical problems in the real world. If a bolt breaks, you replace it, not the entire machine.



Method (also known as message) how code can be used an object of some class. A method is a form of subroutine operating on a single object. Sometimes access to the data of an object is restricted to the methods of its class. A member of a class or object is a method or a data item describing the state of an object. In some languages the general term is feature”. (<http://java.sun.com/docs/books/tutorial/java/concepts/object.html>, 2008.11.23).

#### 2.3.2.4 Inheritance

The article in java.sun.com described Inheritance as “Inheritance provides a powerful and natural mechanism for organizing and structuring your software. This section explains how classes inherit state and behavior from their super classes, and explains how to derive one class from another using the simple syntax provided by the Java programming language.

Object-oriented programming allows classes to inherit commonly used state and behavior from other classes. In this example, Bicycle now becomes the super class of MountainBike, RoadBike, and TandemBike. In the Java programming language, each class is allowed to have one direct superclass, and each superclass has the potential for an unlimited number of subclasses:

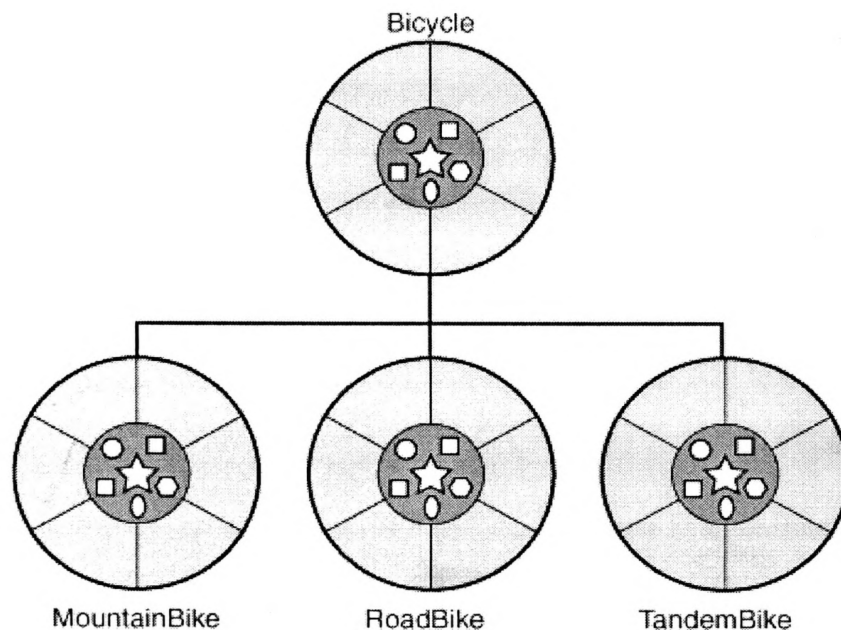


Figure 2.7 A hierarchy of bicycle classes.

The syntax for creating a subclass is simple. At the beginning of your class declaration, use the extends keyword, followed by the name of the class to inherit from:

```
Class Mountain Bike extends Bicycle {  
    // new fields and methods defining a mountain bike would go here  
}
```

This gives Mountain Bike all the same fields and methods as Bicycle, yet allows its code to focus exclusively on the features that make it unique. This makes code for your subclasses easy to read. However, you must take care to properly document the state and behavior that each super class defines, since that code will not appear in the source file of each subclass”. (<http://java.sun.com/docs/books/tutorial/java/concepts/inheritance.html>, 2008.11.23).

### **2.3.2.5 Encapsulation**

The article in Wikipedia described Encapsulation as “Ensuring that code outside a class sees only functional details of that class, but not implementation details. The latter are liable to change, and could allow a user to put an object in an inappropriate state. Encapsulation is achieved by specifying which classes may use the members of an object. The result is that each object exposes to any class a certain interface those members accessible to that class. For example, an interface can ensure that puppies can only be added to an object of the class Dog by code in that class. Members are often specified as public, protected and private, determining whether they are available to all classes, sub-classes or only the defining class. Some languages go further: Java uses the protected keyword to restrict access also to classes in the same package.” ([http://en.wikipedia.org/wiki/Object\\_oriented\\_programming](http://en.wikipedia.org/wiki/Object_oriented_programming), 2008.12.19).

### **2.3.2.6 Interfaces**

The article in java.sun.com described Interface as “An interface is a contract between a class and the outside world. When a class implements an interface, it promises to provide the behavior published by that interface. This section defines a simple interface and explains the necessary changes for any class that implements it.

As you've already learned, objects define their interaction with the outside world through the methods that they expose. Methods form the object's interface with the outside world; the buttons on the front of your television set, for example, are the interface between you and the electrical wiring on the other side of its plastic casing. You press the "power" button to turn the television on and off.

