

**DEVELOPMENT OF AN ONLINE ECOLOGICAL DATA
WAREHOUSE**

CHAM HUI

**FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2016

**DEVELOPMENT OF AN ONLINE ECOLOGICAL DATA
WAREHOUSE**

CHAM HUI

**DESSERTATION SUBMITTED IN FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF MASTER
OF SCIENCE**

**FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2016

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Cham Hui

Matric No: SGR100047

Name of Degree: Master of Science

Title of Dissertation/Thesis: "Development of an Online Ecological Data Warehouse"

Field of Study: Ecological Informatics

I do solemnly and sincerely declare that:

- (1) I am the sole author/writer of this Work;
- (2) This Work is original;
- (3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
- (4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
- (5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
- (6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature

Date:

Subscribed and solemnly declared before,

Witness's Signature

Date:

Name:

Designation:

ABSTRACT

Effective and efficient maintenance and monitoring of tropical water bodies such as lakes, reservoirs and river requires well-documented, validated, and coherent data archives. Data are collected and stored individually and this makes data integration difficult even among close collaborators. Ecological data warehouses for data mining require individual databases that are precisely explained with validated metadata. A data warehouse system for tropical water bodies should have a standard interface and identical ways of representing biological data for data exchange. These can solve the problems of incomplete data, data standardization and data sharing. The data warehouse prototype developed in this study comprises of graphical user interface (GUI) developed using ASP.Net. To facilitate data integration and exchange, standard format and common data exchange method via web services are used. XML, which is known for its high degree of interoperability for data exchange and transfer, is used to represent metadata. Darwin Core is used for formats for ecological and biological data management and for data exchange protocol. The data warehouse system prototype developed in this study aims to provide information management of hydrological and ecological data on Malaysian water bodies with interactive user interface, report generation, data support for data mining tools, and data sharing between water research and management agencies.

ABSTRAK

Keberkesanan penyelenggaraan dan pemantauan badan-badan air tropika seperti tasik, kolam dan sungai memerlukan dokumentasi yang bagus, disahkan, serta arkib data yang kukuh. Data biasanya dikumpulkan dan disimpan secara berasingan, dan perkara ini menjadikan integrasi data sukar walaupun di kalangan rakan kerjasama yang rapat. Gudang data ekologi untuk perlombongan data memerlukan pangkalan data individu boleh dijelaskan dengan tepat menggunakan metadata yang disahkan. Sebuah sistem gudang data untuk badan-badan air tropika haruslah mempunyai antara muka yang berpiawai dan menggunakan cara yang sama bagi mewakili data biologi untuk sebarang aktiviti pertukaran data. Cara ini boleh menyelesaikan masalah data yang tidak lengkap, piawaian data, dan perkongsian data. Gudang data tersebut mestilah terdiri daripada antara muka pengguna grafik (GUI) yang dibangunkan menggunakan ASP.Net. Untuk memudahkan integrasi dan pertukaran data, format berpiawai dan kaedah pertukaran data yang sama melalui perkhidmatan web boleh digunakan. XML, terkenal dengan kebolehan interoperasi yang tinggi untuk pertukaran dan pemindahan data, digunakan untuk mewakili metadata. Untuk format pengurusan data ekologi, Darwin Core digunakan, dan untuk protokol pertukaran data, pengurusan data biologi digunakan. Prototaip sistem gudang data yang dibangunkan dalam kajian ini adalah bertujuan untuk menyediakan pengurusan maklumat data hidrologi dan ekologi di kawasan perairan di Malaysia dengan antara muka yang interaktif pengguna, mempunyai penjana laporan, mempunyai data sokongan untuk data alat-alat perlombongan dan menggalakkan perkongsian data antara agensi penyelidikan air dan agensi pengurusan.

ACKNOWLEDGEMENTS

I would like to use this great opportunity to give thanks to everyone who has given their support on my study and research work. Thousand thanks to my supervisor Prof. Datin Dr. Aishah Binti Salleh for consistent support, motivation, guidance, and full support on documentation and administration work. Great thanks to my second supervisor Dr. Sorayya Malek for consistent support throughout my master research. I appreciate her guidance, patient and support in my system development and thesis writing.

Special thanks to Dr Pozi Milow for support on my journal writing and knowledge sharing.

Last but not least, I would like to share the achievement of this work of mine with my friends and family especially my parents and Kong Po Lian, and also my brothers Lau Chia Fong, Tang Chee Kuang and Oh Jin Heng. I wouldn't have completed this research without their understanding, help and support.

TABLE OF CONTENTS

Abstract	iii
Abstrak	iv
Acknowledgements	v
Table of Contents	vi
List of Figures	viii
List of Tables.....	x
List of Symbols and Abbreviations.....	xi
List of Appendices	xii
CHAPTER 1: INTRODUCTION AND OBJECTIVES.....	13
1.1 Introduction.....	13
1.2 Objective.....	16
CHAPTER 2: LITERATURE REVIEW.....	17
2.1 Existing ecological data warehouse systems	18
2.2 Data Standard and Web Services.....	25
2.3 Data Mining and GIS	29
CHAPTER 3: METHODOLOGY.....	32
3.1 Study Area	32
3.2 System Architecture.....	37
3.3 System Development Methodology	38
CHAPTER 4: RESULTS AND DISCUSSION	48
4.1 Result and Discussion.....	48

4.1.1	System Input.....	48
4.2	System output	55
4.3	Discussion.....	64
 CHAPTER 5: CONCLUSION.....		69
	References	70
	List of Publications and Papers Presented	75
	Appendix	78

University of Malaya

LIST OF FIGURES

Figure 2.1 World Lake Database information on Lake Chini, Malaysia	19
Figure 2.2 LakeNet information on Lake Chini , Malaysia	20
Figure 2.3 Nahrin website information on Lake Kenyir, Malaysia	21
Figure 3.1 Putrajaya Lake Map	33
Figure 3.2 Lakes and Reservoir of Malaysia	35
Figure 3.3 Star schema of the system.....	36
Figure 3.4 System architecture.....	38
Figure 3.5 Top level view of system modules	39
Figure 3.6 Site and User Module	40
Figure 3.7 Hydro-chemical Module.....	41
Figure 3.8 Biological Module	42
Figure 3.9 Data Mining Module.....	43
Figure 3.10 Web Services Module.....	44
Figure 3.11 Partial Entity Relationship Diagram of the system	46
Figure 4.1 Input interface for water bodies using RAMSAR standard.....	49
Figure 4.2 User Registration	49
Figure 4.3 Configure sampling station.....	50
Figure 4.4 Variable configuration	52
Figure 4.5 Data entry for Hydrological and Chemical Data	53
Figure 4.6 Variable Setting for Biological Data	54
Figure 4.7 Data Entry for Biological Data	54
Figure 4.8 Ramsar report layout a) general site geographic information b) Uses and Threats c) Information and Conservational Issues d) Ecological Data.....	56
Figure 4.9 Hydrological Data Monthly Report	57

Figure 4.10 Multiple Station vs Time series monthly rainfall report.....	58
Figure 4.11 Exported Excel Report for Multiple Station VS Time Rainfall Monthly Report.....	59
Figure 4.12 Multi Hydrological Data VS Time Daily Report	60
Figure 4.13 Matrix Report.....	61
Figure 4.14 WQI Report	62
Figure 4.15 Chlorophylla level visualization on Putrajaya Lake map using ArcMap demonstrating chlorophylla level	63
Figure 4.16 SOAP message for request and response within prototype and other parties	64
Figure 4.17 XML file formatted based on Darwin Core format.	64

LIST OF TABLES

Table 1.1 Selected examples of databases for water bodies and wetlands	14
Table 2.1 Summary of Characteristics of Ecological Data Warehouse System	25
Table 3.1 General Details of Putrajaya Lake	34
Table 4.1 Land Types of Station	51

University of Malaya

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	:	Artificial Neural Network
EA	:	Evolutionary Algorithms
GBIF	:	Global Biodiversity Information Facility
GIS	:	Geographical Information System
HEA	:	Hybrid Evolutionary Algorithm
SSADM	:	Structured Systems Analysis and Design Method
SOAP	:	Simple Object Access Protocol
SOM	:	Self Organizing Feature Map
WSDL	:	Web Services Description Language
WQI	:	Water Quality Index
XML	:	Extensible Markup Language

LIST OF APPENDICES

Appendix A: Information Sheets of Ramsar Wetlands	78
.....	
Appendix B: Appendix B: Stored Procedure for Star Schema	89
.....	

University of Malaya

CHAPTER 1: INTRODUCTION AND OBJECTIVES

1.1 Introduction

The development of ecological dataset requires expertise in database programming knowledge which may not be possessed by biologists or ecologists (Cushing et al., 2007). Researchers still store and manage their data in spreadsheets even though database system is more convenient nowadays. Spreadsheets have shortcomings such as limited recording and storage capacity, and also difficulty in querying out required data. Database system allows researchers to extract and query data for analysis and visualization. Many researchers have already expressed the need for the incorporation of database technology in their research but very few of them are trained in database design. One of the major difficulties that researchers had with data management was integrating datasets in space and time for exploration and analysis. For this reason, many opportunities for knowledge discovery were missed. Furthermore the ecological databases are structured in heterogeneous formats and using different platforms. It makes data sharing a difficult task due to incompatibility issues. These have contributed to the poor dissemination of water research findings and lack of coordination between water management agencies in Malaysia.

In Malaysia water quality of lakes and wetlands are declining due to increasing pollution (Abdullah, 2015). According to Jusoh (2009), 38% of Malaysian lakes are mesotrophic and 62% are eutrophic. Efforts to solve this problem are difficult partly because water quality databases on lakes are scattered and the data are not centralized (Sarip and Zakaria, 2008). Some of the databases are maintained by independent bodies and organizations that are not involved policy-making, enforcement, regulations, and legislation on matters pertaining to lakes management. The data collected from different lakes is not standardized and shared between lake management agencies. Worst still, some of them do not have databases to store data and information for research purposes.

All these make database integration and data sharing of lakes difficult to achieve (Abdullah, 2015).

International lake databases such as World Lake Database (ILEC, 2013), Ramsar Site Database (Ramsar, 2013), and GIS WORLDLAKE database (Kondratyev and Filatov, 1999) also store data on Malaysian wetlands and lakes. However, exchange of information between the international and local water management agencies is still lacking. Among the reasons are unstandardized format of data storage, lack of data updates, and incomplete data. For example, World Lake Database developed by International Lake Environment Committee Foundation is established in 1986 stores environmental and socio-economic data of lakes and reservoirs around the world. However, its data is not standardized partly to enable uploading of non-specific lake information by its users. Its data on a Malaysian lake (Lake Chini) is not updated (ILEC, 2013). Other databases (both local and international) on water bodies are briefly described in Table 1.

Table 1.1 Selected examples of databases for water bodies and wetlands

Databases	Main roles	Comments	Reference(s)
1. LakeNet	Dedicated to the conservation and sustainable development of lake ecosystems around the world; strengthening lake organizations, educating and inspiring people on stewardship of lake ecosystems	Database contains only fundamental data of lakes; contains incomplete information on Malaysian lakes	Ryanzhin et al. (2001)
2. Ramsar Convention	Conservation of and sustainable use of wetlands	Does not store daily data; data on Tasik Bera (Malaysia) was last updated on July 1998	Kondratyev and Filatov (1999)
3. Global Database	Storing data on natural lakes,	Data on geography, chemistry,	Kondratyev and Filatov (1999)

and GIS Worldlakes	reservoirs, and wetlands	hydrology, and others; only brief a brief and out-of-date data for lakes in Malaysia	
4. National Lake Information Database of Malaysia (NAHRIM)	National information repository for all lakes in Malaysia, whether natural or man-made; the database provides relevant information to support the effective and sustainable management of all lakes in Malaysia	Does not enable information sharing and exchange and does not cover all the data of available lakes in Malaysia	NAHRIM (2013)
5. Putrajaya Lake and Wetland Management and Operational System (PLWMOS)	It serves as the main environmental database and spatial analytical tool for the current survey of hydrology, physico- chemical, and biotics parameters of Putrajaya lake and wetlands; also designed as a knowledge-based decision support system particularly to serve as an early warning system to alert for potential environmental problems	Data and information of the system are not for public sharing and it caters only for Putrajaya Lake	PLWMOS (2013)

A standard interface and identical ways of representing biological data for data exchange can help to overcome issues such as incomplete data and problems related to data standardization and sharing. Many types of protocol were used by investigators to share complex ecological data within community or globally. Therefore heterogeneous data should be stored in autonomous databases within the ecological research community (Berkley et al., 2001). XML are used to represent metadata in most of the current ecosystem informatics research and known for its high degree of interoperability for data

exchange and transfer among distributed research group (Seligman and Rosenthal, 2001). Darwin Core has been widely used for standardizing and exchanging ecological and biological data. It is an XML based standard which consists of basic context like properties, elements, fields, attributes, and concepts. Darwin Core was created to develop data models and manage information of biodiversity informatics in Global Biodiversity Information Facility (GBIF, 2013).

Web service is helpful in managing biogeographic archive and analyzing habitat (Best et al., 2007). Web services provide the ability to exchange messages between computers over the internet, therefore allowing remotely distributed computers to share data and computing resources easily. This is accomplished through a set of standard protocols that facilitate how a server documents its available services, how a client requests the server to perform a process, and how services are discovered on the web (Curbera et al., 2002).

The aim of this paper is to propose a data warehouse system prototype for tropical water bodies using Malaysian tropical water bodies' dataset. The data warehouse will be used for data archiving and retrieval to facilitate seamless data transfer using web services between researchers that are within or outside of the database platform. The framework will also help to overcome problems that prevent data sharing such as incomplete and unstandardized data.

1.2 Objective

- i) Design and implement an ecological data warehouse using standard data protocols such as Darwin Core prototype.
- ii) Develop and implement SOAP protocol for data archiving and retrieval to allow data sharing between researchers despite of system platform used.
- iii) Visualize and generate statistical report for water quality management.

CHAPTER 2: LITERATURE REVIEW

Water body is significant accumulation of water on earth surface. The accumulations of water can include oceans, seas, lakes, ponds, wetlands and others. It can be still or contained. Rivers, streams and canals are the examples of natural water. There are also man made artificial water bodies for example reservoirs and wetlands that can be considered as a water management tool.

Malaysia uses these water bodies as water supply for domestic, agricultural, and industrial purposes commonly. Some reservoirs and lakes play a role of flood control detention storage to buffer different flow during wet and dry season. Water bodies are also created when silt retention or detention basins are constructed in city areas. It also supports important ecosystem and repository of rare species. Some of the Lakes are designated for tourism and water based recreational sites for sports, commercial or fishing activities.

However the water bodies all over the world are facing few common problems which are eutrophication, sedimentation and weed infestation. Eutrophication is known as a critical issue in Malaysia. Eutrophication is the result of water bodies enrichment, is increased growth of microscopic floating plants, algae and the formation of dense mats of floating plants. Eutrophication can bring the effects of anoxia which kills fish and invertebrates and result to release of unpleasant and injurious gases. Algae will blooms and other aquatic plants growth uncontrolled. Species and diversity of plants and animals decreased in number. The fishing, angling and recreational activities will be restricted due to the plant accumulation. The solution to this problem is by having effective and efficient maintenance and monitoring of water bodies that are well-documented, validated, and coherent data archives. Ecology data warehouses for data mining are not precisely explained with validated metadata. It is important for a data warehouse system

for water bodies to have a standard interface and identical ways of representing biological data for data exchange. These can solve the problems of incomplete data, data standardization and data sharing. The segment below addresses some of the common ecological data warehouses.

2.1 Existing ecological data warehouse systems

One of the key assumptions underlying data warehouse is that individual components can be extended and combined in various ways. It means that template databases offer greater flexibility than monolithic data models. Furthermore, ecological data are inherently spatial and might involve making observations about structural elements, which are less likely to diverge over time or between different studies, and which also can be used as join points. If databases are built by common and interchangeable representations of spatial data and coherent conceptualization of ecological structural elements, these databases can be more easily managed, provide metadata, allow the development of tools which can be used on many databases, and provide common variables over which some dissimilar data sets can be joined. Functional data can be inserted in an ad hoc manner, and not affect the performance of structure-based templates to provide the obvious benefits of common components. (Vos et al, 2000)

World Lake Database is developed by International Lake Environment Committee Foundation. The database is established in 1986 stores environmental and socio-economic data of lakes and reservoirs around the world. However data on Malaysian lakes are incomplete as shown in Figure 2.1 on Lake Chini. The data is not standardized partly to enable uploading of non-specific lake information by its users.



World Lake Database

International Lake Environment Committee Foundation (ILEC)

[Character size](#) ▶ [Scale up](#) ▶ [Scale down](#) ▶ [Undo](#)

[▶ Contacts](#)

[Search Lakes by Name](#) | [Search Lakes by Country](#) | [Search Data by Free Word](#) | [Search by Free Word](#) | [Home](#) |

[Home](#) > [Search Lakes by Name](#) > [List of Lakes](#) >

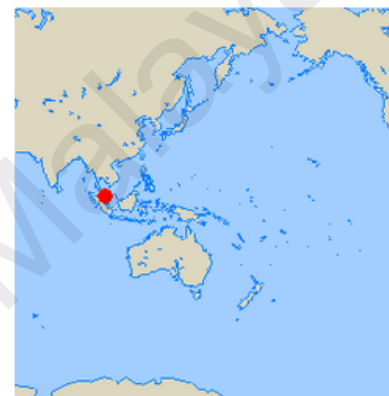
Lake Chini

SASI-043

Surface Area	2 km ²
Volume	
Maximum Depth	
Mean Depth	
Shoreline Length	

Latitude	3°25' N
Longitude	102°56' E
Altitude	0 m
Country	Malaysia
Catchment Area	
Residence Time	
Freezing Period	None
Mixing Type	

[Preview This Lake Data List \(7\)](#)



Location of the Lake

Figure 2.1 World Lake Database information on Lake Chini, Malaysia

LakeNet (Figure 2.2) is another global database system is maintained by people or organizations dedicated to the conservation and sustainable development of lake ecosystems around the world. The LakeNet support information services and exchanges in their website. Their main focus is on strengthening lake organizations, educating and inspiring people on stewardship of lake ecosystems. They only have the fundamental data of lakes and some of the details are incomplete or blank. (LakeNet, 2012)

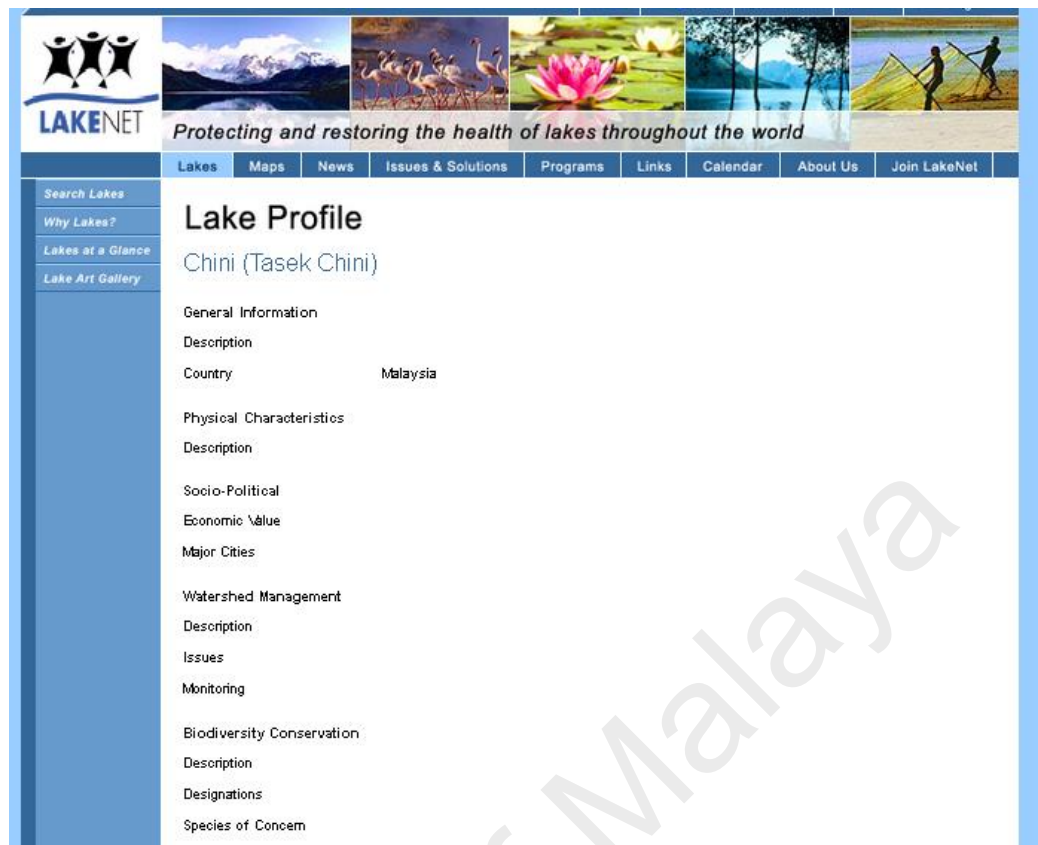


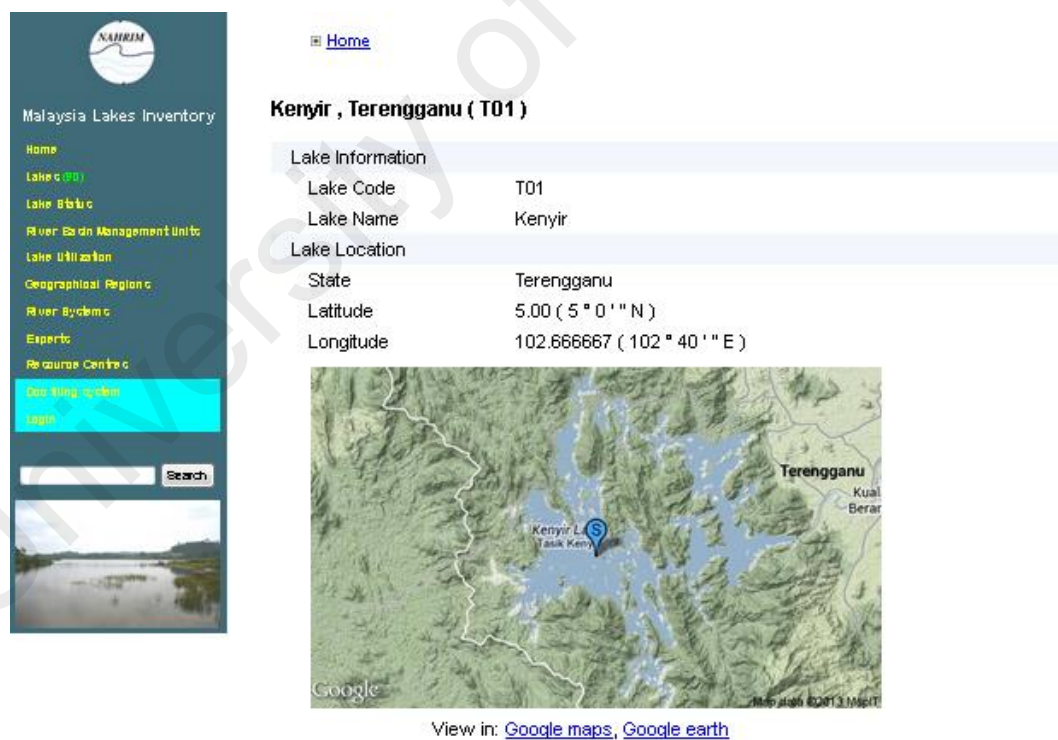
Figure 2.2 LakeNet information on Lake Chini , Malaysia

Ramsar Convention is an intergovernmental treaty on the conservation and sustainable use of natural resources. The Ramsar sites are selected by contracting parties, member states, and designation refer to criteria for Identifying Wetlands of International Importance. The sites join which want to join Ramsar are required to complete the Ramsar Information Sheet. The Ramsar Information Sheet includes accurate data, conservation parameters, and map precisely delimiting boundaries of the site. These data does not include the data which required daily update like physical data, chemico data, or biodiversity. The data of lakes are not up-to –date and it is not updated for a few years.(Ramsar Convention.2013)(Rouchelle, 2013)

The Global Database and GIS Worldlakes stored 35000 natural lakes, 5000 reservoirs, and 220 wetlands data. These data attributes includes geography, chemistry, hydrology,

and others. The data for Lakes in Malaysia are lesser and not up-to-date (Ryanzhin et al 2001).