In its most common form, an interface is a group of related methods with empty bodies. A bicycle's behavior, if specified as an interface, might appear as follows:

```
interface Bicycle {  
    void changeCadence(int newValue);  
    void changeGear(int newValue);  
    void speedUp(int increment);  
    void applyBrakes(int decrement);  
}
```

To implement this interface, the name of your class would change (to `ACMEBicycle`, for example), and you'd use the `implements` keyword in the class declaration:

```
Class ACMEBicycle implements Bicycle {  
    // remainder of this class implemented as before  
}
```

Implementing an interface allows a class to become more formal about the behavior it promises to provide. Interfaces form a contract between the class and the outside world, and this contract is enforced at build time by the compiler. If your class claims to implement an interface, all methods defined by that interface must appear in its source code before the class will successfully compile".(<http://java.sun.com/docs/books/tutorial/java/concepts/interface.html>, 2008.11.23).

### 2.3.2.7 Abstraction

The article [java.sun.com](http://java.sun.com) described Abstraction as "An abstract class contains one or more abstract methods. Generally you want to create an abstract class when you are

creating several objects that will share some method functionality, and differ in some method functionality. The methods that function the same way for every object will be implemented in the abstract super class, whereas the methods that differ in functionality for each object will be declared abstract in the abstract super class, and implemented in each of the subclasses". (<http://java.sun.com/docs/books/tutorial/java/andl/abstract.html>, 2008.11.23)

Take for example, an abstract class that models electronics.

```
public abstract class Electronic
{
    boolean powerOn = false;

    public boolean isOn()
    {
        return powerOn;
    }

    public void turnOn()
    {
        powerOn = true;
        System.out.println("Turned on");
    }

    public void turnOff()
    {
        powerOn = false;
        System.out.println("Turned off");
    }

    public abstract void operate();
}
```

### 2.3.2.7 Polymorphism

The book Object Oriented programming in Java, 1<sup>st</sup> edition described Polymorphism as “Polymorphism is the capability of an action or method to do different things based on the object that it is acting upon. This is the third basic principle of object oriented programming. Overloading and overriding are two types of polymorphism. Now look at the third type: dynamic method binding. Polymorphism is behavior that varies depending on the class, in which the behavior is invoked,

Overloaded methods are methods with the same name signature but either a different number of parameters or different types in the parameter list. For example 'spinning' a number may mean increase it, 'spinning' an image may mean rotate it by 90 degrees. By defining a method for handling each type of parameter you achieve the effect that you want.

Overridden methods are methods that are redefined within an inherited or subclass. They have the same signature and the subclass definition is used”.( Mitchell, W. and Robert, L. (1997), Object Oriented programming in Java, 1<sup>st</sup> edition, Stephen Gilgert and Bill Mccarty publishers, 953p).

### 2.4 Database connection with Java

The article in islandnet.com described Database connection with Java as “Java Database Connectivity (JDBC) is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards relational databases.

Java Database Connectivity (JDBC) is a programming framework for Java developers writing programs that access information stored in databases, spreadsheets, and flat files. JDBC is commonly used to connect a user program to a "behind the scenes" database, regardless of what database management software is used to control the database. In this way, JDBC is cross-platform A database that another program links to is called a data source. Many data sources, including products produced by Microsoft and Oracle, already use a standard called Open Database Connectivity (ODBC)”.(<http://www.islandnet.com/~tmc/html/articles/orareln.htm> ,2009.02.15).

The example below shows creating a statement and executing a query.

```
Statement stmt = conn.createStatement();
```

```

try {
    ResultSet rs = stmt.executeQuery( "SELECT * FROM MyTable" );
    try {
        while ( rs.next() ) {
            int numColumns = rs.getMetaData().getColumnCount();
            for ( int i = 1 ; i <= numColumns ; i++ ) {
                System.out.println( "COLUMN " + i + " = " + rs.getObject(i) );
            }
        }
    } finally {
        rs.close();
    }
} finally {
    stmt.close();
}

```

## 2.5 Oracle Database

### 2.5.1 What is a Database?

The book *Oracle Data Warehousing*, 1<sup>st</sup> edition described Database as “A database management system, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, Paradox, IMS, and Oracle. These systems allow users to create, update, and extract information from their databases. Compared to a manual filing system, the biggest advantages to a computerized database system are speed, accuracy, and accessibility.

A database is a structured collection of data. Data refers to the characteristics of people, things, and events. Oracle stores each data item in its own field. For example, a person's first name, date of birth, and their postal code are each stored in separate fields.

In Oracle, the fields relating to a particular person, thing, or event are bundled together to form a single, complete unit of data, called a record (it can also be referred to as a row or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name. During an Oracle database design project, the analysis of your business needs identifies all the fields or attributes of interest". (Bounie, O. and Michael, S. (1997), *Oracledata Warehousing*, 1<sup>st</sup> edition, Techmedia publishers, 699p, 1997).

#### **2.5.1.1 Database Structure in Oracle**

The article in Wikipedia described Database Structure in Oracle as "An Oracle database comprises an instance and data storage. The instance comprises a set of operating system processes and memory structures that interact with the storage. Typical processes include PMON (the process monitor) and SMON (the system monitor).

Oracle users refer to the server-side memory structure as the SGA (System Global Area). The SGA typically holds cache information like data buffers, SQL commands and user information. In addition to storage, the database consists of online redo logs (which hold transactional history). Processes can in turn archive the online redo logs into archive logs (offline redo logs), which provide the basis (if necessary) for data recovery and for some forms of data replication.