National Lake Information Database of Malaysia (Figure 2.3) has been developed by the National Hydraulic Research Institute of Malaysia as the national information repository for all lakes in Malaysia, whether natural or man-made. The objective of this Database is to provide relevant information to support the effective and sustainable management of all lakes in Malaysia. NAHRIM welcomes all those who have information on any lakes in Malaysia to register as "Contributors" to the Database. The information for each lake in the Database are organized under 3 themes. They are (a) Summary information of the lake, (b) Assessment information of the lake, (c) Lake water quality monitoring information



The screenshot displays the NAHRIM website interface. On the left is a dark sidebar with the NAHRIM logo and a list of navigation links: Home, Lake Code, Lake Status, River Basin Management Units, Lake Utilization, Geographical Regions, River Systems, Experts, Resource Centre, and Data Management System. Below the links is a search bar and a small image of a lake. The main content area is titled 'Kenyer , Terengganu (T01)' and contains a table of lake information. Below the table is a map of Terengganu, Malaysia, with a blue pin marking the location of Lake Kenyir. The map is credited to Google and includes a 'View in: Google maps, Google earth' link.

Lake Information	
Lake Code	T01
Lake Name	Kenyir
Lake Location	
State	Terengganu
Latitude	5.00 (5 ° 0 ' " N)
Longitude	102.666667 (102 ° 40 ' " E)

Figure 2.3 Nahrin website information on Lake Kenyir, Malaysia

The PLWMOS system is intended to serve as the main environmental database and spatial analytical tool for the current survey of hydrology, physico-chemical, and biotics

parameters of Putrajaya Lake and wetlands. The intended PLWMOS system will have functionalities for data entry, environmental modeling and online report submission and shall have the capability to be linked to the existing system (or systems) within CMC (e.g. EMMS). PLWMOS will also be designed as a knowledge-based decision support system particularly to serve as an early warning system to alert for potential environmental problems. It stored hydrology, physic-chemical, and biotic parameters. The data and information of the system are not shared with public and it only stores 1 data on one lake (Putrajaya, 2013).

Matthews natural wetlands database contains a series of files which represents the global coverage of wetlands. These files were developed by compiling vegetation, soil and inundation maps to show the environmental and distribution characteristics of naturally occurring wetlands. About one half of the total wetland area lies between 50 and 70°N in the Matthews database. This high-latitude belt is characterized by peat -rich ecosystems such as bogs and fens. Estimate 35% of the global wetland area is broadly distributed in the latitude zone extending from 20°N to 30°S. This belt is covered by forested and non-forested swamps and marshes, with a smaller contribution from alluvial or floodplain formations (NDSL, 2012).

The ISLSCP database: The ISLSCP (International Satellite Land Surface Climatology Project) database is derived from hydrological maps. The Cogley dataset provides global coverage of different hydrological terrains and classifying wetlands into swamps, marshes, salt marshes, salt flats, and other wetlands. The wetland area identified by ISLSCP is fairly consistently distributed over the continents, with a higher awareness in Europe and Asia. (HALL, 1995)

DISCover database: IGBP/DIS (International Geosphere–Biosphere Programme/Data Information System) has evaluated AVHRR (Advanced Very High Resolution

Radiometer) data to compile a database contains the data of global land cover. DIScover is a purely remote sensing database, whereas the other databases were derived from maps as primary data sources. Wetlands are determined as pixels with herbaceous or woody vegetation and a permanent mixture of water. Based on the study, seasonal wetlands are not represented in DIScover. DIScover database results in smaller wetland areas than data of Matthews and ISLSCP, but it classified more coastal pixels as wetlands than does Matthews or ISLSCP.

The U.S. Geological Survey (USGS) began its National Water-Quality Assessment (NAWQA) Program at 1991. It is a system collecting chemical, biological, and physical water quality data from study units across the United States. In 1999, the NAWQA Program developed a data warehouse to improve on facilitate national and regional analysis of data. The NAWQA data warehouse home page contain a variety of options to retrieve data collected at NAWQA Program sites, generating location maps and summary graphics, information about program design and background, guidance pages, and tutorials for navigating through the data warehouse pages and using the system. Information of the data warehouse is accessed using a series of selection windows. Results of data retrievals can be displayed in a different of table formats depending on the needs and criteria of the user. Data tables can be exported and saved locally in a variety of popular formats, including Excel, tab-delimited, and HTML. (Gurtz, 1994)

Global Lakes and Wetlands Database GLWD is drawing upon a variety of existing maps, information and data. It is a combination of available sources for lakes and wetlands on a global scale (1:1 to 1:3 million resolutions. The application of GIS functionality allow the generation of a database which focuses on large lakes and reservoirs, smaller water bodies, and wetlands. In a validation against documented data GLWD proved to represent a comprehensive global lakes database and give a good

presentation of the maximum global wetland extent. GLWD constitute 2 global polygon maps to which existing lake registers compilations or remote sensing data which linked in order to allow analyses in a GIS environment. It also serves as an estimate of wetland extents for global climatology and hydrology models, and identify large scale wetland distributions and wetland complexes (Lehner, B.& Doll, 2004).

Dataset of Large Reservoir Systems of the World analysis demonstrated the scope and potential impact of reservoir construction on the world river systems. Water storage behind the global population of large dams increased 700% in the standing stock of natural river water and residence times for individual impoundments spanning less than one day to few years. The emergence of GIS based datasets which can characterize the global system of rivers and drainage basins at increasingly finer resolution lends hope that the role of reservoir systems in the hydrological cycle can be more fully articulated. This study also finds that aging can lead to significant changes in the flow regime, water balance, reoxygenation of surface waters, and sediment transport. The pandemic construction of large reservoirs shows an important component of the terrestrial water cycle and merits due consideration in the future of global change studies (Sudip, 2003).

Summary and characteristics of all existing ecological data warehouse discussed in this section are summarized below.

Table 2.1 Summary of Characteristics of Ecological Data Warehouse System

No	Name	Characteristics	Comments
1	MSSL Global Lakes Database - 1 MGLD	Global. The point coordinates derived from satellite and 1:1 million Operational Navigation Charts. It comprises 1409 natural lakes and reservoirs which has attribute include type, name, and area of lakes.	It identifying closed and open lakes. Other attributes are derived from ONC and atlases. Some errors are reported Fundamental data of lakes only and some provided data are incomplete and blank.
2	World Lake Databases- ILEC	Global. No geo-referencing information(location only indicated by name of river and nearest city.)	
3	Global Databases and GIS WORLDLAKES	Global Point coordinates. 35000 natural lakes, 5000 reservoirs, and 220 wetlands. The stored attributes included geography,hydrology and others.	Extensive database and detailed lake characterizations of limnologically studied lakes.Some of the data are not available. Data for lakes in Malaysia are less and not updated.
4	Ramsar Database- RDB wetlands International	Global Representative point coordinates. It includes the site nam, area, designation date and wetland charateristics.	The database mainly contains basic information of wetlands only and the data updated continually.
5	USGS Global land cover characteristics database- GLCC	Global raster map. Available in various resolutions and classifications.	The GLCC database derived from Advanced Very High Resolution Radiometry(AVHRR) is frequently applied and considered among the most convincing land cover products as a global 1-km spatial resolution
6	LakeNet	Global. Focus on strenghtening lake organizations, educating and inspiring people on lake ecosystem.	fundamental data of lakes and incomplete details of some lakes. Provide chemical and biological data.
7	National Lake Information Database of Malaysia	National information repository for all lakes in Malaysia developed by NAHRIM	Some of the data only can access by members.
8	PLWMOS	Survey hydrology, physico-chemicaland biological data for Putrajaya Lakes and Wetlands	It only recorded details data for Putrajaya Lake. Data and information are not sharing to public.
9	Matthews Natural Wetland Database	Files present global coverage of wetlands. Combining vegetation, soil, and inundations	
10	Datasets of Large Reservoir:of dammed river	Comprises 713 large reservoirs with storage capacities. It stored the attributes like dam name, dam height, storage capacity, and the name	Dam names can be different to reservoir and lakes names as provided in other sources.

2.2 Data Standard and Web Services

Researchers and water resource managers needs to access a large and increasing quantity of data for visualization, analysis, and modeling of the water environment. However, it is difficult to locate the most appropriate resource for a particular study quickly. In addition, once the most appropriate data source has been identified, a considerable amount of effort is still required to reformat the data for analysis and visualization, or modeling. As a result, researchers and users spend a lot of time on basic

data gathering and transformations, instead of scientific analysis and decision making. (Best et al, 2007)

This problem mostly occurs when one requires data collected by multiple individuals or agencies for a particular analysis task. Atmospheric science communities, however, have different data models and data formats than those commonly used in the hydrologic sciences. If a researcher wants to make use of weather data in an analysis, it often first requires that he or she learn the file format or visualization tool used by that community. Therefore, the interoperability of data between scientific sub-disciplines, although necessary in terms of application, remains bulky to implement. While it is difficult to estimate the exact cost that results from a lack of interoperability, it would be significantly reduced if hydrologic data were communicated between systems using a standard and machine accessible protocol instead of heterogeneous web pages. (Stein, 2002). Web services provide the ability to exchange messages between computers over the Internet, therefore allowing geographically distributed computers to share data and computing resources easily. This is accomplished through a set of standard protocols that facilitate how a server documents its available services, how a client requests the server to perform a process, and how services are discovered on the web. The standardization of web service protocols is impacting a wide area of fields from business to science (JonathanL, 2008).

A standard interface and identical ways of representing biological data for data exchange can help to overcome issues such as incomplete data and problems related to data standardization and sharing. Many types of protocol were used by investigators to share complex ecological data within community or globally. Therefore heterogeneous data should be stored in autonomous databases within the ecological research community (Berkley et al., 2001). XML are used to represent metadata in most of the current

ecosystem informatics research and known for its high degree of interoperability for data exchange and transfer among distributed research group (Seligman and Rosenthal, 2001). Darwin Core has been widely used for standardizing and exchanging ecological and biological data. It is an XML based standard which consists of basic context like properties, elements, fields, attributes, and concepts. Darwin Core was created to develop data models and manage information of biodiversity informatics in Global Biodiversity Information Facility (GBIF, 2013).

Web service is helpful in managing biogeographic archive and analyzing habitat (Best et al., 2007). Web services provide the ability to exchange messages between computers over the internet, therefore allowing remotely distributed computers to share data and computing resources easily. This is accomplished through a set of standard protocols that facilitate how a server documents its available services, how a client requests the server to perform a process, and how services are discovered on the web (Curbera et al., 2002).

Web services use extensible Markup Language (XML) as a common language for communicating between systems. While XML schemas have been proposed for basic communication via web services, individual parties must supplement these generic protocols for particular domains. XML schemas have been proposed for describing data in many other geosciences communities (Goodall et al., 2008). Web services have attracted attention in the geosciences literature. They are being used as the foundation for next generation environmental models (Mineter et al., 2003), for communicating data within remote sensor networks (Liang et al., 2005), and for building virtual databases where the data are distributed across multiple machines or platform (Frehner and Brandli, 2006). Web services have been particularly popular in bioinformatics as a tool for exchanging genomic data (Pillai et al., 2005; Stein 2002; Sugawara and Miyazaki, 2003).

Apart from its use in the geosciences and bioinformatics, web services application in hydrology have been limited to date.

Web services may be used to keep a local achieve up-to-date in real time, while in other cases, the web services may also be used for direct data access within an analysis, report generating, or modeling routine. Web Service built on industry standards like SOAP and WSDL which can be used by any software systems able to consume web services. Web services become an attractive option for distributed scientific computing. It is already a critical need for hydrologic sciences because the increased availability of data and the time required to access and integrate data from heterogeneous sources. If researcher and users can use a standard protocol to implement the data sharing with web services, it will increase the interoperability of data and make scientists utilize the datasets in scientific research or studies more easily.

By using the web services, researchers and end users allow to access the Global Biodiversity Information Facility (GBIF, 2013). The Global Biodiversity Information Facility (GBIF) has developed a worldwide information infrastructure through which natural history collections which can publish their databases, and become part of a large distributed global network of shared biodiversity data (Edwards, 2004; Lane, 2006). All the internet users can access a vast global biodiversity data service and use the query out the desired data services at the same time. As of April 2007, the GBIF data portal has access to 120 million species-occurrence records from over 1000 collections housed in 200 institutions in 34 countries. All data adhere to a common set of standards protocol for data and metadata (Graham et al., 2005) and use the same methods for data exchanging over the Internet (Stein and Wieczorek, 2004), GBIF search results are returned to the user in a common xml format and present to user based own their own system and settings.

Web services provides standard format and common data exchange method (Stein, 2002). The European Bioinformatics Institute (EBI) also uses web services as their interface to allow researchers to access their database. It provides interoperability and integration within applications which require only lightweight processing program to communicate with EBI servers (Labarga et al., 2007).

2.3 Data Mining and GIS

There is an increasing demand for methods and tools in the area of ecological data analysis based on novel approaches from machine learning and information theory that would match classical statistical methods. It would increase the number of tasks significantly that can be addressed with data analysis and improve quality analysis results. Data mining uses machine learning methods which can utilize approaches from classical statistics as well as information theory. Machine learning tools have been successfully used for data and statistical analysis and learning of qualitative and quantitative models from the provided data. Decisions trees, decision roles and equation discovery are the most frequently used for the analysis for ecological data. Due to their structural properties, models induced by machine learning methods are easy to interpret and can be used to forecast values of variable which can be simple or structured (Santos et al, 2006).

The output of a data mining algorithm basically is a pattern that is valid in the provided data. A pattern is defined as a statement in a given language which describes relationships among the facts in a subset of the given data and is simpler than the enumeration of all facts in the subset. A data mining algorithm will typically have a built-in class of patterns that it considered and the particular patterns considered will depend on the given attribute and value or data. Most of the data mining algorithms come from the fields of machine learning and statistics. A common view in machine learning is machine learning algorithms perform a heuristic search through a space of hypotheses which explain valid

the data at hand. Similarly, we can view data mining algorithms as searching, heuristically or exhaustively, a space of patterns in order to find similar patterns that are valid in the given data (Marko et al, 2012).

Population dynamics studies the behavior of a community of living organisms over time, usually taking into account biotic factors and other living communities in the environment. For example, the studies of phytoplankton populations in a given lake will relate to water temperature, concentrations of nutrients pollutants and the biomass of zooplankton. The modeling formalism usually used by ecological researcher is the formalism of differential equations, which can used to describe the change of state of a dynamic system over time. A typical method of modeling population researchers writes a set of differential equations that capture the most important relationships in the domain. These are often linear differential equations. The coefficients of these equations are then determined by using measured data.

Relationships among organisms and their biotic environment normally can be highly nonlinear. Population dynamics models must have to reflect this to be realistic. The population models caused a surge of interest in the use of techniques such as neural networks for ecological modeling. Measured data are used to train a neural network which can then be used to predict future behavior and attribute of the studied organisms population.

In recent years, Evolutionary Algorithms (EA) have earned popularity for data-driven modeling due to their self-learning capacity, generality, and intrinsic parallelism and (Bäck et al., 1997), and their distinct capacity for explicit representation of make models by multivariate rules or functions. Since earlier work of EA applications for ecological modeling by Bobbin and Recknagel (2001) and Whigham and Recknagel (2001), Cao et

al. (2006) have developed the hybrid evolutionary algorithm (HEA) that is now applied for data-driven modeling of cyanobacteria blooms in lakes and rivers worldwide.

The environmental information gathered at the required spatial resolution for the entire study area can be stored in a GIS. Four main sources may be identified for the gathering of such environmental data like field surveys or observational studies, printed or digitized maps, remote sensing data, and maps obtained from GIS-based modeling.

Field data can be field measurements or a network of meteorological measurements mainly at further interpolating climatic maps. Soil units, spatial data on geology and hydrology most commonly originate from existing printed or digitized maps. Land use, snow cover, potential moisture, rocky surfaces, and vegetation maps can be retrieved from aerial photographs or satellite scenes. GIS is using overlays of environmental variables, measures of similarity measures of variation, and final rules to combine single probabilities.

CHAPTER 3: METHODOLOGY

The aim of this study is to developed prototype for data warehouse system on tropical water bodies using dataset from Malaysian tropical water bodies. Putrajaya Lake has been selected to demonstrate the implementation of the prototype in this study. The developed data warehouse prototype will be used for data archiving and retrieval to facilitate seamless data transfer between researchers that are within or outside of the database platform. The prototype will also help to overcome problems that prevent data sharing such as incomplete and unstandardized data.

3.1 Study Area

The ecological data warehouse prototype developed in this study is tested using data from Putrajaya Lake and Wetlands and NAHRIM. Putrajaya Lake and Wetlands was created by inundating the valleys of Sungai Chuau and Sungai Bisa. Putrajaya Lake and Wetlands is a man-made wetland in Malaysia and one of the largest fully constructed freshwater wetland in the tropics. The 197 hectare project resulted in changing an oil palm site into wetland ecosystem with the help of modern technology and severe environmental management methods in design and construction.

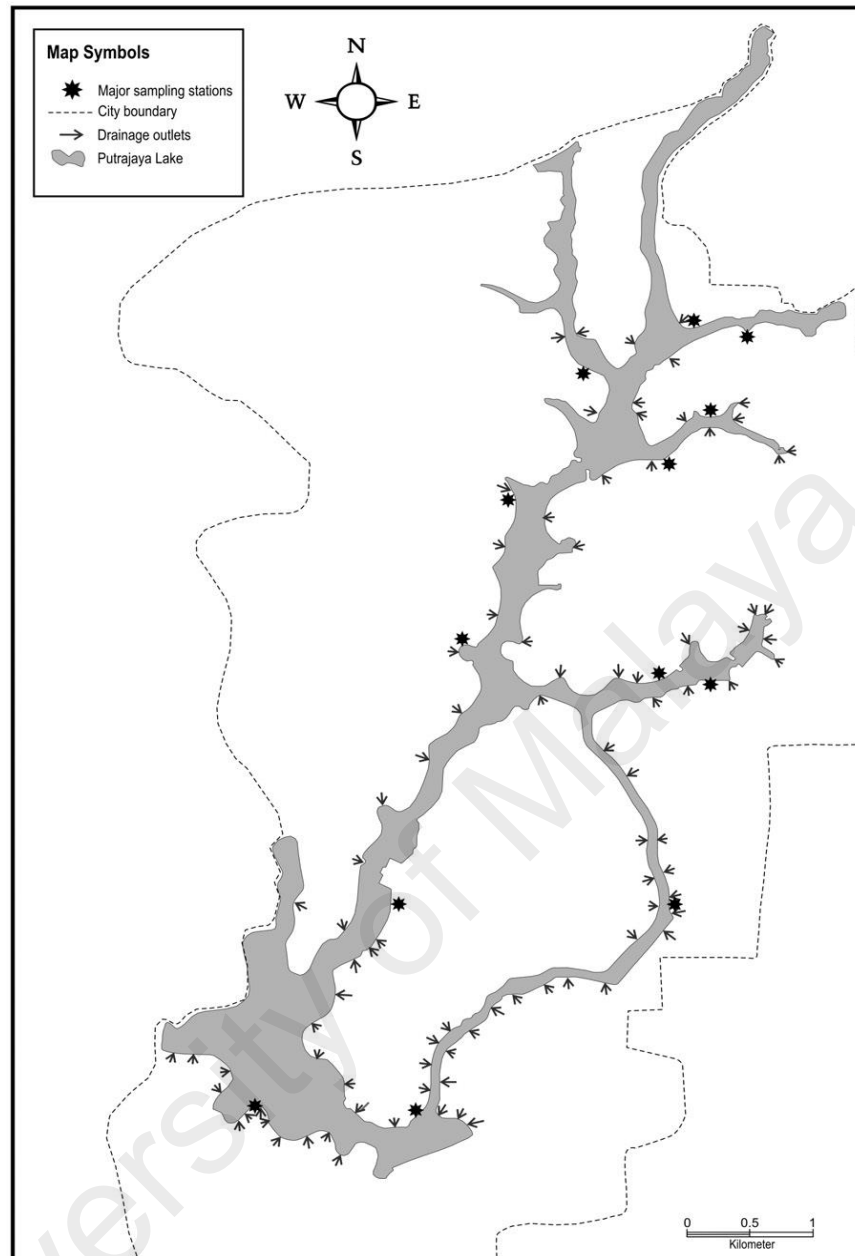


Figure 3.1 Putrajaya Lake Map

To balance the ecosystems of Putrajaya Lakes and maintain the water quality standard, over 70 species of wetlands plant in total amounts of 12.3millions plants had been planted into the area. The following figure 3.1 point out the water sampling points from Putrajaya Lakes and Wetlands. The following table depicts the general characteristics of the Putrajaya Lake.

Table 3.1 General Details of Putrajaya Lake

Putrajaya Lakes and Wetlands	
Climate	Tropical
Trophic Status	Origotrophic
Putrajaya Wetlands	
Total Areas	197.2Hectares
Planted Area	77.70Hectares
Open Waters	76.80Hectares
Weirs and Islands	9.60Hectares
Zone of Intermittent Inundation	23.70Hectares
Maintenance Tracks	9.40Hectares
Putrajaya Wetlands	
Catchment Area	50.90 KM ²
Water Level	RL 21.00M
Surface Area	400Hectares
Storage Volume	23.50Mil M ³
Average Depth	6.60M
Average Catchments Inflow	200 millions L
Average Retention Time	132days

The data set of Putrajaya Lakes included hydrological, chemical, and biological from year 2001 to 2009.

NAHRIM is a center of excellence for water and its environment, Research and Development services as well as to coordinate research activities to meet the needs of the various sector of the country. Figure 3.2 illustrates NAHRIM sampling stations in Malaysia which includes lakes and reservoirs.