The Oracle RDBMS stores data logically in the form of table spaces and physically in the form of data files. Table spaces can contain various types of segments, for example, Data Segments, Index Segments etc. Segments in turn comprise one or more extents. Extents comprise groups of contiguous data blocks. Data blocks from the basic units of data storage. At the physical level, data files comprise one or more data blocks, where the block size can vary.

Oracle keeps track of its data storage with the help of information stored in the SYSTEM table space. The SYSTEM table space contains the data dictionary - and often (by default) indexes and clusters. (A data dictionary consists of a special collection of tables that contains information about all user objects in the database). Since version 8i, the Oracle RDBMS also supports "locally managed" table spaces

which can store space management information in bitmaps in their own headers rather than in the SYSTEM table space.

If the Oracle database administrator has instituted Oracle RAC (Real Application Clusters), then multiple instances, usually on different servers, attach to a central storage array. This scenario offers numerous advantages, most importantly performance, scalability and redundancy. However, support becomes more complex, and many sites do not use RAC. In version 10g, grid computing has introduced shared resources where an instance can use (for example) CPU resources from another node (computer) in the grid.

The Oracle DBMS can store and execute stored procedures and functions within it. Such code can operate under PL/SQL (Oracle Corporation's proprietary procedural extension to SQL), or under the object-oriented language Java.

An Oracle database installation traditionally comes with a default schema called Scott. After the installation process has set up the sample tables, the user can log into the database with the username Scott and the password tiger. (The name of the "Scott" schema originated with Bruce Scott, one of the first employees at Oracle (then Software Development Laboratories), who had a cat named Tiger.)“ ([http://en.wikipedia.org/wiki/Oracle\\_database](http://en.wikipedia.org/wiki/Oracle_database), 2008.11.23).

### **2.5.1.2 Relational Databases**

A relational database is a database that groups data using common attributes found in the data set. The resulting "clumps" of organized data are much easier for people to understand.

For example, a data set containing all the real estate transactions in a town can be grouped by the year the transaction occurred; or it can be grouped by the sale price of the transaction; or it can be grouped by the buyer's last name; and so on.

Such a grouping uses the relational model (a technical term for this schema). Hence such a database is called a "RELATIONAL DATABASE."

The software used to do this grouping is called a relational database management system. The term "relational database" often refers to this type of software.



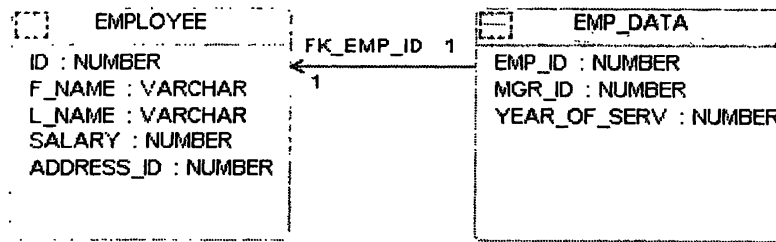


Figure 2.8 Tables relationship

Oracle makes it very easy to link the data in multiple tables: matching an employee to the EMP\_DATA in which they work is one example. This is a key feature of a relational database management system, or RDBMS. They store data in two or more tables and enable you to define relationships between the tables. The link between the tables is based on one or more field values common to both tables.

### 2.5.1.3 Primary Keys

Every table in Oracle has a field or a combination of fields that uniquely identifies each record in the table. This unique identifier is called the primary key, or simply the key. The primary key provides the means to distinguish one record from all the others in a table. It allows the user and the database system to identify, locate, and refer to one particular record in the table.

The database design team determines the best candidate field for the primary key. The employee's first and last names together could be a primary key that is until a new employee with the same name is hired.

Once a table has been assigned a primary key, Oracle won't allow more than one record in the table with the same value for the primary key. No two employees can have the same ID number.

### 2.5.1.4 Foreign Key

Remember that every table in ORACLE has a primary key a field or fields making each record unique. In the employee table, the primary key is the employee ID number, and it is stored in the EMPL\_ID field in the EMP\_DATA table, then EMP\_DATA is the foreign key in EMP\_DATA table.

## **Chapter 03**

### **Methodology**

As mentioned above under literature review, whole project was gone through under the waterfall method. The waterfall method can be sub categorized as follows;

1. Requirement Analysis Phase
2. System Analysis and Design Phase
  - 2.1. Data and Database Design
  - 2.2. Interfaces and Form Design
3. Coding and Implementation Phase
4. Testing Phase
  - 4.1. Unit Testing
  - 4.2. System Testing

#### **3.1. Requirement Analysis**

In the requirement analysis phase initially all the head of division, senior system analysts, engineers, chief executives and Analyst programmers were interviewed and discussed about the disadvantages of the existing system and how it can be overcome etc.

The written articles about the Archiving solution for database system in the web and the library of Codegen were helpful to gather the information.

##### **3.1.1. Functional Requirement Analysis**

Under functional requirement analysis, all are the functionalities that have to be considered, when the system is build were analyzed as follows;

- Which users should use this software?
- What are the privileges should be granted for them?
- How the user authentication should be done?
- What is the back end that should be used for this solution?
- How it should be arranged?

## 3.2. System Analysis and Designing

### 3.2.1. Data and Database design

After the requirement analysis part, aim was pointed out to the backend. At this step, Oracle was decided to create backend. So separated database Schema was created by generating with existing database tables using Case Studio database modeling tool and utility table's data was pumped to above created Schema's tables. Information Lifecycle Management (ILM) was used to implement the system using Java technologies with Generics and JIDEA 6.0 was used as java compiler. And also CASE Studio 2 was used as database modeling tool.

Figure 3.1 shows the graphical user interface of CASE Studio 2 tool.

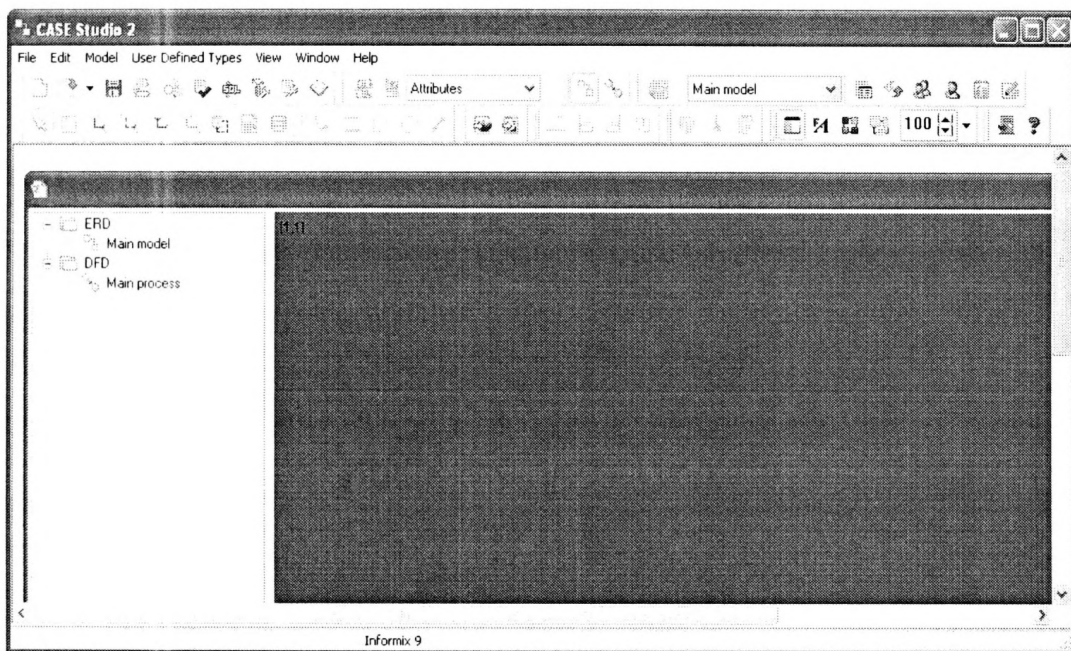


Figure 3.1 Database Modeling Tool

### 3.2.2. Interfaces and form Design

After the database design phase, my attention was involved with designing the front end of the system. For that perspective, all the forms including the Swing form were defined. Here relevant component for each forms were placed in a proper place and bound them with the database in graphical mode. Here JBuilder was used to create GUI.

And also in addition to this task my consideration was pointed to the outer appearance of those forms. Hence it was cared to put relevant colours and rendered GUI components for each forms in this system.