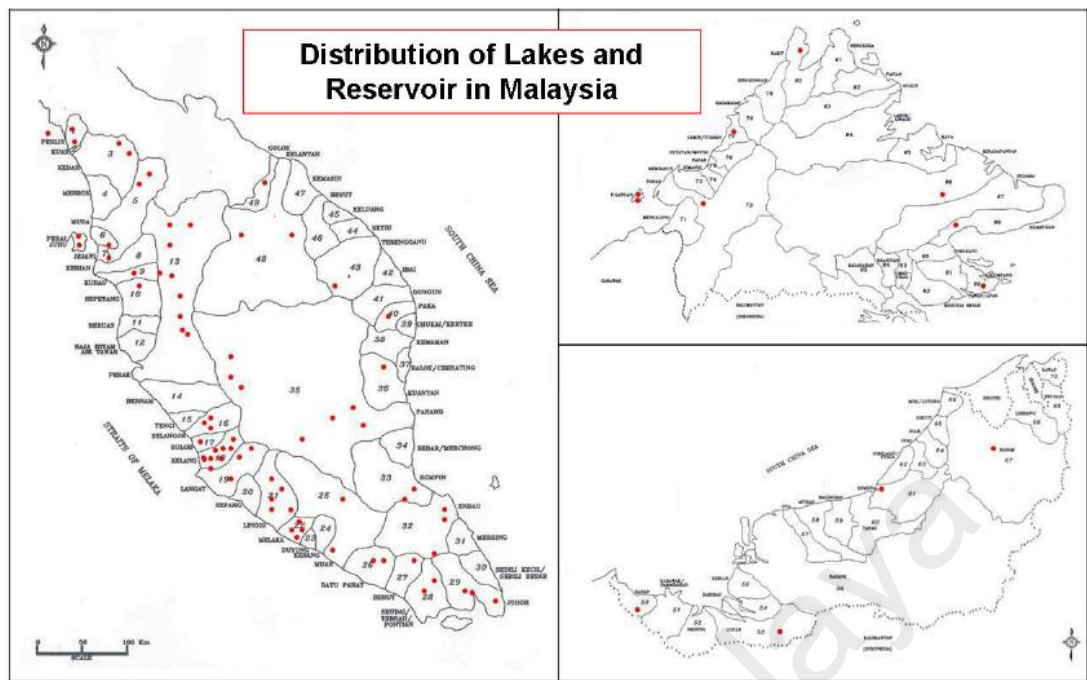


Figure 3.2 Lakes and Reservoir of Malaysia

The data warehouse prototype is developed using star schema. The star schema architecture resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables. Fact tables in a star schema are in third normal form (3NF) whereas dimensional tables are de-normalized. It is most commonly used is recommended by Oracle. Star schema important because it determine the structure of data warehouse and provides the foundation from where the data warehouse was modeled. It is important to identify measurements, events, processes, and relationships that are relevant to the user of the data warehouse which is important in analysis and visualization of information. . Second step is to determine the granularity of analysis. It is define as the level of detail that link with a data object and ranges from very coarse to very fine. Three types of granularity has been identified which is temporal spatial, and sample in this study. Temporal is how frequent the measurements taken. It can be the time in data warehouse such as day, week, month, and years. Spatial is spatial resolution at which measurements are taken. It is determined by the extent and

resolution of the analysis. Then the sample is determined by the number of measurements take at a particular place. The replications of measurements give the user and researcher increased power in statistical analysis. The third step is determining the dimensions in the data warehouses. It is important to understand the relationships that user would like to explore. In this study, spatial and temporal dimensions were recognized as very important in summarizing data at different spatial and temporal granularities. The last step is to determine the measurements of the data warehouse. Measurements in data warehouse schema are normally based on the parameters measured as part of the research and related to statistical analysis. To suit the variety parameters and quantifiers, the system provides a dynamic system to allow users to have more ability to control how it stores the sample. There are a few spatial dimensions that is important in this study for the ecological data warehouse which are site, time, biological, chemical and hydrological data. Figure 3.3 Illustrates star schemas used in this study to develop the ecological datawarehouse system and it also the basis for designing the ERD diagram in this study.

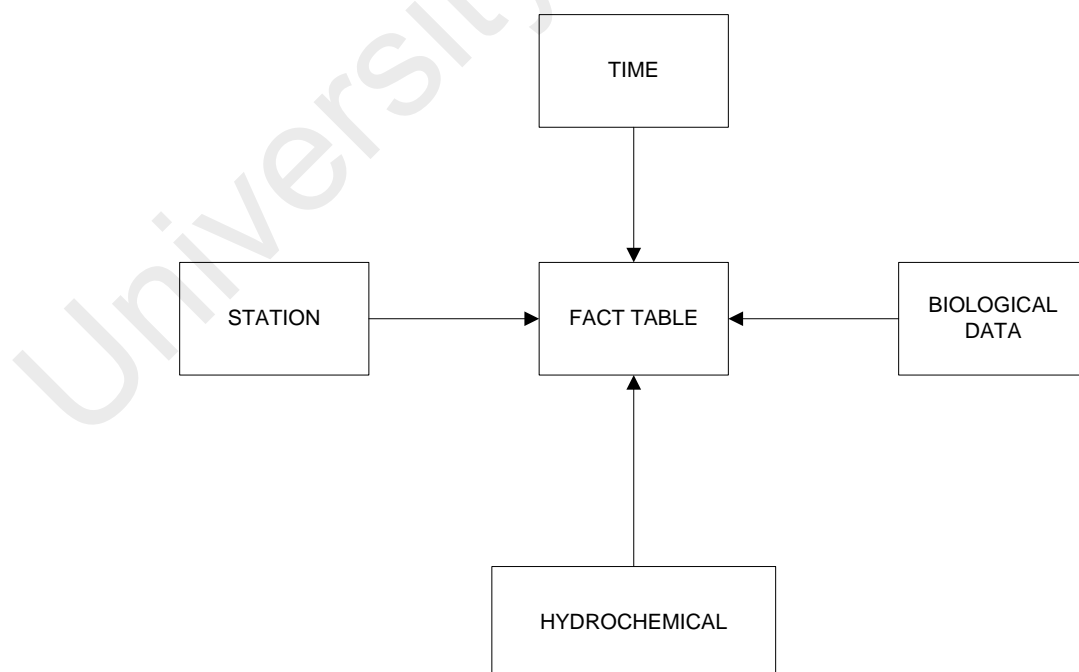


Figure 3.3 Star schema of the system

3.2 System Architecture

The system architecture as depicted in figure 3.4 below, comprises of graphical user interface (GUI) developed using ASP.Net, web services and standards as well as data mining tools. The system aims to provide information management of hydrological and ecological data on Malaysian water bodies with interactive user interface, report generation, data support for data mining tools, and data sharing between users in related field. ASP.Net is chosen because it does not have any platform independency issue and can be used on Windows operating system. Windows offers seamless integration with authentication protocols and services. The .Net framework makes users connect to Windows effortlessly to perform more advanced tasks on operating systems (OS) level. Windows built in authentication makes the building application of ASP.Net safe and secure (MSDN, 2013).

The data warehouse incorporates a module which allows users to transform data to suitable format for data mining. The module generates formatted data suitable for selected data mining tools. These data mining tools include Kohonen Self Organizing Feature Map (SOM), Hybrid Evolutionary Algorithm (HEA), and Geographic Information System (GIS). There is an increasing demand for methods and tools in the area of ecological data analysis based on novel approaches from machine learning and information theory which replace classical statistical methods. These new approaches can increase the number of tasks significantly that can be addressed with data analysis and improve quality analysis results. Data mining, on the other hand, uses machine learning methods that can utilize approaches from classical statistics as well as information theory. Machine learning tools have been successfully used for data and statistical analysis and learning of qualitative and quantitative models from the provided data. Decisions trees, decision roles and equation discovery are the most frequently used for the analysis for ecological data. Due

to their structural properties, models induced by machine learning methods are easy to interpret and can be used to forecast values of variable which can be simple or structured.

The data warehouse also deploys GIS for visualization of ecological parameter distribution over region, location and time frame. The environmental information gathered at the required spatial resolution for the entire study area can be stored in a GIS. Four main sources identified for the gathering of such environmental data are field surveys or observational studies, printed or digitized maps, remote sensing data, and maps obtained from GIS-based modeling.

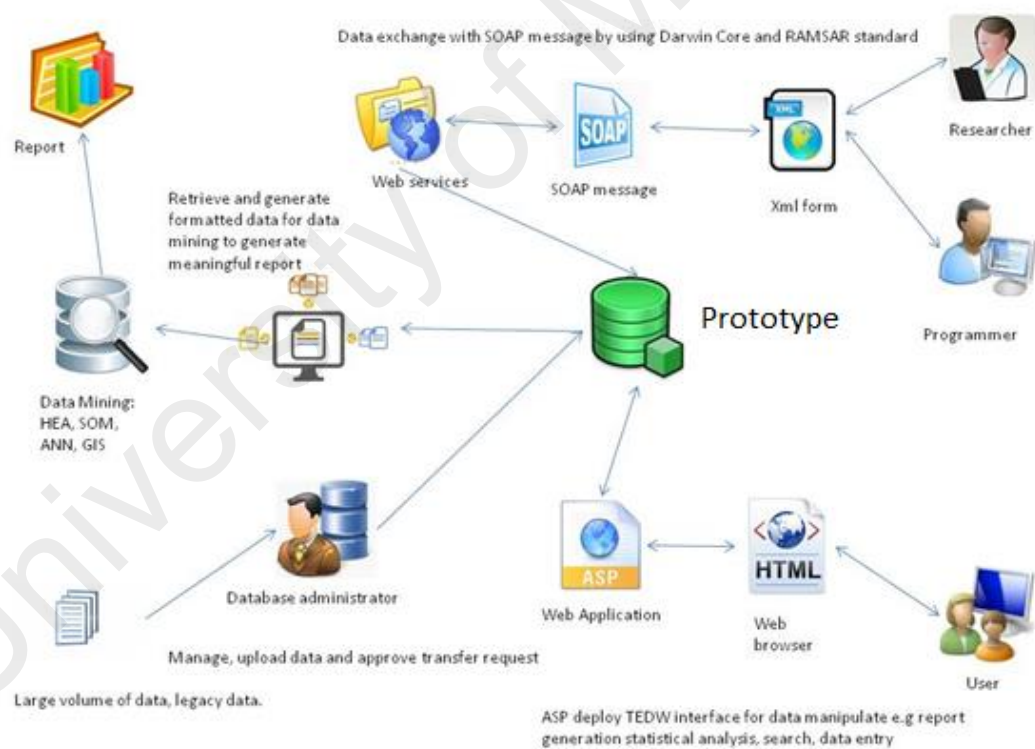


Figure 3.4 System architecture

3.3 System Development Methodology

Methodology used for the system development is the Structured Systems Analysis and Design Method (SSADM). It is traditionally associated with well-defined data modeling

methodology. SSADM is chosen in this study to plan, manage and control a project to ensure product delivery on time. It also emphasizes on the need for having the analysis and strategies to meet user requirements (Rose, 1992). Ecological datasets have high level of variability whereby the variables for each dataset can be different from one another. For an example, there are 4000 species of red algae reported in tropical lakes, and there may also be possibility of discovering new species. This allows authorized users to manipulate the variables themselves as one of its robustness feature.

The data warehouse system developed in this study comprises of five main modules (Figure 3.5). These are web-services, hydro-chemical, biological, GIS, and Site /User Control Module.

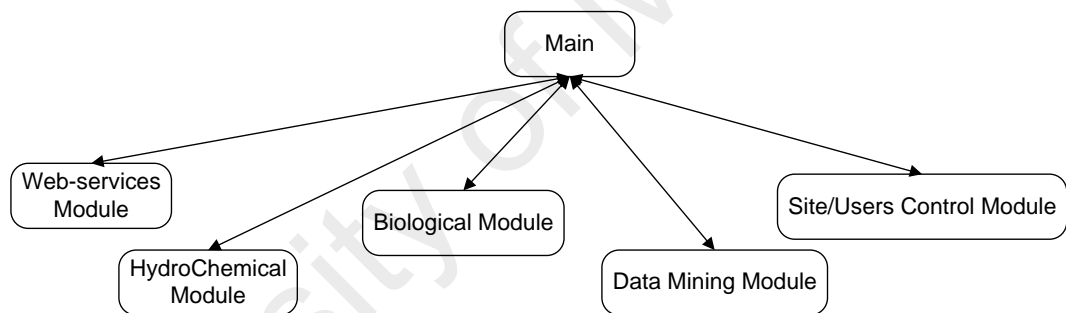


Figure 3.5 Top level view of system modules

The site / user control module in figure 3.6 is responsible for users and site management where it allows the insertion, edition, and deletion of users and site. Site can only be access by users who are assigned by the administrator. The administrator can assign roles or permissions to the users. The administrator can generate a key that allows data transfers using web services.

Site/User Module

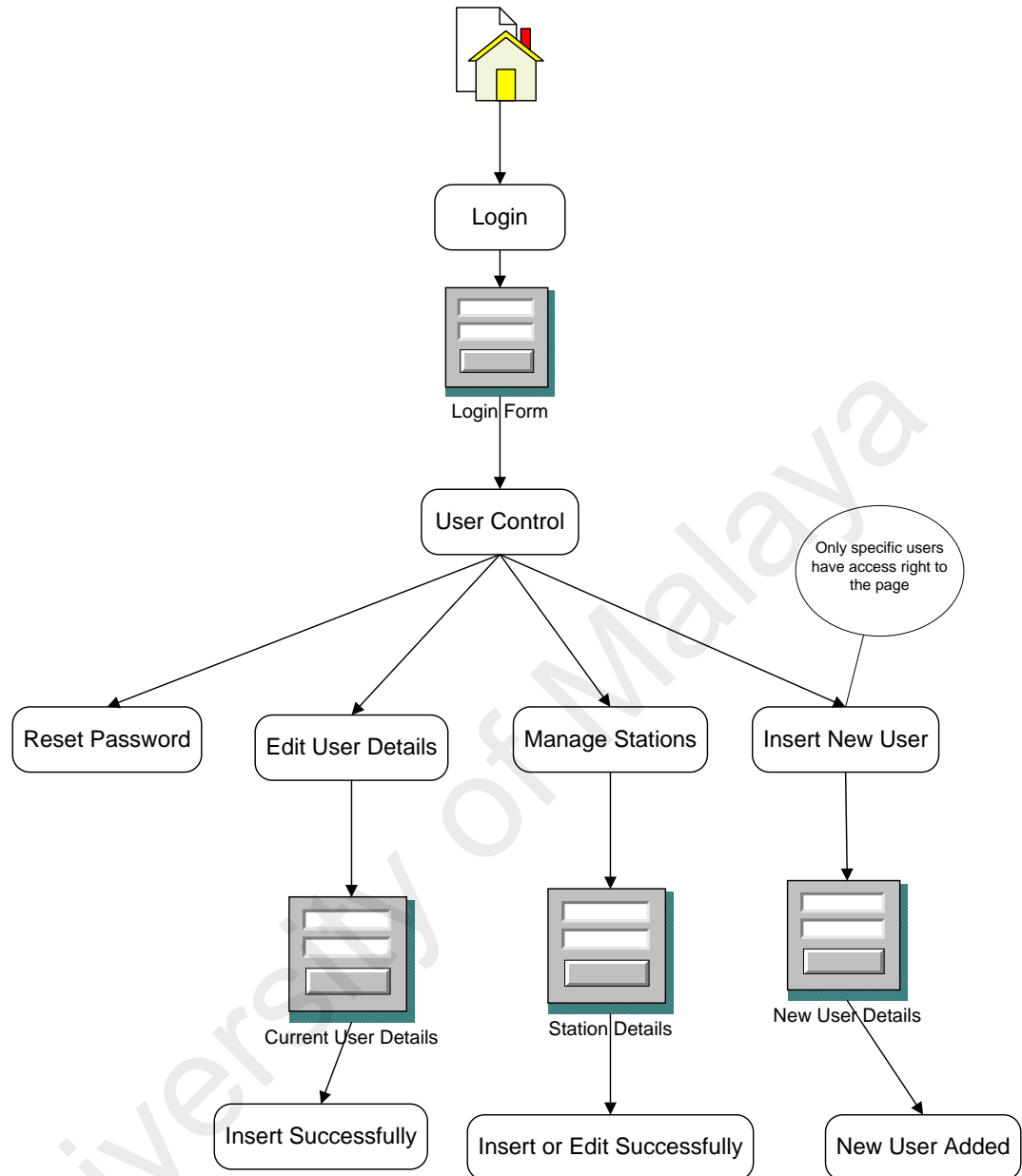


Figure 3.6 Site and User Module

The hydro-chemical module as depicted in figure 3.7 , comprises of data setting which is a one-time process as different water bodies may have different or extra parameters from exiting list of parameters in the data warehouse. Users can create, search, and edit records for each station and generate hydro-chemical report based on the criteria selected.

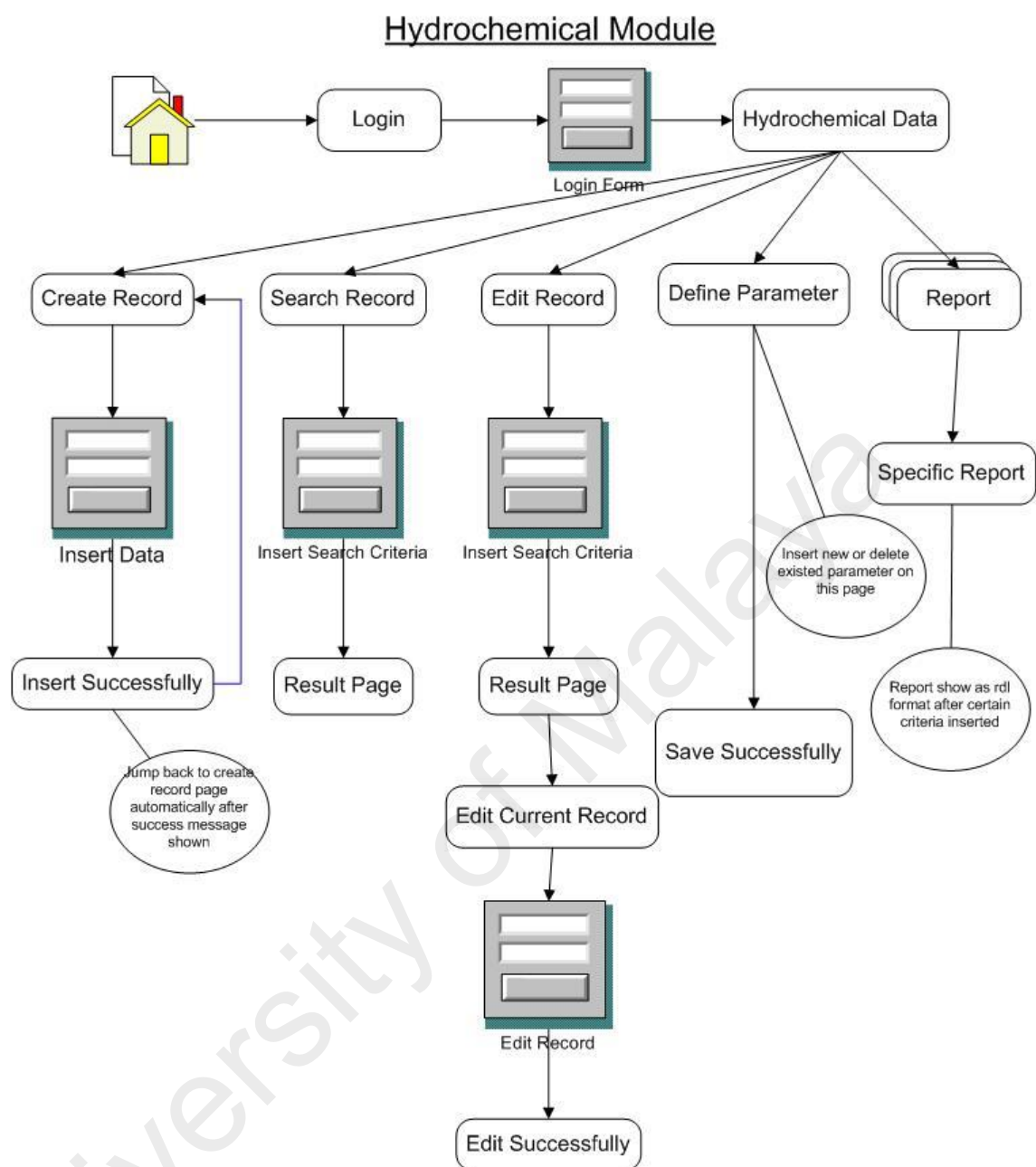


Figure 3.7 Hydro-chemical Module

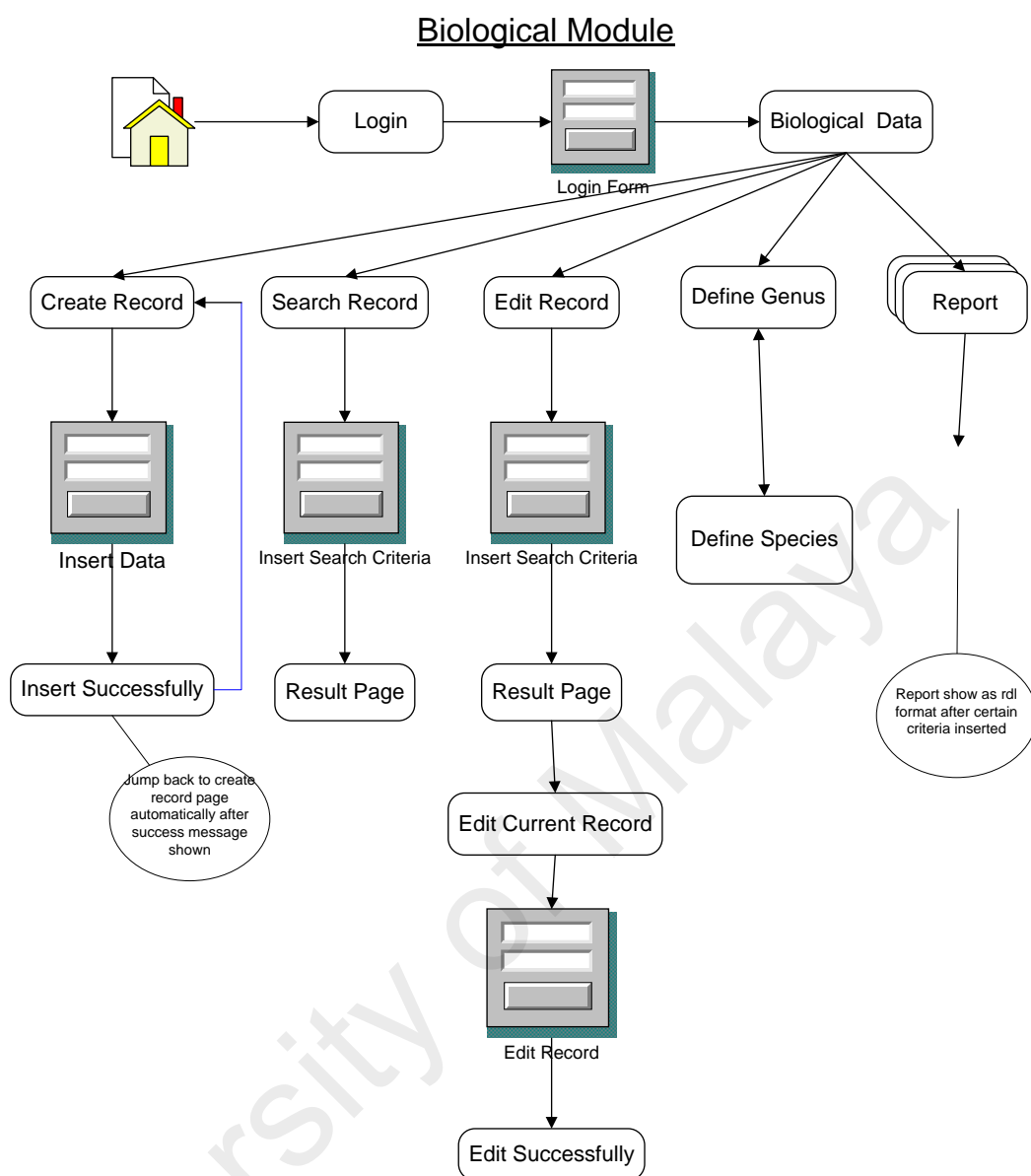


Figure 3.8 Biological Module

The biological module quite similar to the hydrochemical module is shown in Figure 3.8. The only difference is the parameter settings where users are allowed to define the taxonomy of the biological organisms or use standard taxonomy provided by the data warehouse system that is the TWDG (Taxonomic Databases Working Group) is the format based on Biodiversity Information Standards. This standard is used for data exchange using web services for biological or biodiversity data with other systems which uses similar standards.