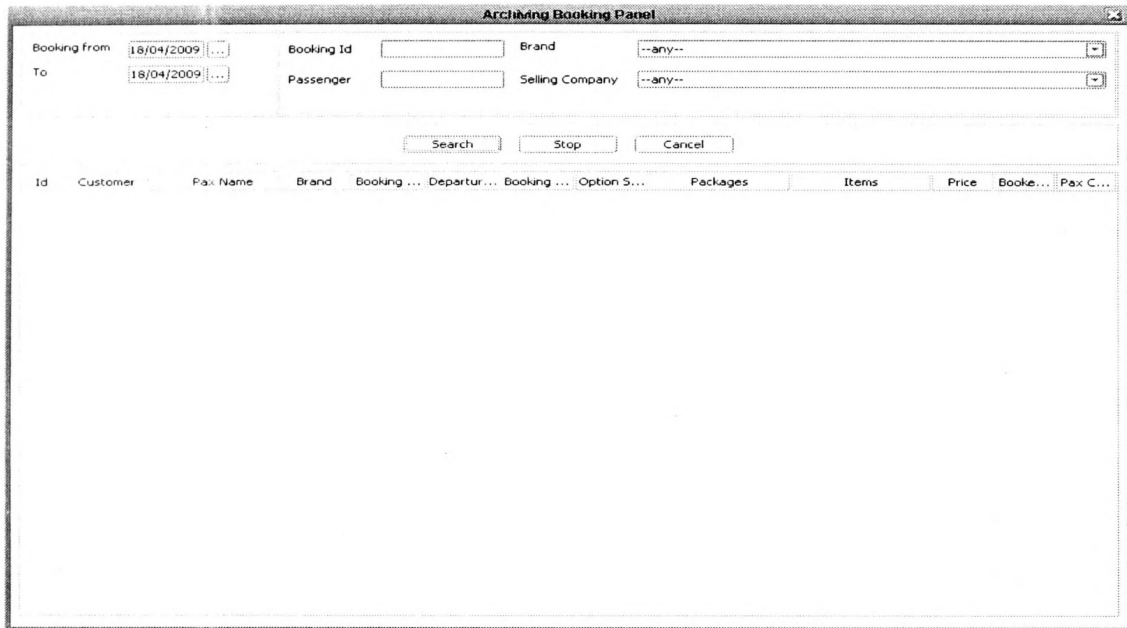


Figure 3.2 Booking Search Panel



Figure 3.4 Database Table View Panel

### 3.3. Coding and Implementation

#### 3.3.1. Coding

After the interfaces designing, the code writing was started for each event of the component in each Panel. In this part the database connection was done in runtime and girthed data from them by using object, available in the java. And also those data were placed in innovated component called 'ResultSet'. The following code was expressed in simple codes for the database connection in runtime between Java and Oracle database.

```
Statement stmt = conn.createStatement();

try {

    ResultSet rs = stmt.executeQuery("SELECT * FROM MyTable");

    try {

        while (rs.next() ) {

            int numColumns = rs.getMetaData().getColumnCount();

            for ( int i = 1 ; i <= numColumns ; i++ ) {

                // Column numbers start at 1.

                // Also there are many methods on the result set to return

                System.out.println( "COLUMN " + i + " = " + rs.getObject(i) );

            }

        }

    } finally {

        rs.close();

    }

} finally {

    stmt.close();

}
```

Here structured query language was used for data gathering, deleting and processing in the database. For an example,

```
strignContract = "SELECT * FROM INS_CONTRACT" +  
                " WHERE INS_CONTRACT.REFERENCE =? "
```

### **3.3.2. Implementation**

This whole system was run under the JAVA platform. Because of this java, platform is platform independence; this system is running under any available OS such as OS400, windows, Linux, MacOS etc.

In addition to that the back end of the system was configured under Oracle. Here this system is running with this database under the environment of java language which is a sophisticated technology introduced by the Sun Microsoft corp. Intelli J Idea 6.0 was used to implement the system.

### **3.4. Testing**

#### **3.4.1. Unit Testing**

After the coding process, this system was tested individually Panel by Panel. By searching the contract data from the Oracle database and moving to new database schema's tables, each Panel was tested with Oracle database table. To test the exception handling part those were introduced in this project. Some mistakes were made explicitly and tested whether the exception are working properly or not.

#### **3.4.2. System Testing**

After testing each Panel individually, the whole system was run and tested as a whole, by navigating between each Panel in this system.

## Chapter 04

### Results and Discussion

As discussed under the back ground session, the TravelBox system has performance problems with database growth due to infrequently access data to TravelBox's databases. Therefore software performance degradation occurs and has to spend more money for the storages. Because of these issues, massive drawbacks were found in existing TravelBox system. To overcome that issue of the TravelBox system, Archiving solution was produced to move historical data from production database to another database schema by keeping both production and historical data an online.

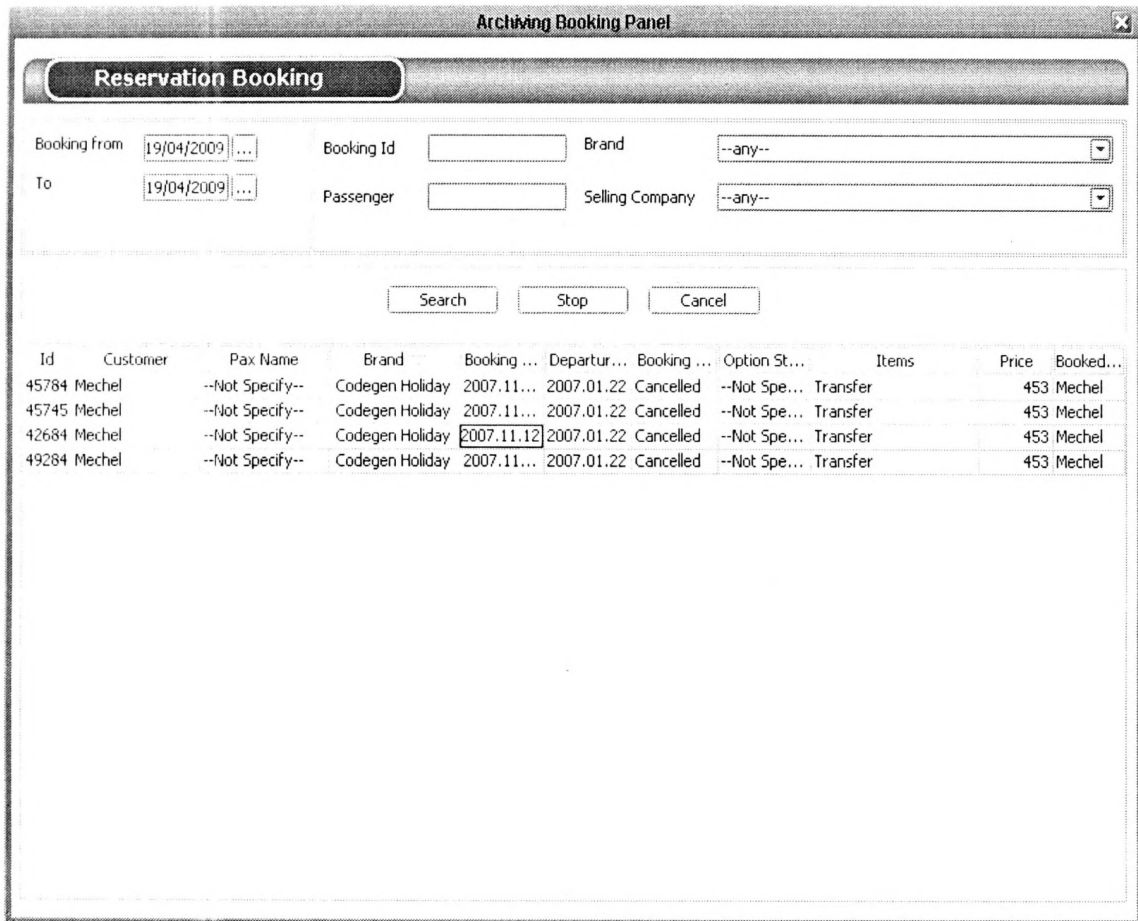
This Archiving solution system has totally been produced by using the sophisticated Java technology. Java language was used to develop this software as a programming language. Because the java technology is extremely plat form independent, this software can be guaranteed that it will work on any platform such as Windows, Linux, Solaris or OS400. And also as described under the literature review the language that has been used in this application is Java. And java is a powerful object oriented language. Because of this object oriented language that was used, all the object oriented features such as inheritance, encapsulation, and polymorphism are embedded in this application.

And as a back end, the oracle database management system was used in this application. Oracle is the most sophisticated and robust high secure database management system that is used among most of the system engineer in the world rather than traditional database management systems such as SQL server, access, MySQL.

In existing TravelBox, Flight allocation contract, Tour contract, Supplier contract and generic contract etc. are added to their databases day by day so databases capacity is increased with TB size. All historical contracts and new contracts are in the same table so that the above issues are occurred.

But according to the new system, historical contract can be moved to new database schema by using Archiving solution tool as well as if someone wants to get archiving data from new database, can be observed using application's Panel (GUI) because of having both historical and active data an online.

Here JBuilder compiler was used to design the GUI for the Archiving Tool. And all the swing components are rendered by implementing new classes. So the created swing components are very attractive. IntelliJIDEA 6.0.5 is easy to implement system than JBuilder compiler because of JBuilder compiler is more memory consuming tool and it is difficult to open 3 or 4 instance at same time.



**Figure 4.1 Reservation Booking Search panel**

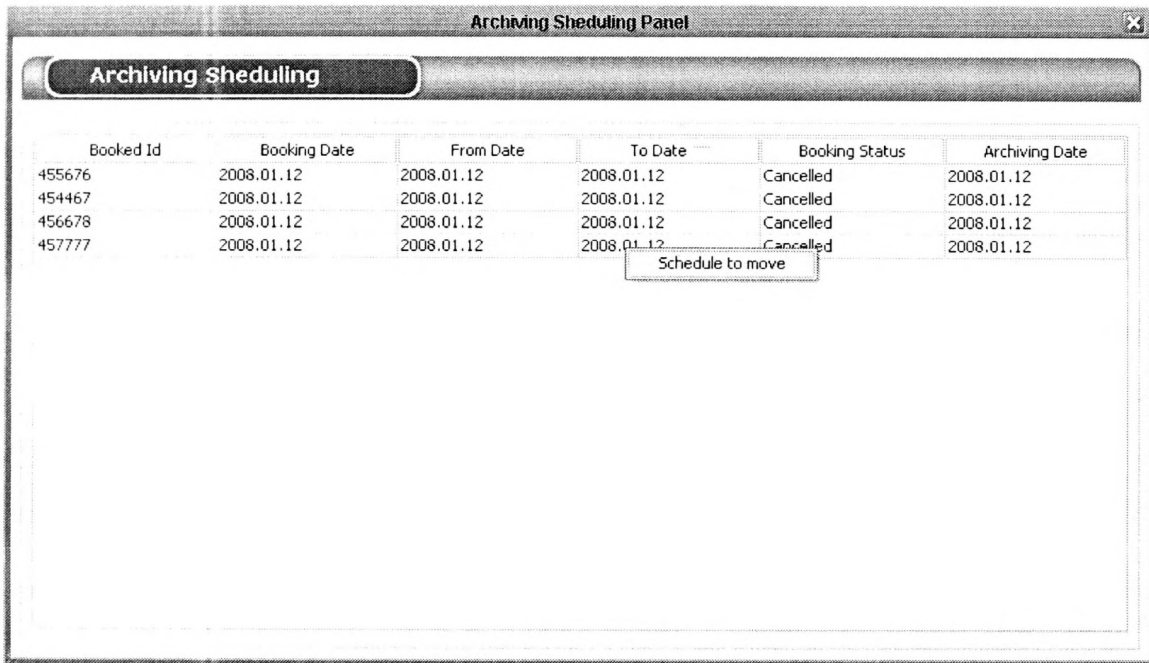
Using Reservation booking search panel, can search booking with selecting or not selecting search criteria such as booking from date, booking to date, selling company, booking product, passenger and brand etc. and also user just select text field from rendered comboBox and don't want to fill the text field given. After booking searched, a selected row object can move from Data Grid to new database schema's