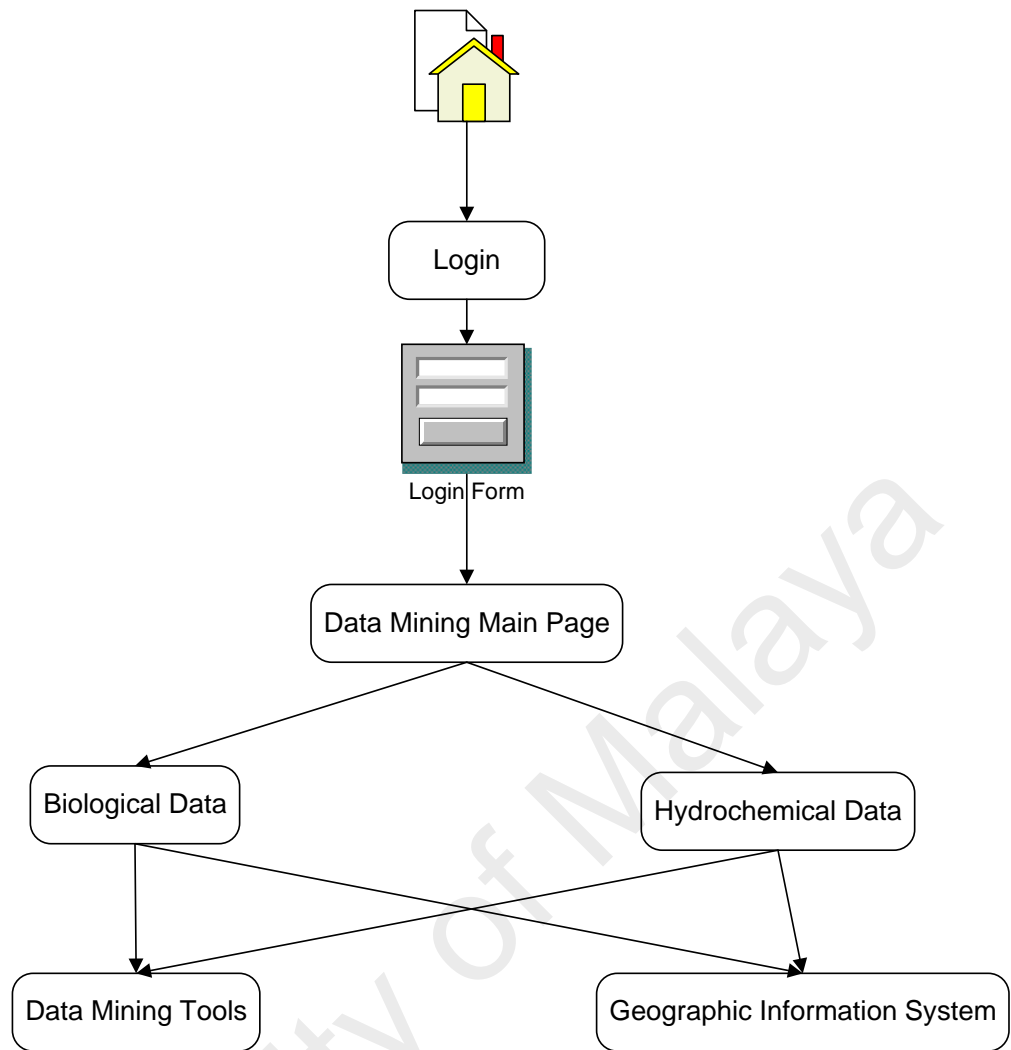


Figure 3.9 Data Mining Module

The data mining module illustrated in Figure 3.9 enables user to retrieve biological and hydrochemical data from the system transformed in a format that can be used by the data mining tools. The users and researchers can use the formatted data for data mining tools such as HEA and Neural Network. Besides this, the users can set the path to GIS file, to enable update of the database inside GIS.

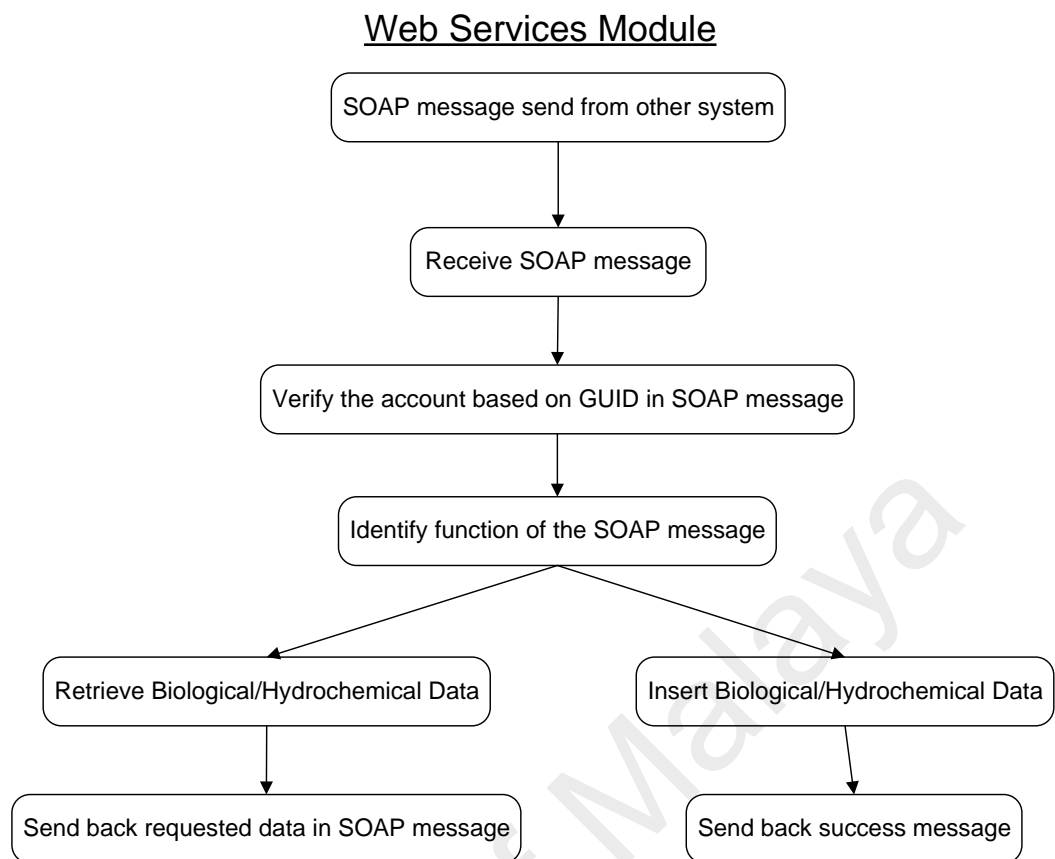


Figure 3.10 Web Services Module

All application component or coding deployed in a system can be transformed into a network service. The data warehouse developed in this study provides a web services platform for users for data communication within systems. The web service is helpful for managing biogeographic archive and analyzing habitat (Best et al, 2007). Users can integrate the data warehouse functionality into their system and act as client application to the integrated functions. The SOAP protocol is a communication message used by web services which also support for ad hoc queries. Browsing through the hyperlink and available pages on internet to search and filter data might not be a good choice. By using web services, users are allowed to query out data easily from online database.

The web service module depicted in Figure 3.10 is responsible for seamless data transfer and exchange. In order for a user from different data warehouse system to exchange data, a SOAP message sent by the data warehouse needs to be verified by the

web services of the data warehouse developed in this study. The SOAP message contains information regarding the type of data sent which will be processed by the receiving data warehouse system. The output for a successful data transfer would be generated as feedback message indicating successful data retrieval or failure to the sender. The SOAP protocol is a communication message used by web services which also support for ad hoc queries and it used in the data warehouse for data transfer.

Darwin Core is used as the data standard format in this study for the developed data warehouse system. Darwin Core is one of the complete existing formats for ecological data. Darwin Core is commonly used by the ecological and biological data management for data standard and exchange protocol. It is an XML based standard consists of basic context like properties, elements, fields, attributes, and concepts. Darwin Core was created to develop data models and manage information of biodiversity informatics in Global Biodiversity Information Facility (GBIF). The developed data warehouse uses web services to process the Darwin Core form of communication message. Besides, Darwin Core, the developed data warehouse also adopted Ramsar Classification System produced by Wetland International Body. It contains sufficient standard identification information of wetlands at each site. The developed data warehouse enables data transformation from ecological data into Extensible Markup Language (XML) format for data exchange for system migration purposes.

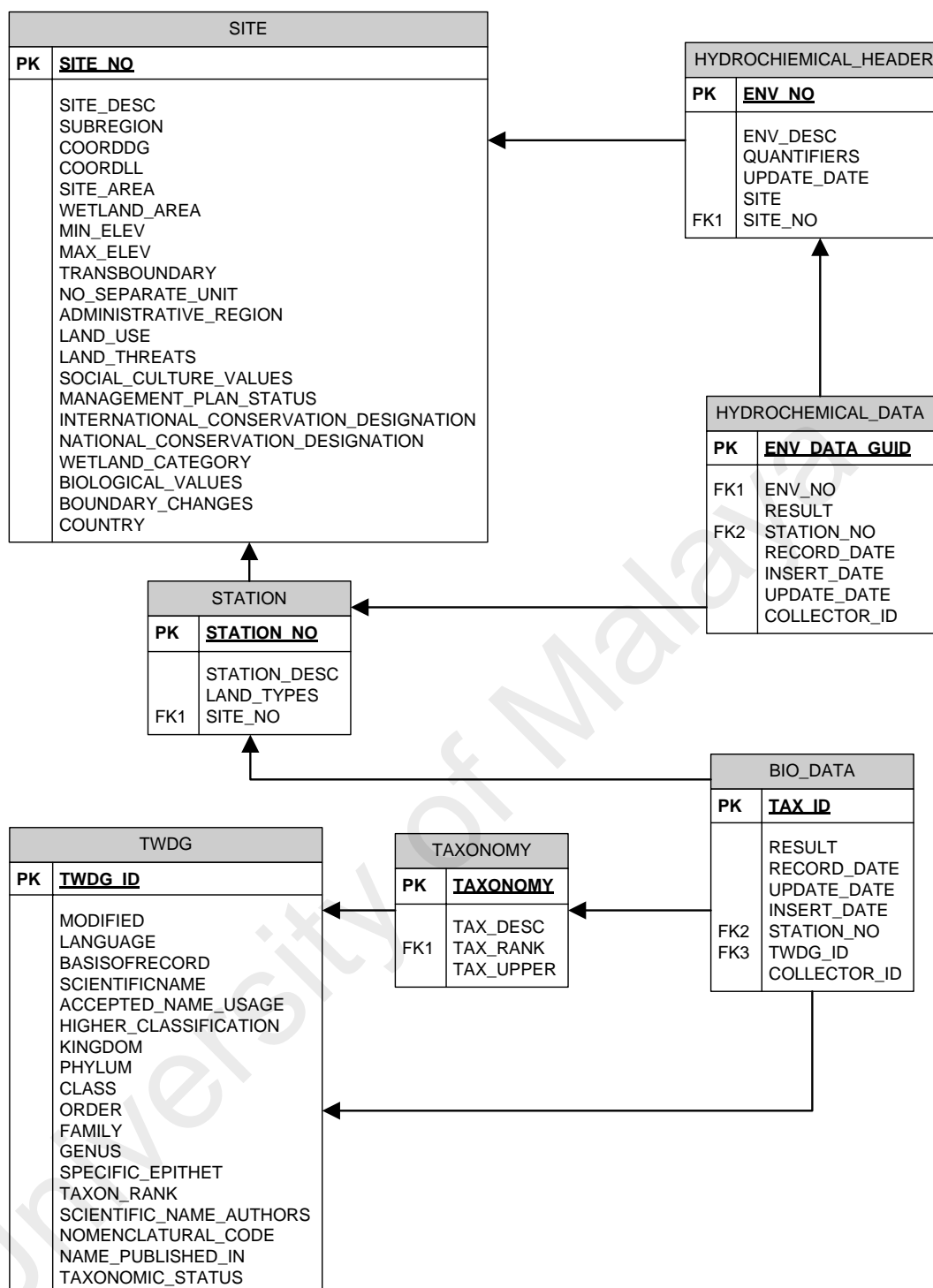


Figure 3.11 Partial Entity Relationship Diagram of the system

Figure 3.10 illustrates database structure designed based on the entity relationship diagram. The partial data structure reflects the properties of hydrological, water quality, and biological data. The hydrological data includes sunshine, rainfall, wind speed, and other environmental parameters. The water quality data should be the physical and chemical properties of water bodies which includes dissolved oxygen, heavy metals, and

concentrations of soluble and insoluble organic and inorganic. The variables will be stored in HYDROCHEMICAL_HEADER table where water quality and hydrological variables differentiated by the ENV_TYPE. The hydrological and water quality data will be stored in another table named HYDROCHEMICAL_DATA. These data can be distinctive by variable number, record date, and station. The users can delete or insert variables for water quality and hydrological. The table structure is designed dynamically to allow users to manipulate the variables. The users may add new variable by entering the variable name and its quantifier.

The biological data represents the population of organism in the lake or wetland. The biological data uses taxonomy in biological classification as variables and will be stored in TAXONOMY table. To make the system suitable for expert and non-expert users, there is function to store the data by species, genus, family or others rank name as variable. TAX_UPPER will record the upper rank of that taxonomic rank. Although all the taxonomic rank stored in the same table, but users can trace back it upper rank based on the TAX_upper. Users may insert the taxonomy with the upper rank and system will arrange the position of the taxonomy automatically. The concept of the biological data store in the BIO_DATA table is same with HYDROCHEMICAL_DATA table. The system is able to store records from multiple lakes such as biological, hydrological, and water quality data.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Result and Discussion

The developed system prototype is tested using dataset explained in chapter 3 which are Putrajaya Lake and Wetlands and Nahrim dataset. The results section explains the input and output of the developed system prototype which covers the data entry module, report generation, web services and data mining.

4.1.1 System Input

To use the developed system prototype user needs to enter details regarding a particular water body. The information is captured using RAMSAR standard illustrated in figure 4.1. The RAMSAR standard covers; general geographic site information, site uses and threats, conservation issues, management plan and biological data. The site coordinate of a RAMSAR site is also stored in the system as well and can be displayed in a map view using Google maps. Other ecological systems which are using similar standards are able to use information generated from the developed prototype. The developed system prototype also allows third party software or users to download the RAMSAR standard information on a particular water body from the system via web browser or the web services. However to enable a user to enter RAMSAR information using web services via SOAP message the users are required to register themselves as depicted in figure 4.2. Only authenticated users by the system administrator are allowed to transfer information into the system using web services method.

Add New Lake

Name :	<input type="text"/>	Total site area :	<input type="text"/>
Country :	<input type="text"/>	Total wetland area :	<input type="text"/>
Region :	<input type="text"/>	Minimum elevation :	<input type="text"/>
Subregion :	<input type="text"/>	Maximum elevation :	<input type="text"/>
Coordinates of site (degrees) :	<input type="text"/>	Transboundary :	<input type="text"/>
Coordinates of site center (decimal lat long) :	<input type="text"/>	Boundary Changes :	<input type="text"/>
		Number of Separate unit :	<input type="text"/>
		Administrative region :	<input type="text"/>

Current Land Use:	Threats:
<input type="text"/>	<input type="text"/>

Social and Culture values:	Management Plan Status:
<input type="text"/>	<input type="text"/>

International Conservation designation:	National Conservation designation:
<input type="text"/>	<input type="text"/>

Wetland Category :

Biological Values:

Figure 4.1 Input interface for water bodies using RAMSAR standard

Add User Details

Username :	<input type="text"/>
Email :	<input type="text"/>
Password :	<input type="text"/>
Organization Code :	<input type="text"/>
User Type :	<input type="text"/>
Full Name :	<input type="text"/>
Designation :	<input type="text"/>

Figure 4.2 User Registration

A registered user will be issued a user name and password by the administrator via email. The user details will be stored in the system. There are three levels of users in the system, which are the administrator, management, and end users. Administrators have full access right to the system to add, delete, modify data or users. The management levels of the users are allowed to enter and manage data regarding a particular site and the end user are only allowed to view reports without rights for data entry or modifications.

Add Station

Station Name:

Station Code:

Land Types:

Seasonal/intermittent freshwater marshes/

Latitude

Longitude

Add

Figure 4.3 Configure sampling station

Each water bodies in the system are associated with sampling station or point where data are collected and stored in the system. Water body information that has been created initially using the RAMSAR standard is associated with sampling point or station that is unique in the system. Although the water bodies are categorized into river, sea, or lake using RAMSAR standard, but each station associated with water body can belong to different land types. The station land types are categorized and identified using the RAMSAR format. Longitude and latitude of the station are stored as well. This enables

the sampling station to viewed using Google map and GIS. Table 4.1 lists down the available land types for the sampling station which appears as drop down box in system interface.

Table 4.1 Land Types of Station

Land Types
Seasonal/intermittent freshwater marshes/pools on inorganic soils
Water storage areas
Permanent shallow marine waters less than six metres deep at low tide; includes sea bays and straits.
Marine subtidal aquatic beds; includes kelp beds, sea-grass beds, tropical marine meadows.
Coral reefs.
Rocky marine shores; includes rocky offshore islands, sea cliffs.
Sand, shingle or pebble shores; includes sand bars, spits and sandy islets; includes dune systems.
Estuarine waters; permanent water of estuaries and estuarine systems of deltas.
Intertidal mud, sand or salt flats.
Intertidal marshes; includes salt marshes, salt meadows, saltings, raised salt marshes; includes tidal brackish and freshwater marshes.
Intertidal forested wetlands; includes mangrove swamps, nipah swamps and tidal freshwater swamp forests.
Coastal brackish/saline lagoons; brackish to saline lagoons with at least one relatively narrow connection to the sea.
Coastal freshwater lagoons; includes freshwater delta lagoons.
Permanent inland deltas.
Permanent rivers/streams/creeks; includes waterfalls.
Seasonal/intermittent/irregular rivers/streams/creeks.
Permanent freshwater lakes (over 8 ha); includes large oxbow lakes.
Seasonal/intermittent freshwater lakes (over 8 ha); includes floodplain lakes.
Permanent saline/brackish/alkaline lakes.
Seasonal/intermittent saline/brackish/alkaline lakes and flats.*
Permanent saline/brackish/alkaline marshes/pools.
Seasonal/intermittent saline/brackish/alkaline marshes/ pools.*
Permanent freshwater marshes/pools; ponds (below 8 ha), marshes and swamps on inorganic soils; with emergent vegetation water-logged for at least most of the growing season.
Seasonal/intermittent freshwater marshes/pools on inorganic soil; includes sloughs, potholes, seasonally flooded meadows, sedge marshes.*
Non-forested peatlands; includes shrub or open bogs, swamps, fens.
Alpine wetlands; includes alpine meadows, temporary waters from snowmelt.
Tundra wetlands; includes tundra pools, temporary waters from snowmelt.
Shrub-dominated wetlands; Shrub swamps, shrub-dominated freshwater marsh, shrub carr, alder thicket; on inorganic soils.*

Freshwater, tree-dominated wetlands; includes freshwater swamp forest, seasonally flooded forest, wooded swamps; on inorganic soils.*
Forested peatlands; peatswamp forest.*
Freshwater springs; oases.
Geothermal wetlands.
Subterranean karst and cave hydrological systems.
Aquaculture (e.g. fish/shrimp) ponds.
Ponds; includes farm ponds, stock ponds, small tanks; (generally below 8 ha).
Irrigated land; includes irrigation channels and rice fields.
Seasonally flooded agricultural land.**
Salt exploitation sites; salt pans, salines, etc.
Water storage areas; reservoirs/barrages/dams/impoundments; (generally over 8 ha).
Excavations; gravel/brick/clay pits; borrow pits, mining pools.
Wastewater treatment areas; sewage farms, settling ponds, oxidation basins, etc.
Canals and drainage channels, ditches.

The image shows a web-based interface for configuring variables. It has a light blue background. At the top, there's a label 'Type of Variable:' followed by a dropdown menu showing 'Hydrological'. Below this is a label 'Insert New Variable:' followed by a text input field. Underneath that is a label 'Quantifier:' followed by another text input field. At the bottom left, there is a button labeled 'Add'.

Figure 4.4 Variable configuration

Figure 4.4 illustrates interface to configure the input variable associated with each water bodies. This provides flexibility which allows users to modify environmental variables and quantifiers according the nature of their data. This step is essential as each water bodies have different hydrological and chemical variables data collected. To cater for these variations in types of data collected for each different water bodies the developed system in this study allows users to add, create, customize or modify variables associated

with each unique water body. The users are also able to customize the quantifiers for each variable as measurement method and quantity of the data can be different for different organization or water bodies.

Tropical Ecological Data Warehouse

Putrajaya Lake

STATUS HYDROLOGICAL CHEMICAL BIOLOGICAL REPORT

Station: CW

Record Date: May 2011

Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Hydrological Data:

Rainfall	0	mm
Wind Speed	0	m/s
Sunshine	0	hr
Temperature	0	celcius
DO	0	mg/l
Sec	0	m
Turb	0	NTU
Cond	0	uS/cm
BOD	0	mg/l
COD	0	mg/l

Save Clear

Figure 4.5 Data entry for Hydrological and Chemical Data

Figure 4.5 illustrates data entry for hydrological variables that have been configured for a particular water body. The user needs to select the sampling station assigned to them and date before entering the data. Users are given an option to use SOAP messages for web services to transfer data into the system. This eliminates the need for manual data entry.

Taxonomy Rank:

Kingdom

Name:

☐

Upper Rank

Kingdom

Upper Rank Name:

Save Clear

Figure 4.6 Variable Setting for Biological Data

< September 2015 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Station:

1CH02

Taxonomy Name:

Compsopogonophyceae

Quantity

Save Clear

Figure 4.7 Data Entry for Biological Data

Figure 4.6 illustrates variable setting for biological data which can represents; fish, mammals, amphibians, insects, phytoplankton and plant. The settings are designed based on the Darwin Core standard. The biological data setting is based on taxonomy level of a species which comprises kingdom, family, genus and species. The system allows users to

select the taxonomic level for which each data will be stored. Some of the dataset collected are up to family level, but some recorded genus and species level. To cater for this scenario, the system allows the users to key in the complete taxonomy rank or select any level of taxonomy based on the data collected.

4.2 System output

This section illustrates the developed system output using dataset collected from Putrajaya and NAHRIM.

The developed data warehouse provides real time data entry, searching, and editing to produce meaningful statistical report via internet browser or web services. The temporal and spatial environmental data of lake and wetlands can be formatted into Ramsar classification which can viewed on the web as shown in figure 4.8. This information as discussed earlier can be transferred via web services using SOAP protocol to other ecological databases.

The data warehouse system provides search function using wildcards, selection list, and calendar. These detailed search results included hydrological, biological, chemical, geographical data for specific time stamp, station, variable, or parameters.

Statistical data and reports generated by the system are based on user selection of region, location, and input variables. These reports can be exported into excel or .pdf files to allow user manipulation of the data.

Ramsar Site Data Entry	
General and Geographical site Information	Uses and Threats
 <p>General and Geographical site Information</p> <p>Name: Tasik Bera Country: Malaysia Region: Asia Subregion: Southeastern Asia Coordinates of site center(degrees): 02°58'00"N 102°39'00"E Coordinates of site center(decimal lat long): 2.97 102.65 Total site area: 38448 Total wetland area: 6100 Minimum elevation: 30.000 Maximum elevation: 35.000 Transboundary: no Boundary Changes: no Number of Separate unit:</p>	<p>Uses and Threats</p> <p>Current Land Use: Within the site's surroundings/catchment: Commercial forestry Plantation (non-orchard type/scale) Within the Ramsar site: Non-urbanized settlements Tourism (unspecified) Hunting (unspecified) Fishing (unspecified) Collection of non-timber natural vegetative products: (unspecified) Freshwater aquaculture Irrigation Shifting arable agriculture Agriculture (unspecified)</p> <p>Threats: Within the site's surroundings/catchment: Erosion Commercial scale forest exploitation Within the Ramsar site: Pesticide/herbicide pollution Unspecified pollution Disturbance from transport artery Sedimentation/siltation Mining exploitation/exploration impacts (unspecified) Drainage/reclamation for industry Erosion Loss of natural vegetation through slash & burn agriculture Desertification from agricultural activities Disturbance to vegetative community through cutting/clearing Habitat burning not related to agriculture</p> <p>Social and Culture values: Water supply Tourism Current scientific research Non-consumptive recreation Unspecified fishing</p>
General and Geographical site Information	Information and Conservation Issues
<p>Information and Conservation Issues</p> <p>Management Plan Status: Site-specific plan in preparation, for part(s) of site (P)</p> <p>International Conservation designation: no other international designation besides Ramsar</p> <p>National Conservation designation: forest reserve</p>	<p>Ecological Data</p> <p>Wetland Category: Inland Wetlands</p> <p>Biological Values: Fauna type: supports rare/endangered mammal species important for mammals outstanding variety of species present crocodilians present Flora type: supports endemic species</p>

Figure 4.8 Ramsar report layout a) general site geographic information b) Uses and Threats c) Information and Conservation Issues d) Ecological Data

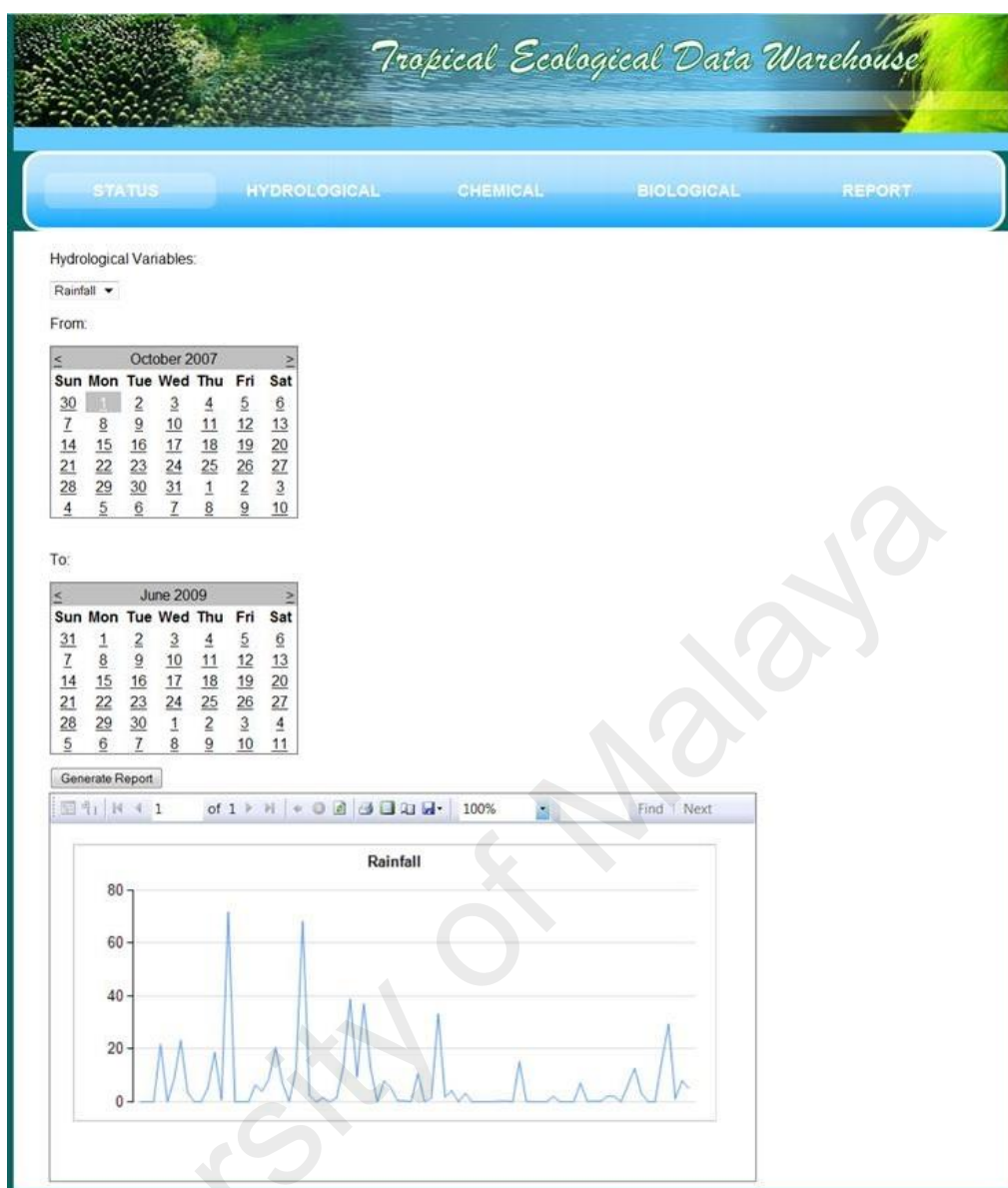


Figure 4.9 Hydrological Data Monthly Report

Figure 4.9 illustrates report generation function using rainfall data from Putrajaya Lakes. The report can be generated for daily, monthly, and yearly data. The data is averaged when monthly or yearly options are chosen.

The system also provides an option to query data based on a single or multiple sampling station or water bodies as illustrated in figure 4.10. Similar options are available for biological and chemical datasets. This option allows lake managers to make comparison across multiple stations based on daily, monthly, or yearly data for a particular parameter. The queried data can be exported into excel files as illustrated in

figure 4.11. The report is separated into graph and data table part in excel file format which makes it easier for the user to analyze the data.

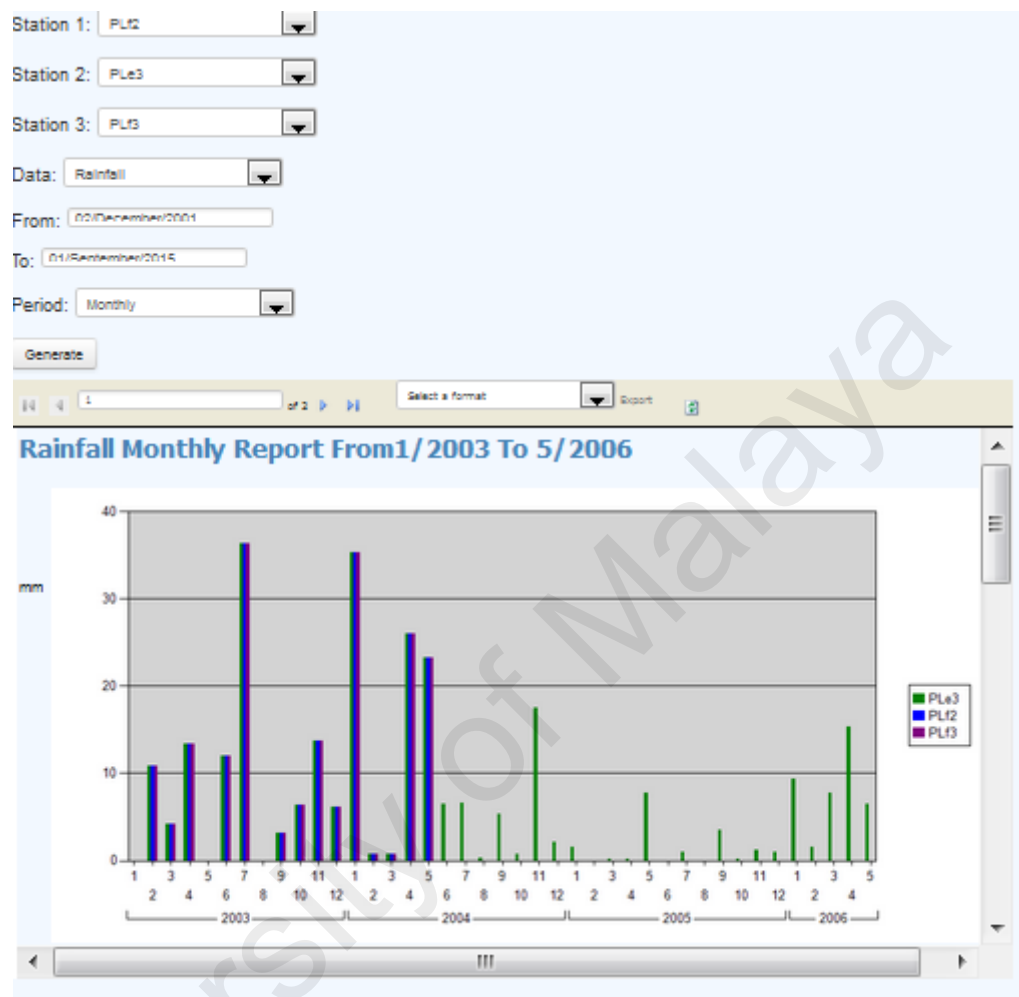


Figure 4.10 Multiple Station vs Time series monthly rainfall report

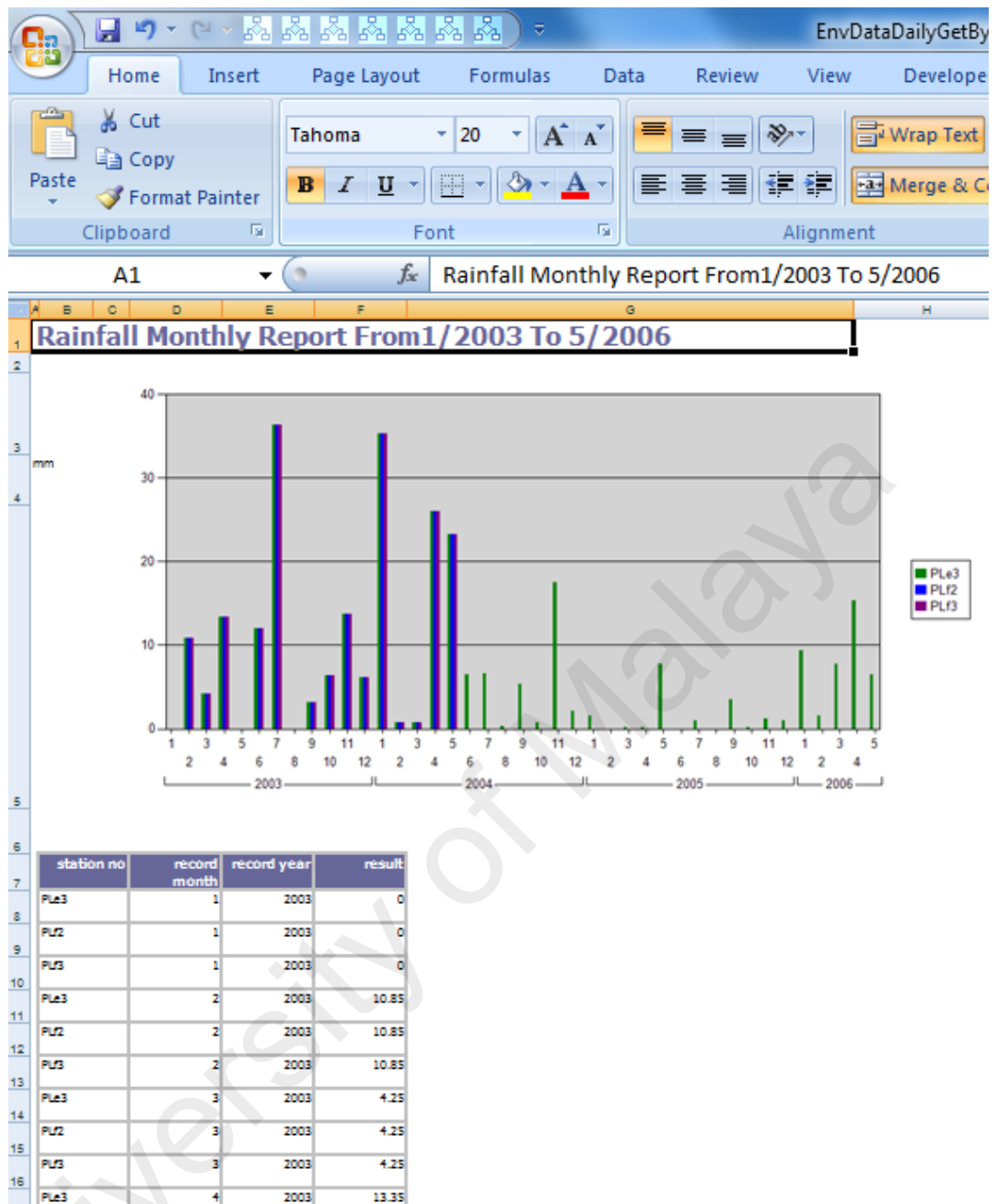


Figure 4.11 Exported Excel Report for Multiple Station VS Time Rainfall Monthly Report

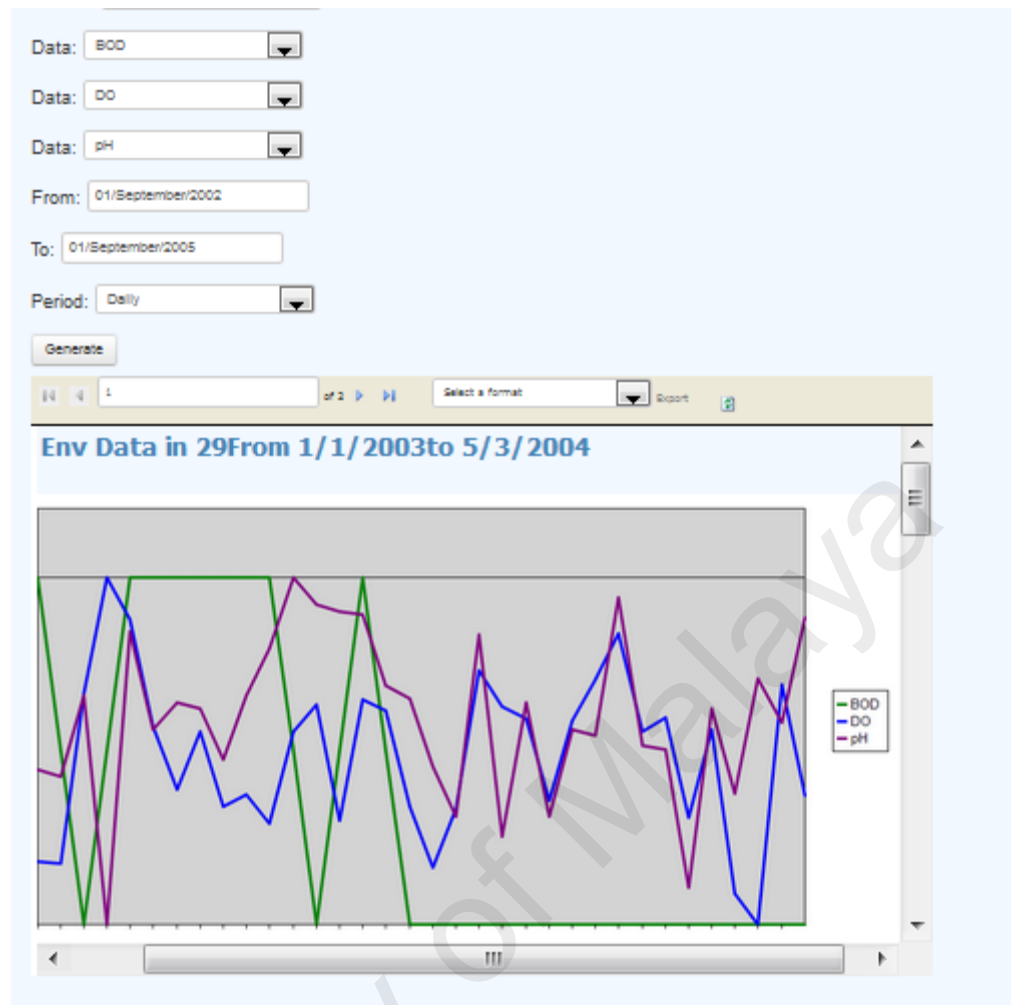


Figure 4.12 Multi Hydrological Data VS Time Daily Report

Multiple variables comparison for a particular water body or sampling station is an important analysis. This enables researchers to see how different variables influence each other and it is useful information for water quality management. This is shown in figure 4.12 using multiple variables from hydrological dataset. The report generated is based on the data normalization concept which enables variables of different units to be compared. The user can still retrieve the non-normalized values of the data once the report has been exported into excel format.

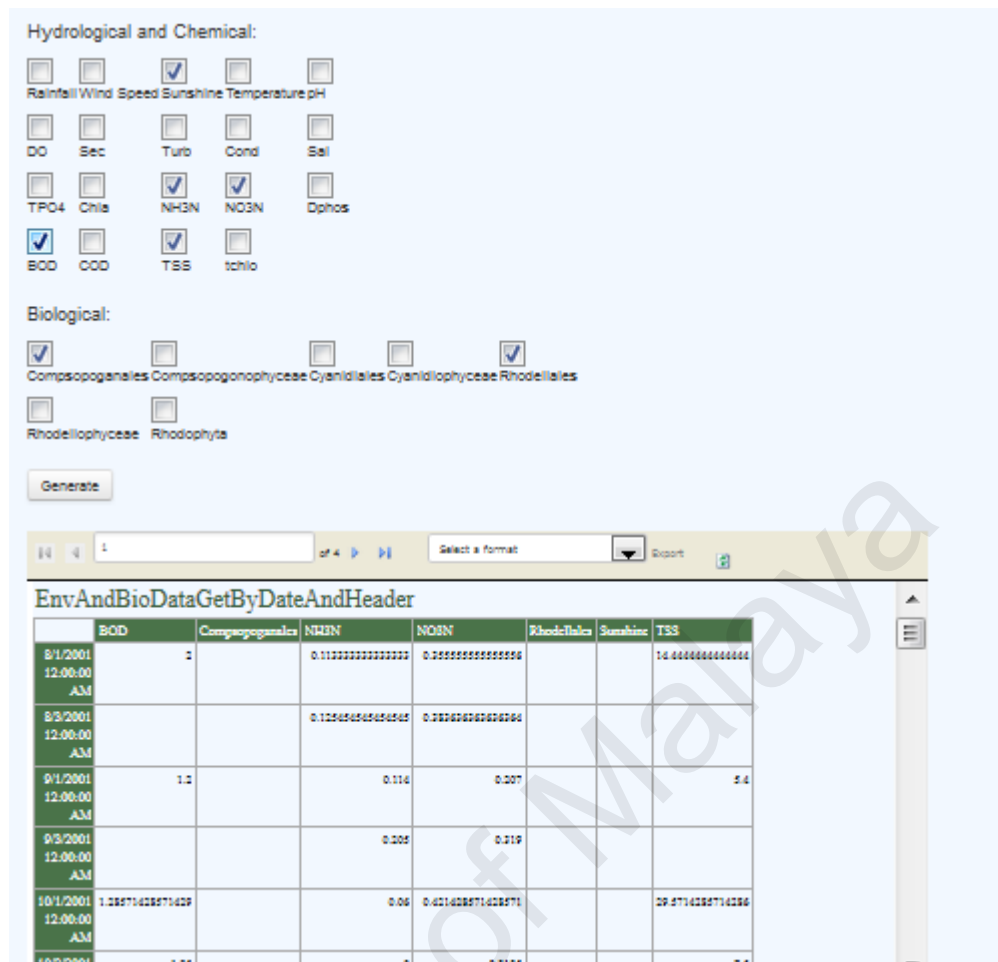


Figure 4.13 Matrix Report

Figure 4.13 illustrates matrix report which is basic form of report available in most data warehouse system. The matrix report can be generated for hydrological, biological and chemical dataset and it can be downloaded into various format such as into csv, excel, txt format that can be used latter for data analysis using data mining tools like HEA, ANN, or SOM.

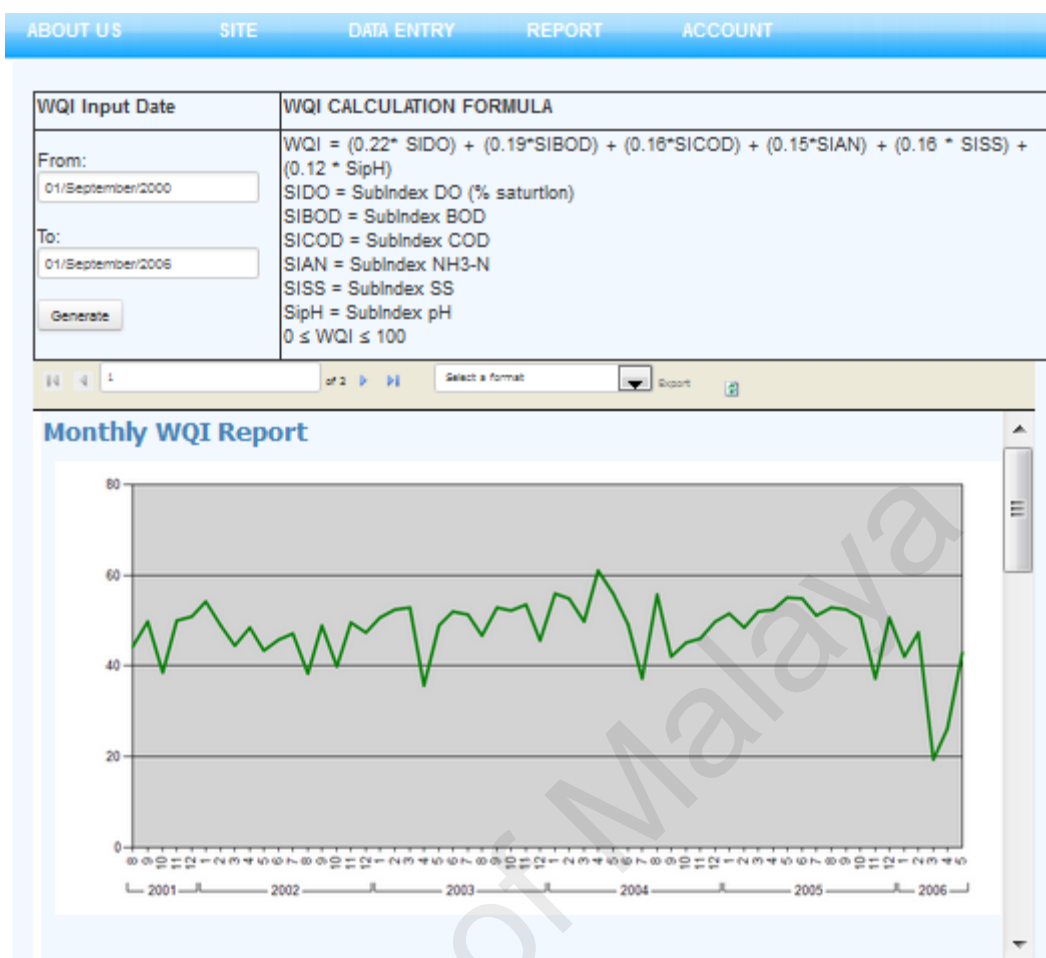


Figure 4.14 WQI Report

Besides RAMSAR report another type of report which is important in maintenance and governance of water quality is the WQI report. The WQI report is based on calculation provided by the WEPA (Water Environment Partnership in Asia). The DOE (Department of Environment) used WQI to evaluate status of the river water quality. WQI serves as a foundation for environment assessment of a watercourse in relative to pollution level provided for under National Water Quality Standards for Malaysia (NWQS). The system generates accurate daily, monthly, and yearly WQI report after users selects required data.