tables. And also if we want to view moved data to new schema database's table, that data can be viewed by using above Reservation booking search panel.

If we want to stop while searching, stop button can be used for that. And also search criteria can be reset by using the Reset button. Whenever, someone wants to close this panel, close button can be used for this process as well.

As well, the date and time can be scheduled to move and delete the object (contract) from existing database tables by using below figure.

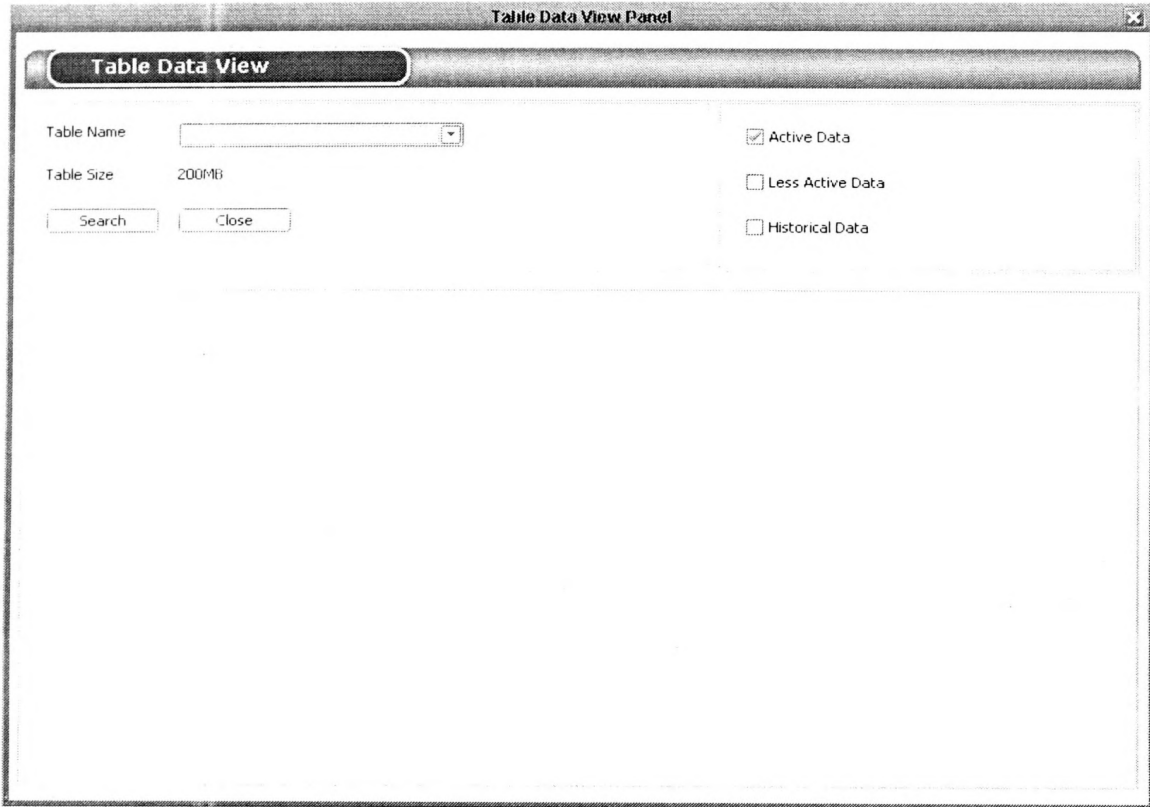


Booked Id	Booking Date	From Date	To Date	Booking Status	Archiving Date
455676	2008.01.12	2008.01.12	2008.01.12	Cancelled	2008.01.12
454467	2008.01.12	2008.01.12	2008.01.12	Cancelled	2008.01.12
456678	2008.01.12	2008.01.12	2008.01.12	Cancelled	2008.01.12
457777	2008.01.12	2008.01.12	2008.01.12	Cancelled	2008.01.12

Schedule to move

**Figure 4.2 Archiving Sheduling panel**

Using below panel table data can be viewed with selecting given search criteria.



**Figure 4.3 Table Data View panel**

This archiving solution can increase the application performance by relocating historical data to another separate database schema. As well ,both production data and historical data keep online.