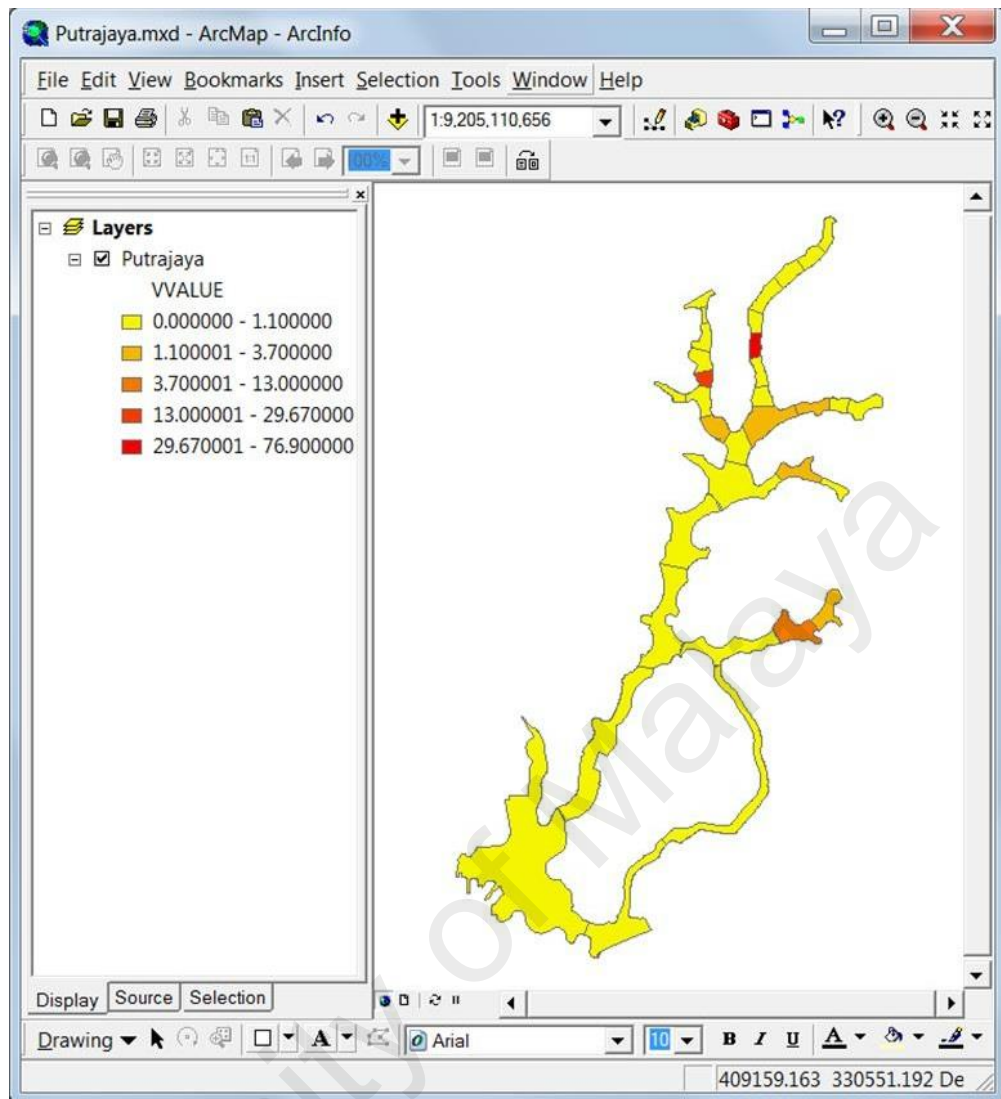


Figure 4.15 Chlorophyll level visualization on Putrajaya Lake map using ArcMap demonstrating chlorophyll level

Data visualization is an important aspect in ecological data warehouse Figure 4.15 illustrates chlorophyll a concentration in Putrajaya Lake The developed ecological data warehouse uses ArcGis to visualize selected data by users which can be hydrological, chemical, or biologically the database file of the Putrajaya Lake map in ArcGIS will be updated and the selected data value will be visualized on the map.

To enable seamless data transfer developed system uses web services to process the Darwin Core form of communication message. Besides Darwin Core, System also supports data transfer format for Ramsar Classification System produced by wetland

International Body (Figure 4.16). Web services convert ecological data that needs to be transferred into Extensible Markup Language (XML) format (Figure 4.17).

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /EDNWS.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/DarwinCoreSimpleGet"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DarwinCoreSimpleGet xmlns="http://tempuri.org/">
      <Species_No>string</Species_No>
    </DarwinCoreSimpleGet>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DarwinCoreSimpleGetResponse xmlns="http://tempuri.org/">
      <DarwinCoreSimpleGetResult>
        <xsd:schema>schema</xsd:schema>xml</DarwinCoreSimpleGetResult>
      </DarwinCoreSimpleGetResponse>
    </soap:Body>
  </soap:Envelope>
```

Figure 4.16 SOAP message for request and response within prototype and other parties

```
- <diffgr:diffgram>
- <NewDataSet>
- <Table1 diffgr:id="Table1" msdata:rowOrder="0">
  <modified>2006-05-04T18:13:51.0Z</modified>
  <language>en</language>
  <basisOfRecord>Taxon</basisOfRecord>
  <scientificName>Centropyge flavicauda Fraser-Brunner 1933</scientificName>
  <acceptedNameUsage>Centropyge fisheri (Snyder 1904)</acceptedNameUsage>
  <parentNameUsage>Centropyge Kaup, 1860</parentNameUsage>
- <higherClassification>
  Animalia;Chordata;Vertebrata;Osteichthyes;Actinopterygii;Neopterygii;Teleostei;Acanthopterygii;Perciformes;
</higherClassification>
  <kingdom>Animalia</kingdom>
  <phylum>Chordata</phylum>
  <class>Osteichthyes</class>
  <order>Perciformes</order>
  <family>Pomacanthidae</family>
  <genus>Centropyge</genus>
  <specificEpithet>flavicauda</specificEpithet>
  <taxonRank>species</taxonRank>
  <scientificNameAuthorship>Fraser-Brunner 1933</scientificNameAuthorship>
  <nomenclaturalCode>ICZN</nomenclaturalCode>
```

Figure 4.17 XML file formatted based on Darwin Core format.

4.3 Discussion

Researchers and water resource managers are able to access a large and increasing quantity of data for analysis, visualization, and modeling of the water environment. The number and few kinds of available data sources, however, make it difficult to quickly locate the most appropriate resource for a particular study. Furthermore, once the most appropriate data source has been identified, a considerable amount of effort is still

required to reformat the data for analysis and visualization, or modeling. As a result, researchers and users spend a lot of time on basic data gathering and transformations, instead of scientific analysis and decision making.

This study discovered a number of interface requirements needed to support testing and hypothesis generation for the framework formulation. First, the system must have the function to allow the researcher to view multiple measurements from a number of viewpoints. A researcher basically developed a deep understanding on the dataset through exploration before performing any type of analysis or testing. It included the development of visualizations from a dataset in order to find relationships, trends, and temporal patterns. It makes an exploratory method is required to allow the researcher to interact with the dataset to observe how the variables are related, research question, generate report and charts, and time series visualizations.

Then the proposed system should enable researchers to find out the patterns and relationship easily from the data. Samples taken from sites and visualizing it with GIS and other data mining prove to be helpful in other research (McGuire et al., 2008). As some of the researchers may be inexperienced in GIS or data mining tools, database system should include an interface to allow researchers to explore the GIS and data mining tools without having to manage this complex software.

The developed data warehouse provides real time data entry, searching, and editing to produce meaningful statistical report via internet browser or web services. The temporal and spatial environmental data of lake and wetlands can be formatted into Ramsar classification. The data warehouse system provides search function using wildcards, selection list, and calendar. These detailed search results included hydrological, biological, chemical, geographical data for specific time stamp, station, variable, or parameters. The Darwin Core standard and classification system of Ramsar are deployed

for input biological data and the developed data warehouse also provides flexibility which allows users to modify environmental variables and quantifiers according the nature of their data.

Statistical data and reports generated by the system are based on user selection of region, location, and input variables. These reports can be exported into excel or .pdf files to allow user manipulation of the data. Visualization of data using GIS as by the data warehouse system allows users to displays the density of hydrological data by directly importing data from the system. Formatted data from data warehouse system can be used for data mining tools such as HEA and ANN for prediction models.

Real time or frequently up-to-date data are required for more accurate analysis and decision making. Critical decision can be made by using the suites of predictions (Steel et al., 2008). This could not be accomplished as the current tropical lakes databases does not provide any data mining tools and sufficient and up-to-date data to researcher and expertise to do prediction or analysis to assist decision makers. Data mining tools such Kohonen Self Organizing Feature Map (SOM), Hybrid Evolutionary Algorithm (HEA), and Geographic Information System (GIS) are widely used for water quality monitoring. HEA generate the rule sets or arithmetic functions to forecast water quality or biological data. HEA provides highly understandable rule models and important information based on threshold values of water quality conditions. It improves the understanding of ecological relationships of algal growth in basic. (Cao et al., 2013). SOM used for modeling ecological data accurately to find similarity between dataset. It also discover the disturbances within the environmental effects and suitable to implement for monitoring and managing ecosystem quality (Malek et al., 2009). Combining SOM and HEA may enhance understanding of complex community environment relationships for targeted water body (Chon, 2011). GIS is usually used for clustering, visualization, and

discovery of relationship of ecological data for the ecological modeling (Santos et al., 2006; Aspinall and Pearson, 2000).

In recent years, Evolutionary Algorithms (EA) have earned popularity for data-driven modeling due to their self-learning capacity, generality, and intrinsic parallelism and (Best et al., 1997), and their distinct capacity for explicit representation of make models by multivariate rules or functions. Since earlier work of EA applications for ecological modeling by Bobbin and Recknagel (2001) and Whigham and Recknagel (2001) have developed the hybrid evolutionary algorithm (HEA) that is now applied for data-driven modeling of cyanobacteria blooms in lakes and rivers worldwide (Recknagel et al., 2008).

All application component or coding deployed in a system can be transformed into a network service. The developed data warehouse system provides a web services platform for users for data communication within systems. The web service is helpful for managing biogeographic archive and analyzing habitat (Best et al., 2007). Furthermore, ecological data are inherently spatial and may involve making observations about structural elements, which are less likely to diverge over time or between different studies, and which also can be used as join points. If databases are built by common and interchangeable representations of spatial data and coherent conceptualization of ecological structural elements, these databases can be more easily managed, provide metadata, allow the development of tools which can be used on many databases, and provide common variables over which some dissimilar data sets can be joined. Functional data can be inserted in an ad-hoc manner without affecting the performance of structure-based templates to serve the benefits of common components. Users can integrate the data warehouse functionality into their system and act as client application to the integrated functions. The developed data warehouse system provides authentication via ASP.Net client side page online to process users access right and data transfer. The data warehouse

system allows expert users to embed web services into their own system for data sharing and transfer. The embedded web services at the client/user side enables archived data by users to be transformed into a XML standard message called SOAP. The system uses XML version similar to what is used by Darwin Core protocol that is version 1.0 and UTF-8 for the encoding. The system supports imports of large volume of data in Excel style, access or text file using default function of MSSQL.

University of Malaya

CHAPTER 5: CONCLUSION

The data warehouse framework proposed in this study has the flexibility to adapt to various types of tropical water bodies such as lakes, wetlands and rivers. The data warehouse developed based on the framework archives fragmented ecological data to be processed and presented into a standardized format for data transfer and manipulation to solve the issues of data standard, data sharing and data incompleteness. Users and researchers can access the system easily with internet connections for data exchange and generation of reports. The data warehouse system can be integrated into other ecological data system easily because it is equipped with metadata for data migration. The future enhancement will focus on the remote sensor for data collection in real time. The web services provide a suitable platform for data entry by using remote sensor. Web services can be used to collect remote sensed data. It is suitable for small scale hydrological applications. Although the remote sensor has limited processing performance and transferring small dataset in real time with low internet speed, web services can overcome these limitations to assist hydrologic data distribution system.

REFERENCES

- Aspinall, R., & Pearson, D. (2000). Integrated geographical assessment of environmental condition in water catchments: Linking landscape ecology, environmental modelling and GIS. *Journal of Environmental Management*, 59(4), 299-319.
- Alonso, J. C., Martín, C. A., Alonso, J. A., Palacín, C., Magaña, M., & Lane, S. J. (2004). Distribution dynamics of a great bustard metapopulation throughout a decade: influence of conspecific attraction and recruitment. *Biodiversity & Conservation*, 13(9), 1659-1674.
- Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997). Handbook of evolutionary computation. Release, 97(1), B1. IEEE.
- Berkley, C., Jones, M., Bojilova, J., & Higgins, D. (2001). Metacat: a schema-independent XML database system. In *Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on* (pp. 171-179). IEEE.
- Best, B. D., Halpin, P. N., Fujioka, E., Read, A. J., Qian, S. S., Hazen, L. J., & Schick, R. S. (2007). Geospatial web services within a scientific workflow: Predicting marine mammal habitats in a dynamic environment. *Ecological Informatics*, 2(3), 210-223.
- Bobbin, J., & Recknagel, F. (2001). Knowledge discovery for prediction and explanation of blue-green algal dynamics in lakes by evolutionary algorithms. *Ecological Modelling*, 146(1), 253-262.
- Cao, H., Recknagel, F., & Orr, P. T. (2013). Enhanced functionality of the redesigned hybrid evolutionary algorithm HEA demonstrated by predictive modelling of algal growth in the Wivenhoe Reservoir, Queensland (Australia). *Ecological Modelling*, 252, 32-43.
- Cao, H., Recknagel, F., Cetin, L., & Zhang, B. (2008). Process-based simulation library SALMO-OO for lake ecosystems. Part 2: Multi-objective parameter optimization by evolutionary algorithms. *Ecological Informatics*, 3(2), 181-190.
- Chon, T. S. (2011). Self-organizing maps applied to ecological sciences. *Ecological Informatics*, 6(1), 50-61.
- Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., & Weerawarana, S. (2002). Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet computing*, 6(2), 86.
- Dr. Marko Debeljak, Josef Stefan, Ljubljana Slovenia. (2012) Introduction to Data Mining for Forest Ecology and Forestry.
- Edwards, J. L. (2004). Research and societal benefits of the Global Biodiversity Information Facility. *BioScience*, 54(6), 485-486.

- Frehner, M., & Brändli, M. (2006). Virtual database: Spatial analysis in a Web-based data management system for distributed ecological data. *Environmental Modelling & Software*, 21(11), 1544-1554.
- Freitas, M. C., Duarte, J. C., Santos, C. N., Milidiú, R. L., Rentería, R. P., & Quental, V. (2006). A machine learning approach to the identification of appositives. In *Advances in Artificial Intelligence-IBERAMIA-SBIA 2006* (pp. 309-318). Springer Berlin Heidelberg.
- Goodall, J. L., Horsburgh, J. S., Whiteaker, T. L., Maidment, D. R., & Zaslavsky, I. (2008). A first approach to web services for the National Water Information System. *Environmental Modelling & Software*, 23(4), 404-411.
- Global Biodiversity Information Facility, 2013 Accessed October 10 - <http://www.gbif.org/>
- Gurtz, M. E. (1994). Design of biological components of the National Water-Quality Assessment (NAWQA) program. *Biological monitoring of aquatic systems: Boca Raton, Fla.*, Lewis Publishers, 323-354.
- Hall, F. G., & Sellers, P. J. (1995). First international satellite land surface climatology project (ISLSCP) field experiment (FIFE) in 1995. *Journal of Geophysical Research: Atmospheres* (1984–2012), 100(D12), 25383-25395.
- Hedges, M., & Rose, G. (1992). *SSADM Version 4 Book*. Pitman.
- ILEC (2010). International Lake Environment Committee Foundation, Retrieve February 2010 www.ilec.or.jp.
- JonathanL. Goodall et al (2008), A first approach to web services for the National Water Information System. *Environmental Modelling & Software*. Volume 23, Issue 4, Pages 404-411
- Labarga, A., Valentin, F., Anderson, M., & Lopez, R. (2007). Web services at the European bioinformatics institute. *Nucleic acids research*, 35(suppl 2), W6-W11.
- LakeNet (2012). World Lkes Website, Retrieve February 2010 www.worldlakes.org.
- Lehner, B., & Doll, P. (2004). Global Lakes and Wetlands Database GLWD. GLWD Documentation.
- Liang, S. (2005). *Quantitative remote sensing of land surfaces* (Vol. 30). John Wiley & Sons.
- Malek, S., Salleh, A., & Ahmad, S. M. S. (2009, April). Analysis of Algal Growth Using Kohonen Self Organizing Feature Map (SOM) and its Prediction Using Rule Based Expert System. In *Information Management and Engineering, 2009. ICIME'09. International Conference on* (pp. 501-504). IEEE.
- McLaren, C. Graham , Bruskiewich, R. M., Portugal, A. M., & Cosico, A. B. (2005). The International Rice Information System. A platform for meta-analysis of rice crop data. *Plant Physiology*, 139(2), 637-642.

- Mineter, M.J., Jarvis, C.H., Dowers, S., 2003. From stand-alone programs towards grid-aware services and components: a case study in agricultural modelling with interpolated climate data. *Environmental Modelling & Software* 18 (4), 379e 391.
- McIntosh, A. C., Cushing, J. B., Nadkarni, N. M., & Zeman, L. (2007). Database design for ecologists: composing core entities with observations. *Ecological informatics*, 2(3), 224-236.
- MSDN, Retrieved June 30, 2013 from <https://msdn.microsoft.com/>
- NSDL National Science Digital Library (1995), Topics by Science.gov. Retrieved July 30, 2012 from www.science.gov
- NAHRIM (2005), Retrieved July 30, 2012 from www.nahrim.gov.my
- Perbadanan Putrajaya (2006). In Portal Rasmi Perbadanan Putrajaya. Retrieved July 25, 2012, from www.ppj.gov.my
- Perbandanan Putrajaya, 2013 PLWMOS, Accessed on November 25 - <http://plwmos.ppj.gov.my/>
- RAMSAR convention (2013), Retrieved July 30, 2014 from <http://www.ramsar.org>
- Rouchelle R. Dinglasan (2013), NCR's last bird sanctuary makes Ramsar List of globally important wetlands, *GMA News Online*.
- Ryanzhin, S.V., Straskraba, M., Geller, W., 2001. Developing WORLDLAKE: database and GIS for limnological studies. *Proceedings of the Ninth International Conference on the Conservation and Management of Lakes*, Otsu, Japan, 10 – 15 November, 2001. ILEC Publications, pp. 25 – 28.
- RYANZHIN, D. S. V. Development of Global Database and Geographical Information System for World Lakes (GIS WORLDLAKE).
- Santos, X., Brito, J. C., Sillero, N., Pleguezuelos, J. M., Llorente, G. A., Fahd, S., & Parellada, X. (2006). Inferring habitat-suitability areas with ecological modelling techniques and GIS: a contribution to assess the conservation status of *Vipera latastei*. *Biological Conservation*, 130(3), 416-425.
- Shahrizaila Abdullah, Lessons from Malaysia on Developing a Nationwide Strategic Plan for the Management of Lakes and Reservoirs, Integrated Lake Bsinmaangement. Training 2013.
- Sharip, Z., & Jusoh, J. (2010). Integrated lake basin management and its importance for Lake Chini and other lakes in Malaysia. *Lakes & Reservoirs: Research & Management*, 15(1), 41-51.
- Sharip, Z., & Zakaria, S. (2007). Lakes and reservoir in Malaysia: management and research challenges. In *Proceedings of Taal2007: the 12th World lake conference* (Vol. 1349, p. 1355).

- Seligman, L., & Roenthal, A. (2001). XML's impact on databases and data sharing. *Computer*, 34(6), 59-67.
- Stein, B. R., & Wieczorek, J. R. (2004). Mammals of the world: MaNIS as an example of data integration in a distributed network environment. *Biodiversity Informatics*, 1
- Sudip Mitra, Reiner Wassmann and Paul L.G. Vlek(2003) Global Inventory of Wetlands and their Role in the Carbon Cycle, *Discussion Papers on Development Policy Bonn*
- Vos, P., Meelis, E., & Ter Keurs, W. J. (2000). A framework for the design of ecological monitoring programs as a tool for environmental and nature management. *Environmental monitoring and assessment*, 61(3), 317-344.
- Whigham, P. A., & Recknagel, F. (2001). An inductive approach to ecological time series modelling by evolutionary computation. *Ecological Modelling*, 146(1), 275-287.
- Zati S, and Salmah Z: Lakes and Reservoir in Malaysia: Management and Research Challenges. In: *TAAL 2007: The 12 World Lake Conference*, pp 1349-55. Ministry of Environment & Forests, Govt of India, Jaipur, India.

University of Malaya

LIST OF PUBLICATIONS AND PAPERS PRESENTED



Frontiers in Life Science



ISSN: 2155-3769 (Print) 2155-3777 (Online) Journal homepage: <http://www.tandfonline.com/loi/tfls20>

Ecological data prediction and visualization system

Sorayya Malek, Cham Hui, Lau C. Fong, Mogeib A.A. Mosleh, Pozi Milow, Sarinder K. Dhillon & Sharifah M. Syed

To cite this article: Sorayya Malek, Cham Hui, Lau C. Fong, Mogeib A.A. Mosleh, Pozi Milow, Sarinder K. Dhillon & Sharifah M. Syed (2015): Ecological data prediction and visualization system, Frontiers in Life Science, DOI: [10.1080/21553769.2015.1041167](https://doi.org/10.1080/21553769.2015.1041167)

To link to this article: <http://dx.doi.org/10.1080/21553769.2015.1041167>



Published online: 15 May 2015.



Submit your article to this journal [↗](#)



Article views: 70



View related articles [↗](#)



View Crossmark data [↗](#)

Ecological data prediction and visualization system

Sorayya Malek^{a*}, Cham Hui^a, Lau C. Fong^a, Mogeab A.A. Mosleh^b, Pozi Milow^a, Sarinder K. Dhillon^a
and Sharifah M. Syed^c

^a*Institute of Biological Sciences, Faculty of Science, University of Malaya, Kuala Lumpur, Malaysia;* ^b*Artificial Intelligent Department, Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, Malaysia;* ^c*Department of Computer & Communication System Engineering, University of Putra Malaysia, Serdang, Selangor Darul Ehsan, Malaysia*

(Received 31 October 2014; accepted 12 April 2015)

Temporal patterns in ecological data can be visualized and communicated effectively through graphical means. The aim of this study was to develop a data prediction and visualization system based on historical data and thematic map technology to visualize forecast temporal ecological changes. The visualization system consists of prediction and data visualization modules. The prediction module is developed using a hybrid evolutionary algorithm (HEA) to classify and predict noisy ecological data. The visualization module is developed using Dotnet Framework 2.0 to implement thematic cartography for volume visualization. The visualization system is evaluated by its capability in representing the output data on a map, and by predicting the abundance of Chlorophyta based on other water quality parameters. Rules for predicting Chlorophyta abundance had a success rate of almost 90%. The integration of computational data mining using HEA and visualization using thematic maps promises practical solutions and better techniques for forecasting temporal ecological changes, especially when data sets have complex relationships without clear distinction between various variables.

Keywords: data visualization; ecological system; prediction; water quality; HEA

Introduction

Data visualization is important as it enables the communication of information clearly and effectively through graphical means. It is useful for analysing and exploring the huge amounts of ecological data that are an immense source of important information for research and decision making (MacEachren & Kraak 2001). Ecological data contain an array of distinctive features making them valuable, yet challenging to visualize (Helly et al. 1999). The goal of ecological visualization is to combine the strengths of human vision, creativity and general knowledge with the storage capacity and the computational power of modern computers in order to explore large geospatial data sets. One way of undertaking this task is by presenting a multitude of graphic representations of the data to the user, which allows him or her to interact with the data and change the views in order to gain insight and to draw conclusions quickly (Keim et al. 2005). It is also important to be able to visualize the relationships and patterns discovered from complex ecological data.

Effective visual data-mining tools are required to display multivariate and spatiotemporal ecological data to easily perceive patterns and relationships. Common visualization techniques for multivariate data are geometric, glyph or icon-based pixel-oriented and hierarchical methods (Keim & Kriegel 1996; Schroeder 2005). Spatiotemporal data require a map display to visualize the

spatial attributes of the data in order to reveal, analyse and understand patterns (Andrienko et al. 2003, 2005). Thematic maps are an effective method of data visualization and are widely used for the representation of ecological data (Few 2014). A thematic map is designed with a specific theme and topic. It does not contain any physical features such as rivers, roads or subdivisions; it usually uses city locations, country maps, rivers and other geographical locations as its base maps.

Several computational methods such as the hybrid evolutionary algorithm (HEA) and artificial neural network (ANN) have been developed in different areas including machine learning and data mining, which can analyse large data volumes and automatically extract knowledge. HEAs and ANNs have been successfully applied to unravel and predict complex and non-linear ecological data, e.g. algal population dynamics for eutrophication modelling and lake management (Melesse et al. 2008; Welk et al. 2008). Algae respond to a wide range of pollutants and are thus a good indicator of eutrophication. The temporal dynamics of algal communities are influenced by a complex array of biotic and abiotic factors operating through both direct and indirect pathways (Carrillo et al. 1995). ANN models are highly flexible functions that can be used to model non-linear relationships. HEAs can also be used to model non-linear relationships. However, the HEA has the advantage over ANN that it uses genetic programming

*Corresponding author. Email: sorayya@um.edu.my

Framework for Tropical Ecological Data Warehouse (TEDW) FOR Governance and Maintenance of Tropical Lakes

Hui Cham¹, Sorayya Bibi Malek Binti Malek Abd Rashid², Sharifah Mumtazah Syed Ahmad and
Aishah Binti Salleh

Abstract. The current data collection on tropical lakes and wetlands especially in Malaysia is carried out in a disparate manner which does facilitate database integration and data sharing. This can be considered as an inefficient use of available resources. Such an issue is partly due to the lack of proper standards on tropical data warehousing that would otherwise provide the best practices for common platform and coherent efforts in lake managements. Therefore the main objective of this paper is to propose a suitable framework for tropical ecological data warehouse (TEDW) of lake and wetlands water bodies. The design of TEDW is robust as it caters for different ecological objects, formats and variables which are desirable for actual data warehouse implementation. In addition, TEDW also provides a real-time data entry with web interfaces as well as basic statistical analysis and graphical representations. In order to facilitate data sharing with outside parties, TEDW provides a restricted access to authenticated parties. This paper discusses the general scope, data structures and implementation of the TEDW framework with the aim of facilitating the best practices in lake management through monitoring and evaluation of ecological data of tropical water bodies.

Keywords: ecological, data warehouse, data mining, web services

1. Introduction

Jusoh [1] reported that 38% and 62% of the 90 monitored lakes in Malaysia were classified as mesotrophic and eutrophic respectively. To prevent eutrophication, the quality of water source needs to be observed and monitored. Hence proper management of ecological data is imperative for effective lake and wetlands monitoring and governance. However, there are no comprehensive monitoring programs carried out in most lakes in Malaysia, except Putrajaya Lakes [2]. In addition, there is no reported study on an integrated approach for proper database development at a national level [3].

At the international level there are many lake databases developed and maintained such as the World Lake Database [4], Ramsar Site Database [5], and GIS WORLDLAKE database [6]. The World Lake Database consist data on 500 lakes from 73 countries. The data is separated into a few categories which are location, description, physical dimensions, lake water quality, physiographic features, biological features, Socio-economic conditions, lake utilization, deterioration of lake environments and hazards, wastewater treatment, improvement works in the lake, legislative and institutional measures for upgrading lake environments, development plans, and sources of data.

The Ramsar Sites Database meanwhile is an internet accessible database which can generate report for public usage. The database includes the details from the Ramsar Information Sheet, the Ramsar National Report and information provided by contract parties. GIS WORLDLAKE consists of data of geographic, morphometric, hydrological, meteorological and climatological, hydrochemical, hydrothermal, and others.

However the information contained in these databases on tropical lakes in Malaysia are not comprehensive enough.

* Corresponding author. Tel.: + 60127609992.
E-mail address: chamhui@siswa.um.edu.my.

APPENDIX

Appendix A: Information Sheets of Ramsar Wetlands

Conference of the Parties in the Annex to COP9 Resolution IX.6 and provided a report in line with paragraph 28 of that Annex, prior to the submission of an updated RIS.

b) Describe briefly any major changes to the ecological character of the Ramsar site, including in the application of the Criteria, since the previous RIS for the site:

7. Map of site:

Refer to Annex III of the *Explanatory Note and Guidelines*, for detailed guidance on provision of suitable maps, including digital maps.

a) A map of the site, with clearly delineated boundaries, is included as:

i) a hard copy (required for inclusion of site in the Ramsar List): ☐;

ii) an electronic format (e.g. a JPEG or ArcView image) ☐;

iii) a GIS file providing geo-referenced site boundary vectors and attribute tables ☐.

b) Describe briefly the type of boundary delineation applied:

e.g. the boundary is the same as an existing protected area (nature reserve, national park, etc.), or follows a catchment boundary, or follows a geopolitical boundary such as a local government jurisdiction, follows physical boundaries such as roads, follows the shoreline of a waterbody, etc.

8. Geographical coordinates (latitude/longitude, in degrees and minutes):

Provide the coordinates of the approximate centre of the site and/or the limits of the site. If the site is composed of more than one separate area, provide coordinates for each of these areas.

9. General location:

Include in which part of the country and which large administrative region(s) the site lies and the location of the nearest large town.

10. Elevation: (in metres: average and/or maximum & minimum)

11. Area: (in hectares)

12. General overview of the site:

Provide a short paragraph giving a summary description of the principal ecological characteristics and importance of the wetland.

13. Ramsar Criteria:

Tick the box under each Criterion applied to the designation of the Ramsar site. See Annex II of the *Explanatory Notes and Guidelines* for the Criteria and guidelines for their application (adopted by Resolution VII.11). All Criteria which apply should be ticked.

1 • 2 • 3 • 4 • 5 • 6 • 7 8 • 9

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

14. Justification for the application of each Criterion listed in 13 above:

Provide justification for each Criterion in turn, clearly identifying to which Criterion the justification applies (see Annex II for guidance on acceptable forms of justification).

15. Biogeography (required when Criteria 1 and/or 3 and /or certain applications of Criterion 2 are applied to the designation):

Name the relevant biogeographic region that includes the Ramsar site, and identify the biogeographic regionalisation system that has been applied.

a) biogeographic region:

b) biogeographic regionalisation scheme (include reference citation):

16. Physical features of the site:

Describe, as appropriate, the geology, geomorphology; origins - natural or artificial; hydrology; soil type; water quality; water depth, water permanence; fluctuations in water level; tidal variations; downstream area; general climate, etc.

17. Physical features of the catchment area:

Describe the surface area, general geology and geomorphological features, general soil types, and climate (including climate type).

18. Hydrological values:

Describe the functions and values of the wetland in groundwater recharge, flood control, sediment trapping, shoreline stabilization, etc.

19. Wetland Types

a) presence:

Circle or underline the applicable codes for the wetland types of the Ramsar “Classification System for Wetland Type” present in the Ramsar site. Descriptions of each wetland type code are provided in Annex I of the *Explanatory Notes & Guidelines*.

Marine/coastal: A • B • C • D • E • F • G • H • I • J • K •

Zk(a)

Inland: L • M • N • O • P • Q • R • Sp • Ss • Tp Ts • U • Va •

Vt • W • Xf • Xp • Y • Zg • Zk(b)

Human-made: 1 • 2 • 3 • 4 • 5 • 6 • 7 • 8 • 9 • Zk(c)

b) dominance:

List the wetland types identified in a) above in order of their dominance (by area) in the Ramsar site, starting with the wetland type with the largest area.

20. General ecological features:

Provide further description, as appropriate, of the main habitats, vegetation types, plant and animal communities present in the Ramsar site, and the ecosystem services of the site and the benefits derived from them.

21. Noteworthy flora:

Provide additional information on particular species and why they are noteworthy (expanding as necessary on information provided in 14, Justification for the application of the Criteria) indicating, e.g., which species/communities are unique, rare, endangered or biogeographically important, etc. *Do not include here taxonomic lists of species present – these may be supplied as supplementary information to the RIS.*

22. Noteworthy fauna:

Provide additional information on particular species and why they are noteworthy (expanding as necessary on information provided in 14, Justification for the application of the Criteria) indicating, e.g., which species/communities are

unique, rare, endangered or biogeographically important, etc., including count data. *Do not include here taxonomic lists of species present – these may be supplied as supplementary information to the RIS.*

23. Social and cultural values:

a) Describe if the site has any general social and/or cultural values e.g., fisheries production, forestry, religious importance, archaeological sites, social relations with the wetland, etc. Distinguish between historical/archaeological/religious significance and current socio-economic values:

b) Is the site considered of international importance for holding, in addition to relevant ecological values, examples of significant cultural values, whether material or non-material, linked to its origin, conservation and/or ecological functioning?

If Yes, tick the box ☐ and describe this importance under one or more of the following categories:

- i) sites which provide a model of wetland wise use, demonstrating the application of traditional knowledge and methods of management and use that maintain the ecological character of the wetland:
- ii) sites which have exceptional cultural traditions or records of former civilizations that have influenced the ecological character of the wetland:

iii) sites where the ecological character of the wetland depends on the interaction with local communities or indigenous peoples:

iv) sites where relevant non-material values such as sacred sites are present and their existence is strongly linked with the maintenance of the ecological character of the wetland:

24. Land tenure/ownership:

a) within the Ramsar site:

b) in the surrounding area:

25. Current land (including water) use:

a) within the Ramsar site:

b) in the surroundings/catchment:

26. Factors (past, present or potential) adversely affecting the site's ecological character, including changes in land (including water) use and development projects:

a) within the Ramsar site:

b) in the surrounding area:

27. Conservation measures taken:

a) List national and/or international category and legal status of protected areas, including boundary relationships with the Ramsar site:

In particular, if the site is partly or wholly a World Heritage Site and/or a UNESCO Biosphere Reserve, please give the names of the site under these designations.

b) If appropriate, list the IUCN (1994) protected areas category/ies which apply to the site (tick the box or boxes as appropriate):

Ia ☐; Ib ☐; II ☐; III ☐; IV ☐; V ☐; VI ☐

c) Does an officially approved management plan exist; and is it being implemented?:

d) Describe any other current management practices:

28. Conservation measures proposed but not yet implemented:

e.g. management plan in preparation; official proposal as a legally protected area, etc.

29. Current scientific research and facilities:

e.g., details of current research projects, including biodiversity monitoring; existence of a field research station, etc.

30. Current communications, education and public awareness (CEPA) activities related to or benefiting the site:

e.g. visitors' centre, observation hides and nature trails, information booklets, facilities for school visits, etc.

31. Current recreation and tourism:

State if the wetland is used for recreation/tourism; indicate type(s) and their frequency/intensity.

32. Jurisdiction:

Include territorial, e.g. state/region, and functional/sectoral, e.g. Dept of Agriculture/Dept. of Environment, etc.

33. Management authority:

Provide the name and address of the local office(s) of the agency(ies) or organisation(s) directly responsible for managing the wetland. Wherever possible provide also the title and/or name of the person or persons in this office with responsibility for the wetland.

34. Bibliographical references:

Scientific/technical references only. If biogeographic regionalisation scheme applied (see 15 above), list full reference citation for the scheme.

Please return to: **Ramsar Convention Secretariat, Rue Mauverney 28, CH-1196 Gland,
Switzerland**

Telephone: **+41 22 999 0170** • Fax: **+41 22 999 0169** • e-mail: **ramsar@ramsar.org**

Appendix B: Stored Procedure for Star Schema

```
USE [EDW]

GO

/***** Object:      StoredProcedure      [dbo].[EnvAndBioDataGetByDate]
Script Date: 09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvAndBioDataGetByDate]

    @envchk as nvarchar(1000)

    ,@biochk as nvarchar(1000)

as

declare @sqlquery varchar(max);

set @sqlquery = '

    select  ENV_DATA.env_no as pid, result, ENV_HEADER.ENV_DESC as pdesc,
RECORD_DATE      from

    ENV_DATA left join ENV_HEADER on ENV_DATA .ENV_NO = ENV_HEADER .ENV_NO

    where ENV_DATA .ENV_NO in ('+ @envchk +')

union all

    select BIO_DATA .TAX_ID as pid, result, TAXONOMY .tax_desc as pdesc,
RECORD_DATE      from

    BIO_DATA left join TAXONOMY on BIO_DATA .TAX_ID = TAXONOMY .tax_id

    where TAXONOMY .TAX_ID in ('+ @biochk +')';
```

```

exec (@sqlquery)

GO

/***** Object:  StoredProcedure [dbo].[caseinsert]      Script Date:
09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[caseinsert]

@c_name as nvarchar(100),

@c_desc as nvarchar(100),

@c_level as nvarchar(50),

@c_status as nvarchar(50),

@c_suspect as nvarchar(50),

@c_suspectIC as nvarchar(50),

@c_target as nvarchar(50),

@start_date as date,

@record_date as date

as

insert into CCASE (

c_name,

c_desc,

c_level,

c_status,

c_suspect,

c_suspectIC,

```

```

c_target,

start_date,

record_date

)

values (

@c_name ,

@c_desc ,

@c_level ,

@c_status,

@c_suspect,

@c_suspectIC,

@c_target,

@start_date,

@record_date

)

GO

/***** Object:  StoredProcedure [dbo].[WQIMonthly]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[WQIMonthly]

@fromDate as nvarchar(20),

@toDate as nvarchar(20)

```

```

as

begin

SELECT      (DO.SIDO*0.22*0)+( (100.4-(4.23*BOD.SIBOD) ) *0.19)+( ( (-
1.33*COD.SICOD)+99.1) *0.16)+( (100.5-(105*AN.SIAN) ) *0.15)+( (97.5*exp(-
0.573*SS.SISS)-5*SS.SISS-2)      *0.16)+( (-181+(82.4*PH.SIPH)-
(6.05*PH.SIPH*PH.SIPH) ) *0.12)      as      result,      DO.record_month,
DO.record_year from

(select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as SIDO,

month(env_data.record_date) as
record_month, YEAR(env_data.record_date) as record_year
,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

where

env_data.ENV_NO =6

AND

(@fromDate < RECORD_DATE ) and (@toDate > RECORD_DATE )

group by month(env_data.record_date), YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC

) DO

left join

(select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as SIBOD,

month(env_data.record_date) as
record_month, YEAR(env_data.record_date) as record_year
,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

where

env_data.ENV_NO =16

AND

(@fromDate < RECORD_DATE ) and (@toDate > RECORD_DATE )

```

```

    group by month(env_data.record_date), YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC

) BOD on BOD.record_month = DO.record_month and BOD.record_year =
DO.record_year

left join

(select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as SICOD,

    month(env_data.record_date) as
record_month, YEAR(env_data.record_date) as record_year ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

where

env_data.ENV_NO =17

AND

(@fromDate < RECORD_DATE ) and (@toDate > RECORD_DATE )

group by month(env_data.record_date), YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC

) COD on COD.record_month = DO.record_month and COD.record_year =
DO.record_year

left join

(select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as SIAN,

    month(env_data.record_date) as
record_month, YEAR(env_data.record_date) as record_year ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

where

env_data.ENV_NO =13

AND

(@fromDate < RECORD_DATE ) and (@toDate > RECORD_DATE )

group by month(env_data.record_date), YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC

) AN on AN.record_month = DO.record_month and AN.record_year =
DO.record_year

```

```

left join

(select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as SISS,

    month(env_data.record_date) as record_year
record_month, YEAR(env_data.record_date) as record_year,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

where

env_data.ENV_NO =18

AND

(@fromDate < RECORD_DATE ) and (@toDate > RECORD_DATE )

group by month(env_data.record_date), YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC

) SS on SS.record_month = DO.record_month and SS.record_year =
DO.record_year

left join

(select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as SIPH,

    month(env_data.record_date) as record_year
record_month, YEAR(env_data.record_date) as record_year,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

where

env_data.ENV_NO =5

AND

(@fromDate < RECORD_DATE ) and (@toDate > RECORD_DATE )

group by month(env_data.record_date), YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC

) PH on PH.record_month = DO.record_month and PH.record_year =
DO.record_year

```

```

order by DO.record_year, DO.record_month

end

GO

/***** Object:  StoredProcedure [dbo].[TWDGGet]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[TWDGGet]

as

select * from TWDG

GO

/***** Object:  StoredProcedure [dbo].[TaxonomyUpdate]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====

```



```

CREATE PROCEDURE [dbo].[TaxonomyUpdate]

    @tax_id as int,

    @tax_desc as nvarchar(100),

    @tax_rank as int,

    @tax_upper as int

AS

BEGIN

    update TAXONOMY SET

    tax_desc =@tax_desc , tax_rank =@tax_rank , tax_upper = @tax_upper

    where tax_id = @tax_id

END

GO

/***** Object: StoredProcedure [dbo].[TaxonomySelectByID]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

-- =====

-- Author:          <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =====

CREATE PROCEDURE [dbo].[TaxonomySelectByID]

    @tax_id as nvarchar(100)

```

```

AS

BEGIN

    select      TAXONOMY.tax_id,TAXONOMY.tax_desc      ,TAXONOMY.tax_rank
,TAXONOMY.tax_upper,tax2.tax_rank as upper_rank from TAXONOMY

    left join (select tax_id, tax_rank from TAXONOMY)tax2

    on tax2.tax_id = TAXONOMY.tax_upper

    where TAXONOMY.tax_id =@tax_id

END

GO

/***** Object:      StoredProcedure      [dbo].[TaxonomySearchByDesc]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====

CREATE PROCEDURE [dbo].[TaxonomySearchByDesc]

    @tax_desc as nvarchar(100)

AS

BEGIN

```

```

        select top 10 tax.tax_id as ID, tax.tax_desc as Name, tax.tax_rank
as Rank, uppertax.tax_desc as Upper_Rank from

        (select top 10 * from TAXONOMY where tax_desc like '%' + @tax_desc +
'%' order by tax_id desc ) tax

left join

(select tax_id, tax_desc from TAXONOMY) uppertax

on tax.tax_upper = uppertax.tax_id

order by tax.tax_id desc

END

GO

/***** Object:  StoredProcedure [dbo].[TaxonomyInsert]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====

CREATE PROCEDURE [dbo].[TaxonomyInsert]

        @tax_desc as nvarchar(100),

        @tax_rank as int,

        @tax_upper as int

```

```

AS

BEGIN

    insert into TAXONOMY

    (tax_desc, tax_rank , tax_upper)

    values

    (@tax_desc ,

    @tax_rank ,

    @tax_upper)

END

GO

/***** Object:  StoredProcedure [dbo].[taxonomyGetByUpper]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[taxonomyGetByUpper]

@tax_upper as int

as

select * from TAXONOMY where tax_upper =@tax_upper

GO

/***** Object:  StoredProcedure [dbo].[TaxonomyGetByRank]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

```

```

GO

CREATE procedure [dbo].[TaxonomyGetByRank]

@tax_rank as int

as

select tax_id,tax_desc from TAXONOMY where tax_rank = @tax_rank
order by tax_desc

GO

/***** Object: StoredProcedure [dbo].[TaxonomyGetByLatest]    Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[TaxonomyGetByLatest]

as

select top 10 tax.tax_id as ID,tax.tax_desc as Name, tax.tax_rank as
Rank, uppertax.tax_desc as Upper_Rank from

(select top 10 * from TAXONOMY order by tax_id desc)tax

left join

(select tax_id,tax_desc from TAXONOMY)uppertax

on tax.tax_upper = uppertax.tax_id

order by tax.tax_id desc

```

```

GO

/***** Object:  StoredProcedure [dbo].[TaxHeaderGetAll]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[TaxHeaderGetAll]

as

select TAX_ID ,tax_desc  from TAXONOMY order by tax_desc

GO

/***** Object:  StoredProcedure [dbo].[StationUpdate]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[StationUpdate]

@Station_No as int,

@Land_Types as nvarchar(100),

@Station_Desc as nvarchar(100)

as

update STATION set LAND_TYPES = @Land_Types
,Station_Desc=@Station_Desc where station_no = @station_no

GO

```

```

/***** Object:  StoredProcedure [dbo].[stationselectall]      Script
Date: 09/08/2016 12:31:39 *****/

```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create procedure [dbo].[stationselectall]
```

```
as
```

```
select * from STATION order by STATION_DESC
```

```
GO
```

```

/***** Object:  StoredProcedure [dbo].[StationInsert]      Script Date:
09/08/2016 12:31:39 *****/

```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE procedure [dbo].[StationInsert]
```

```
@station_desc as nvarchar(100),
```

```
@lake_no as int,
```

```
@land_types as nvarchar(100)
```

```
as
```

```
insert into STATION(STATION_DESC, LAKE_NO ,LAND_TYPES )
```

```
values (@station_desc ,@lake_no ,@land_types )
```

```
GO
```

```

/***** Object:  StoredProcedure [dbo].[StationGetByStationNo]
Script Date: 09/08/2016 12:31:39 *****/

```

```

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[StationGetByStationNo]

@Station_No as int

as

select STATION_NO, STATION_DESC, LAND_TYPES from STATION where
STATION_NO = @Station_No

GO

/***** Object: StoredProcedure [dbo].[StationGetByLake]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[StationGetByLake]

@Lake_No as int

as

select STATION_NO, STATION_DESC from STATION order by STATION_DESC

GO

/***** Object: StoredProcedure [dbo].[StationGetAll]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

```



```

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[StationGetAll]

@Lake_No as int

as

select STATION_NO, STATION_DESC from STATION order by STATION_DESC

GO

/***** Object:  StoredProcedure [dbo].[StationDelete]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[StationDelete]

@station_no as int

as

delete from STATION where STATION_NO = @station_no

GO

/***** Object:  StoredProcedure [dbo].[sptestng]          Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

```

```

GO

create procedure [dbo].[sptestng]

AS

select sunshine from rawdata where sunshine is not null

GO

/***** Object:  StoredProcedure [dbo].[selectall]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[selectall]

as

select * from env_header

GO

/***** Object:  StoredProcedure [dbo].[ReportEnvData]    Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

/***** Script for SelectTopNRows command from SSMS *****/

create procedure [dbo].[ReportEnvData]

as

select env_data.env_no,ENV_DESC, result, record_date from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

```

```

where RECORD_DATE between '1/1/2005' and '1/1/2011'

order by ENV_DESC ,RECORD_DATE

GO

/***** Object:  StoredProcedure [dbo].[rainfallvstime]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[rainfallvstime]

as

select result,MONTH(record_date) as record_month, YEAR(record_date) as
record_year,record_date from env_data where station_no='3' and
env_no='1' order by record_date

GO

/***** Object:  StoredProcedure [dbo].[LakeSearchByDesc]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[LakeSearchByDesc]

@Desc as nvarchar(100)

as

select lake_no as No,lake_desc as Name, Country from LAKE where
LAKE_DESC like '%' + @Desc + '%'

```

```

GO

/***** Object:      StoredProcedure [dbo].[LakeSearchByCountryOrDesc]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[LakeSearchByCountryOrDesc]

@Desc as nvarchar(100),

@Country as nvarchar(100)

as

select lake_no as No, lake_desc as Name, Country from LAKE where
country = @Country or LAKE_DESC = @Desc

GO

/***** Object:      StoredProcedure [dbo].[LakeSearchByCountryAndDesc]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[LakeSearchByCountryAndDesc]

@Desc as nvarchar(100),

@Country as nvarchar(100)

```

as

```
select lake_no as No,lake_desc as Name, Country from LAKE where  
country = @Country and LAKE_DESC = @Desc
```

GO

```
/****** Object: StoredProcedure [dbo].[LakeSearchByCountry]    Script  
Date: 09/08/2016 12:31:39 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE procedure [dbo].[LakeSearchByCountry]
```

```
@Country as nvarchar(100)
```

as

```
select lake_no as No,lake_desc as Name, Country from LAKE where  
country = @Country
```

GO

```
/****** Object: StoredProcedure [dbo].[lakeinsert]           Script Date:  
09/08/2016 12:31:39 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE procedure [dbo].[lakeinsert]
```

```
@lake_desc as nvarchar(100),
```

```

@country as nvarchar(100),
@region as nvarchar(100),
@subregion as nvarchar(100),
@coordDG as nvarchar(100),
@coordLL as nvarchar(100),
@sitearea as nvarchar(100),
@wetlandarea as nvarchar(100),
@minelev as numeric(18,3),
@maxelev as numeric(18,3),
@transboundary as nvarchar(100),
@No_separate_unit as int,
@Administrative_region as nvarchar(100),
@LandUse as text,
@LandThreats as text,
@Social_Culture_values as text,
@management_plan_status as text,
@international_conservation_designation as text,
@national_conservation_designation as text,
@wetland_category as nvarchar(100),
@biological_values as text,
@boundary_changes as nvarchar(100)

```

as