## **Chapter 05**

### **Conclusion**

As mentioned above under the discussion session, the existing system has several draw backs. The major disadvantages of the existing system and how they have been overcome by Archiving solution system can be discussed as follows;

In the existing system, all active data and historical data are together in tables, so when need a record from the table, all the data in the database table is searched in this process. Thus the task is more time consuming and application performance is low. If company decided to delete these historical data from the system, company may loose their data from the system. As the size of the database is very large they have to allocate a lot of money for the storage devices. Thus the modified system in this study is very efficient and its application performance is more significant rather than the existing system.

And also in the existing system historical data has to delete manually but new system can schedule the date as it needed to delete data and it can schedule date that is needs to move to new system database. After moving data to new database that data is automatically deleted from the existing application database.

The existing system can search both historical and active data in the existing system's database. But new system can view either historical or active data as we need.

## REFERENCES

Andrew Stellman and Jennifer Greene (2005). Applied Software Project Management . Cambridge, MA: O'Reilly Media. ISBN 0596009488.

Bounie, O. and Michael, S. (1997), Oracledata Warehousing, 1<sup>st</sup> edition, Techmedia publishers, 699p.

Cars, H. and Gary, C. (2001), Core java volume 01 fundamental, 5<sup>th</sup> edition, Addison Wesley Longman(Singapore) Pvt Ltd, 736p.

Coderanch,(2009.01.09.), Advantage of java, <http://www.coderanch.com/t/406766/Java-General-beginner/java/What-Advantage-java-bean>.

Mitchell, W. and Robert, L. (1997), Object Oriented programming in Java, 1<sup>st</sup> edition, Stephen Gilgert and Bill Mccarty publishers, 953p.

Muhammad saquib,(2008.12.16),Object Oriented Programming Techniques,<http://www.hotscripts.com/Detailed/39268.html>.

Petor, N. and William, S. (1997), Guide to Java Programming, 1<sup>st</sup> edition, Techmedia publishers, 875p.

Rick, G. (1998), Orche Web Application Server 3,1<sup>st</sup> edition, Prentic Hall publishers,474p.

Sandip, J. (2008.11.25), Waterfall - Software Development Model <http://www.buzzle.com/editorials/3-13-2005-67039.asp>.

Schach, R. (2009.02.08), Software Engineering, Fourth Edition, McGraw-Hill, Boston, MA.<http://www.buzzle.com/editorials/3-13-2005-67039.asp>

Sommerville, I. (1995), The fifth edition of Software Engineering.Addison Wesley Publishers in autumn, pp.210-400.

Tim McLellan,(1994),What is an Oracle Relational Database, <http://www.islandnet.com/~tmc/html/articles/orareln.htm>.

Wieggers, Karl E. (2003). Software Requirements 2: Practical techniques for gathering and managing requirements throughout the product development cycle, 2<sup>nd</sup> ed., Redmond: Microsoft Press. ISBN 0-7356-1879-8.

Wikipedia,(2008.12.03),Object-oriented programming, [http://en.wikipedia.org/wiki/Object\\_oriented\\_programming](http://en.wikipedia.org/wiki/Object_oriented_programming)

Wikipedia,(2008.11.29), Oracledatabase, [http://en.wikipedia.org/wiki/Oracle\\_database](http://en.wikipedia.org/wiki/Oracle_database).

Wikipedia,(2009.01.23),HSM and ILM,[http://en.wikipedia.org/wiki/File\\_area\\_network#HSM\\_and\\_ILM](http://en.wikipedia.org/wiki/File_area_network#HSM_and_ILM)

**National Digitization Project**  
**National Science Foundation**

Institute : Sabaragamuwa University of Sri Lanka


1. Place of Scanning : Sabaragamuwa University of Sri Lanka, Belihuloya

2. Date Scanned : ...2017.09.22.....

3. Name of Digitizing Company : Sanje (Private) Ltd, No 435/16, Kottawa Rd,  
Hokandara North, Arangala, Hokandara

4. Scanning Officer

Name : .....G.A.G. Gandaraman.....

Signature : ..........

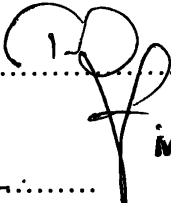
Certification of Scanning

*I hereby certify that the scanning of this document was carried out under my supervision, according to the norms and standards of digital scanning accurately, also keeping with the originality of the original document to be accepted in a court of law.*

Certifying Officer

Designation : LIBRARIAN.....

Name : T.N. NEIGHSOOREI.....

Signature : ..........

Date : .....2017.09.22.....

**Mrs. T.N. NEIGHSOOREI**  
**(MSSc.PGD.ASLA,BA)**  
**Librarian**  
**Sabaragamuwa University of Sri Lanka**  
**P.O.Box 02, Belihuloya, Sri Lanka**  
**Tel: 0094 45 2280045**  
**Fax: 0094 45 2280045**

*"This document/publication was digitized under National Digitization Project of the National Science Foundation, Sri Lanka"*