```

insert into lake(
lake_desc,
country,
region,
subregion,

```

```

coordDG,

coordLL,

sitearea,

wetlandarea,

minelev ,

maxelev ,

transboundary,

No_separate_unit,

Administrative_region,

LandUse,

LandThreats,

Social_Culture_values,

management_plan_status,

international_conservation_designation,

national_conservation_designation ,

wetland_category ,

biological_values,

boundary_changes

) values (@lake_desc,

@country ,

@region ,

@subregion,

@coordDG ,

@coordLL ,

@sitearea ,

@wetlandarea ,

@minelev ,

@maxelev ,

@transboundary ,

```

```

@No_separate_unit,

@Administrative_region,

@LandUse ,

@LandThreats,

@Social_Culture_values,

@management_plan_status ,

@international_conservation_designation ,

@national_conservation_designation ,

@wetland_category ,

@biological_values

,@boundary_changes

)

GO

/***** Object:  StoredProcedure [dbo].[LakeGetByLakeNo]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[LakeGetByLakeNo]

@Lake_No as int

as

SELECT

lake_desc,

country,

region,

subregion,

coordDG,

coordLL,

```



```

sitearea,

wetlandarea,

minelev ,

maxelev ,

transboundary,

boundary_changes,

No_separate_unit,

Administrative_region,

LandUse,

LandThreats,

Social_Culture_values,

management_plan_status,

international_conservation_designation,

national_conservation_designation ,

wetland_category ,

biological_values

```

```

FROM [EDW].[dbo].[LAKE]

```

```

where LAKE_NO = @Lake_No

```

```

order by lake_desc

```

```

GO

```

```

/***** Object:      StoredProcedure [dbo].[LakeGet]      Script Date:
09/08/2016 12:31:39 *****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

create procedure [dbo].[LakeGet]

as

SELECT    [LAKE_NO]

          , [LAKE_DESC]

FROM      [EDW].[dbo].[LAKE]

order by lake_desc

GO

/***** Object:  StoredProcedure [dbo].[lakeEdit]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[lakeEdit]

@lake_no as int,

@lake_desc as nvarchar(100),

@country as nvarchar(100),

@region as nvarchar(100),

@subregion as nvarchar(100),

@coordDG as nvarchar(100),

@coordLL as nvarchar(100),

@sitearea as nvarchar(100),

@wetlandarea as nvarchar(100),

@minelev as numeric(18,3),

@maxelev as numeric(18,3),

@transboundary as nvarchar(100),

@boundary_changes as nvarchar(100),

@No_separate_unit as int,

```

```

@Administrative_region as nvarchar(100),

@LandUse as text,

@LandThreats as text,

@Social_Culture_values as text,

@management_plan_status as text,

@international_conservation_designation as text,

@national_conservation_designation as text,

@wetland_category as nvarchar(100),

@biological_values as text

```

```

as

```

```

update lake set

```

```

lake_desc= @lake_desc,

country= @country,

region = @region,

subregion= @subregion ,

coordDG = @coordDG ,

coordLL = @coordLL ,

sitearea = @sitearea ,

wetlandarea = @wetlandarea ,

minelev = @minelev ,

maxelev = @maxelev ,

transboundary = @transboundary ,

No_separate_unit = @No_separate_unit ,

Administrative_region= @Administrative_region ,

LandUse = @LandUse ,

LandThreats = @LandThreats ,

Social_Culture_values = @Social_Culture_values ,

```

```

management_plan_status = @management_plan_status ,

international_conservation_designation
=@international_conservation_designation ,

national_conservation_designation
=@national_conservation_designation ,

wetland_category =@wetland_category ,

biological_values =@biological_values ,

boundary_changes = @boundary_changes

where (lake_no = @lake_no)

GO

/***** Object:  StoredProcedure [dbo].[LakeDelete]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[LakeDelete]

@LAKE_NO as int

as

DELETE FROM LAKE

WHERE LAKE_NO = @LAKE_NO

GO

/***** Object:  StoredProcedure [dbo].[blablabla]      Script Date:
09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[blablabla]

```

```

@lakeno as nvarchar(100),

@lake_desc as nvarchar(100)

as

select * from LAKE where LAKE_NO Like '%' + @lakeno + '%'

GO

/***** Object: StoredProcedure [dbo].[BioDataSearch]    Script Date:
09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[BioDataSearch]

@station_no as int,

@record_date as nvarchar(100),

@tax_ID as int

as

select

bio_data.tax_id

, [RESULT]

, [RECORD_DATE]

, [UPDATE_DATE]

```

```

, [INSERT_DATE]

,bio_data.STATION_NO

,STATION.STATION_DESC

,TAXONOMY .tax_desc

FROM [EDW].[dbo].[BIO_DATA]

left join TAXONOMY on TAXONOMY.tax_id = BIO_DATA .TAX_ID

left join STATION on STATION .STATION_NO = BIO_DATA .STATION_NO

where

BIO_DATA.TAX_ID = @tax_ID and BIO_DATA .RECORD_DATE = @record_date
and BIO_DATA .STATION_NO =@STATION_NO

GO

/***** Object:  StoredProcedure [dbo].[BioDataInsert]    Script Date:
09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====

create PROCEDURE [dbo].[BioDataInsert]

    @tax_id as int,

    @result as int,

    @station_no as int,

    @recorddate as nvarchar(20),

    @insertdate as nvarchar(20),

    @updatedate as nvarchar(20)

```

```

AS

BEGIN

    insert into BIO_DATA

        (tax_id ,result,STATION_NO,RECORD_DATE ,INSERT_DATE,UPDATE_DATE)

    values

        (@tax_id ,

        @result ,

        @station_no ,

        @recorddate,

        @insertdate ,

        @updatedate )

END

GO

/***** Object:  StoredProcedure [dbo].[BioDataEdit]      Script Date:
09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[BioDataEdit]

    @Result as int,

    @Update_date as nvarchar(100),

    @station_no as int,

    @record_date as nvarchar(100),

    @tax_ID as int

as

update Bio_Data

set

```

```

result = @result,

update_date= @update_date

where station_no = @station_no

and record_date = @record_date

and TAX_ID = @tax_ID

GO

/***** Object: StoredProcedure [dbo].[algaequantity]    Script Date:
09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[algaequantity]

as

select distinct algae.RECORD_DATE,* from

(select result as algae_quantity, record_date from algae_data where
station_no='CW' and algae_data_grp='24')algae

left join

(select result as temp, record_date from ENV_DATA where
STATION_NO='cw' and ENV_NO='4')temp

on algae.RECORD_DATE =temp.RECORD_DATE

left join

(select result as pH, record_date from ENV_DATA where STATION_NO='cw'
and ENV_NO='5')pH

on algae.RECORD_DATE =pH.RECORD_DATE and pH.RECORD_DATE
=temp.RECORD_DATE

order by algae.record_date

```



```

GO

/***** Object:  StoredProcedure [dbo].[envheaderselectall]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[envheaderselectall]

as

select * from ENV_HEADER order by ENV_NO

GO

/***** Object:  StoredProcedure [dbo].[envheaderInsert]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[envheaderInsert]

@EnvDesc as nvarchar(100),

@EnvType as nvarchar(50),

@EnvQuantifier as nvarchar(100)

as

insert into ENV_HEADER (ENV_NO, ENV_DESC ,Env_Type, ENV_QUANTIFIER)
values (((Select max(env_no) from env_header)+1),@EnvDesc
,@EnvType,@EnvQuantifier)

GO

```

```

/***** Object:      StoredProcedure [dbo].[envheaderhydroselectall]
Script Date: 09/08/2016 12:31:39 *****/

```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create procedure [dbo].[envheaderhydroselectall]
```

```
as
```

```
select * from ENV_HEADER where env_type = 'hydro' order by ENV_NO
```

```
GO
```

```

/***** Object:      StoredProcedure [dbo].[EnvHeaderGetAll]      Script
Date: 09/08/2016 12:31:39 *****/

```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create procedure [dbo].[EnvHeaderGetAll]
```

```
as
```

```
select ENV_NO, ENV_DESC from ENV_HEADER order by ENV_DESC
```

```
GO
```

```

/***** Object:      StoredProcedure [dbo].[EnvHeaderGet]      Script Date:
09/08/2016 12:31:39 *****/

```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```

create procedure [dbo].[EnvHeaderGet]

as


select ENV_NO, ENV_DESC from ENV_HEADER order by ENV_DESC

GO

/***** Object:  StoredProcedure [dbo].[EnvHeaderDelete]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[EnvHeaderDelete]

@envno as int

as


delete from ENV_HEADER where ENV_NO =@envno

GO

/***** Object:  StoredProcedure [dbo].[envheaderchemicoselectall]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[envheaderchemicoselectall]

as

```

```

select * from ENV_HEADER where env_type ='chemico' order by ENV_NO

GO

/***** Object:  StoredProcedure [dbo].[envheaderByEnvType]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[envheaderByEnvType]

@EnvType as nvarchar(50)

as

select * from ENV_HEADER where env_type =@EnvType order by ENV_DESC

GO

/***** Object:  StoredProcedure [dbo].[EnvGraphGet]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvGraphGet]

@FromDate as nvarchar(100),

@ToDate as nvarchar(100),

@Env_No as int,

@Station_No as int

as

```

```

select RESULT, RECORD_DATE from ENV_DATA where

Env_NO = @Env_No and

Station_No = @Station_No and

(Record_date between @FromDate and @ToDate)

GO

/*****          Object:          StoredProcedure
[dbo].[EnvGetByDateAndMultiStationYearly]      Script Date: 09/08/2016
12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[EnvGetByDateAndMultiStationYearly]

@EnvNo as nvarchar(3),

@station1 as nvarchar(3),

@station2 as nvarchar(3),

@station3 as nvarchar(3),

@fromDate as nvarchar(20),

@toDate as nvarchar(20)

as

begin

```

```

select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as result,

    STATION.STATION_DESC as station_no, YEAR(env_data.record_date) as
record_year , ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from
env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO


where

    (env_data.env_no= @EnvNo) and ((env_data.station_no = @station1)
or (env_data.station_no = @station2) or (env_data.station_no = @station3))

    and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
env_data.RECORD_DATE )


group by STATION.STATION_DESC, YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC


order by YEAR(env_data.record_date)


end

GO

/***** Object: StoredProcedure
[dbo].[EnvGetByDateAndMultiStationMonthly] Script Date: 09/08/2016
12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvGetByDateAndMultiStationMonthly]

```

```

@EnvNo as nvarchar(3),

@station1 as nvarchar(3),

@station2 as nvarchar(3),

@station3 as nvarchar(3),

@fromDate as nvarchar(20),

@toDate as nvarchar(20)

as

begin

select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as result,

    STATION.STATION_DESC as station_no, month(env_data.record_date) as
record_month, YEAR(env_data.record_date) as record_year ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO


where

    (env_data.env_no= @EnvNo) and ((env_data.station_no = @station1)
or (env_data.station_no = @station2) or (env_data.station_no = @station3))

```

```

        and      (@fromDate    <    env_data.RECORD_DATE    )    and      (@toDate    >
env_data.RECORD_DATE )

group          by          STATION.STATION_DESC
month(env_data.record_date), YEAR(env_data.record_date)
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC

order by YEAR(env_data.record_date), month(env_data.record_date)

end

GO

/***** Object:  StoredProcedure [dbo].[EnvGetByDateAndMultiStation]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvGetByDateAndMultiStation]

@EnvNo as nvarchar(3),

@station1 as nvarchar(3),
@station2 as nvarchar(3),
@station3 as nvarchar(3),

@fromDate as nvarchar(20),

@toDate as nvarchar(20)

```



```

as

begin

    select  env_data.result,      STATION.STATION_DESC      as  station_no,
env_data.record_date,  ENV_HEADER.ENV_QUANTIFIER,  ENV_HEADER.ENV_DESC
from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO


where

    (env_data.env_no= @EnvNo)      and  ((env_data.station_no = @station1)
or(env_data.station_no = @station2) or(env_data.station_no = @station3))

    and  (@fromDate  <  env_data.RECORD_DATE  )      and  (@toDate  >
env_data.RECORD_DATE  )

    order by record_date

end

GO

/***** Object:      StoredProcedure [dbo].[EnvGetByDateAndMultiEnvNo]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvGetByDateAndMultiEnvNo]

@EnvNo1 as nvarchar(3),

@EnvNo2 as nvarchar(3),

@EnvNo3 as nvarchar(3),

@station as nvarchar(3),

```

```

@fromDate as nvarchar(20),

@toDate as nvarchar(20)

as

begin

(select

    (env_data.result-(select min(env_data.result) from env_data where
    (env_data.env_NO = @EnvNo1) and (@fromDate < env_data.RECORD_DATE ) and
    (@toDate > env_data.RECORD_DATE )and (ENV_DATA.STATION_NO=@station )))

    /((select max(env_data.result) from env_data where (env_data.env_NO =
    @EnvNo1) and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
    env_data.RECORD_DATE ) and (ENV_DATA.STATION_NO=@station ))-(select
    min(env_data.result) from env_data where (env_data.env_NO = @EnvNo1) and
    (@fromDate < env_data.RECORD_DATE ) and (@toDate > env_data.RECORD_DATE
    ) and (ENV_DATA.STATION_NO=@station ))) as result

    , env_data.station_no, env_data.record_date, ENV_HEADER.ENV_DESC
from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO

where

    (env_data.env_no= @EnvNo1) and (env_data.station_no = @station)

    and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
    env_data.RECORD_DATE )

)

union all

(select

```

```

        (env_data.result-(select min(env_data.result) from env_data where
(env_data.env_NO = @EnvNo2) and (@fromDate < env_data.RECORD_DATE ) and
(@toDate > env_data.RECORD_DATE )and (ENV_DATA.STATION_NO=@station )) )

        /((select max(env_data.result) from env_data where (env_data.env_NO =
@EnvNo2) and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
env_data.RECORD_DATE )and (ENV_DATA.STATION_NO=@station ))-(select
min(env_data.result) from env_data where (env_data.env_NO = @EnvNo2) and
(@fromDate < env_data.RECORD_DATE ) and (@toDate > env_data.RECORD_DATE
)and (ENV_DATA.STATION_NO=@station ))) as result

        , env_data.station_no, env_data.record_date, ENV_HEADER.ENV_DESC
from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO

where

(env_data.env_no= @EnvNo2) and (env_data.station_no = @station)

and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
env_data.RECORD_DATE )

)

union all

(select (env_data.result-(select min(env_data.result) from env_data
where (env_data.env_NO = @EnvNo3) and (@fromDate < env_data.RECORD_DATE
) and (@toDate > env_data.RECORD_DATE )and (ENV_DATA.STATION_NO=@station
))) )

        /((select max(env_data.result) from env_data where (env_data.env_NO =
@EnvNo3) and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
env_data.RECORD_DATE )and (ENV_DATA.STATION_NO=@station ))-(select
min(env_data.result) from env_data where (env_data.env_NO = @EnvNo3) and
(@fromDate < env_data.RECORD_DATE ) and (@toDate > env_data.RECORD_DATE
)and (ENV_DATA.STATION_NO=@station ))) as result

        , env_data.station_no, env_data.record_date, ENV_HEADER.ENV_DESC
from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO

where

(env_data.env_no= @EnvNo3) and (env_data.station_no = @station)

```

```

        and      (@fromDate <  env_data.RECORD_DATE      )      and      (@toDate >
env_data.RECORD_DATE )

    )

    order by RECORD_DATE

end

GO

/***** Object:  StoredProcedure [dbo].[EnvGetByDateAndEnvNoYearly]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[EnvGetByDateAndEnvNoYearly]

@EnvNo as nvarchar(3),

@station as nvarchar(3),

@fromDate as nvarchar(20),

@toDate as nvarchar(20)

as

begin

select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as result,

env_data.station_no, YEAR(env_data.record_date) as record_year ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

```

```

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO


where

(env_data.env_no= @EnvNo) and (env_data.station_no = @station)

and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
env_data.RECORD_DATE )


group by env_data.station_no, YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC


order by YEAR(env_data.record_date)


end

GO

/***** Object:  StoredProcedure [dbo].[EnvGetByDateAndEnvNoMonthly]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvGetByDateAndEnvNoMonthly]

@EnvNo as nvarchar(3),

@station as nvarchar(3),

@fromDate as nvarchar(20),

```

```

@toDate as nvarchar(20)

as

begin

select CAST(AVG(1. * env_data.RESULT) AS numeric(12, 2)) as result,

env_data.station_no,          month(env_data.record_date) as
record_month, YEAR(env_data.record_date) as record_year ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO


where

(env_data.env_no= @EnvNo) and (env_data.station_no = @station)

and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
env_data.RECORD_DATE )


group by env_data.station_no,
month(env_data.record_date), YEAR(env_data.record_date) ,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC

order by YEAR(env_data.record_date), month(env_data.record_date)

end

```

```

GO

/***** Object:      StoredProcedure      [dbo].[EnvGetByDateAndEnvNo]
Script Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvGetByDateAndEnvNo]

@EnvNo as nvarchar(3),

@station as nvarchar(3),

@fromDate as nvarchar(20),

@toDate as nvarchar(20)

as

begin

select env_data.result, env_data.station_no, env_data.record_date,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO

where

(env_data.env_no= @EnvNo) and (env_data.station_no = @station)

and (@fromDate < env_data.RECORD_DATE ) and (@toDate >
env_data.RECORD_DATE )

```

```

order by record_date

end

GO

/***** Object:  StoredProcedure [dbo].[EnvGetByDate]    Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvGetByDate]

@fromDate as nvarchar(20),

@toDate as nvarchar(20)

as

begin

select env_data.result, env_data.station_no, env_data.record_date,
ENV_HEADER.ENV_QUANTIFIER, ENV_HEADER.ENV_DESC, STATION.STATION_DESC,
LAKE.LAKE_DESC      from env_data

left join ENV_HEADER on ENV_HEADER .ENV_NO = ENV_DATA .ENV_NO

left join STATION on STATION .STATION_NO = ENV_DATA .STATION_NO

left join LAKE on LAKE.LAKE_NO = STATION .LAKE_NO

where

(@fromDate < env_data.RECORD_DATE ) and (@toDate >
env_data.RECORD_DATE )

```



```

order by record_date

end

GO

/***** Object:  StoredProcedure [dbo].[envDescByEnvType]      Script
Date: 09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[envDescByEnvType]

@EnvType as nvarchar(50)

as

select ENV_DESC,ENV_NO from ENV_HEADER where env_type =@EnvType
order by ENV_DESC

GO

/***** Object:  StoredProcedure [dbo].[EnvDataUpdate]      Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

-- =====

-- Author:          <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =====

create PROCEDURE [dbo].[EnvDataUpdate]

```

```

        @envNo as nvarchar(50),

        @result as float,

        @station as nvarchar(50),

        @recorddate as nvarchar(20),

        @updatedate as nvarchar(20)

AS

BEGIN

    update ENV_DATA SET

        RESULT=@result,UPDATE_DATE=@updatedate

        where Station_No = @station and ENV_NO=@envNo and
RECORD_DATE=@recorddate

END

GO

/***** Object: StoredProcedure [dbo].[EnvDataSearch]    Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[EnvDataSearch]

        @station as nvarchar(50),

        @recorddate as datetime

AS

select result from env_data

```

```

GO

/***** Object:  StoredProcedure [dbo].[EnvDataInsert]    Script Date:
09/08/2016 12:31:39 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====

CREATE PROCEDURE [dbo].[EnvDataInsert]

    @envNo as nvarchar(50),

    @result as float,

    @station as nvarchar(50),

    @recorddate as nvarchar(20),

    @insertdate as nvarchar(20),

    @updatedate as nvarchar(20)

AS

BEGIN

    insert into ENV_DATA

    (ENV_NO ,RESULT,STATION_NO,RECORD_DATE ,INSERT_DATE,UPDATE_DATE)

    values

    (@envNo ,

    @result ,

    @station ,

    @recorddate,

    @insertdate ,

```

```

        @updatedate )

END

GO

/***** Object:  StoredProcedure [dbo].[EnvDataGetLatest]      Script
Date: 09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

--          Batch          submitted          through          debugger:
SQLQuery1.sql|0|0|C:\Users\Administrator\AppData\Local\Temp\2\~vsDF51.
sql

create PROCEDURE [dbo].[EnvDataGetLatest]

@RECORD_DATE as nvarchar(50),

@ENV_NO as int

AS

SELECT

EDATA.RESULT,

EDATE.STATION_NO

FROM

(
SELECT DISTINCT STATION_NO, MAX(RECORD_DATE) AS RECORD_DATE, ENV_NO
FROM ENV_DATA

WHERE ENV_NO=@ENV_NO AND RECORD_DATE<@RECORD_DATE

GROUP BY STATION_NO, ENV_NO) EDATE

LEFT JOIN

(SELECT RESULT, STATION_NO, RECORD_DATE, ENV_NO FROM ENV_DATA) EDATA

```

```
ON EDATA.STATION_NO =EDATE.STATION_NO AND EDATA.RECORD_DATE =  
EDATE.RECORD_DATE AND EDATA.ENV_NO =EDATE.ENV_NO
```

```
GO
```

```
/****** Object:      StoredProcedure      [dbo].[EnvDataGetByRecordDate]  
Script Date: 09/08/2016 12:31:38 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE procedure [dbo].[EnvDataGetByRecordDate]
```

```
@Recorddate as nvarchar(50),
```

```
@StationNo as nvarchar(20),
```

```
@EnvType as nvarchar(50)
```

```
AS
```

```
select
```

```
env_data.ENV_NO as ENV_NO,
```

```
env_header.ENV_DESC as ENV_DESC,
```

```
env_data.STATION_NO as STATION_NO,
```

```
env_data.RESULT as RESULT,
```

```
env_data.INSERT_DATE as INSERT_DATE,
```

```
env_data.RECORD_DATE as RECORD_DATE,
```

```
env_data.UPDATE_DATE as UPDATE_DATE,
```

```
env_header.env_quantifier as ENV_QUANTIFIER
```

```
from
```

```
(select * from env_data) as env_data
```

```

left join (select env_no, env_desc,env_type,env_QUANTIFIER from
ENV_HEADER ) as env_header

on env_data.env_no = env_header.env_no

Where env_data.STATION_NO = @StationNo

and env_data.RECORD_DATE= @Recorddate

and env_header.env_type = @EnvType

order by env_header.ENV_NO

GO

/*****          Object:          StoredProcedure
[dbo].[EnvDataGetByDurationAndStation]      Script Date: 09/08/2016
12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE procedure [dbo].[EnvDataGetByDurationAndStation]

@startdate as nvarchar(100) ,
@enddate as nvarchar(100),
@station as int,
@envHeader as int

as

select * from

(select distinct STATION_NO,result, record_date, env_no from ENV_DATA

where RECORD_DATE between @startdate and @enddate and ENV_NO
=@envHeader and STATION_NO=@station)ENV_DATA

left join

```

```

(select station_no, station_desc from STATION) station on
station.STATION_NO = ENV_DATA.STATION_NO

left join

(select env_no, env_desc, env_quantifier from ENV_HEADER ) env_header
on
env_header.ENV_NO = ENV_DATA .ENV_NO

order by RECORD_DATE , STATION_DESC

GO

/***** Object:      StoredProcedure      [dbo].[EnvDataGetByDuration]
Script Date: 09/08/2016 12:31:38 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

create procedure [dbo].[EnvDataGetByDuration]

@startdate as nvarchar(100) ,

@enddate as nvarchar(100),

@envHeader as int

as

select * from

(select result, record_date, STATION_NO from ENV_DATA

where RECORD_DATE between '2001-01-1' and '2009-01-01') ENV_DATA

GO

```

```

/***** Object:  StoredProcedure [dbo].[CountryGet]      Script Date:
09/08/2016 12:31:38 *****/

```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create procedure [dbo].[CountryGet]
```

```
as
```

```
select distinct country from LAKE
```

```
GO
```

University of Malaya