Durham E-Theses

Computer-Aided System for Wind Turbine Data Analysis

Chen, Bindi

Computer-Aided System for Wind Turbine Data Analysis

Student Name: Bindi Chen

Supervisor Name: Dr. William Song

Submitted as part of the degree of **MSc in Internet System & E-Business** to the

Board of Examiners in the Department of Computer Science, Durham University.

Abstract

Context: The current work on wind turbine failure detection focuses on researching suitable signal processing algorithms and developing efficient diagnosis algorithms. The laboratory research would involve large and complex data, and it can be a daunting task.

Aims: To develop a Computer-Aided system for assisting experts to conduct an efficient laboratory research on wind turbine data analysis. System is expected to provide data visualization, data manipulation, massive data processing and wind turbine failure detection.

Method: 50G off-line SCADA data and 4 confident diagnosis algorithms were used in this project. Apart from the instructions from supervisor, this project also gained help from two experts from Engineering Department. Java and Microsoft SQL database were used to develop the system.

Results: Data visualization provided 6 different charting solutions and together with robust user interactions. 4 failure diagnosis solutions and data manipulations were provided in the system. In addition, dedicated database server and Matlab API with Java RMI were used to resolve the massive data processing problem.

Conclusions: Almost all of the deliverables were completed. Friendly GUI and useful functionalities make user feel more comfortable. The final product does enable experts to conduct an efficient laboratory research. The end of this project also gave some potential extensions of the system.

Keywords – Wind Energy, Wind Turbine, Failure Detection Systems, SCADA system, CM system, Data Visualization

I. INTRODUCTION

Wind energy has gained much attention due to the maturity of the technology and the recent energy crisis. In order to harvest wind energy more efficiently, the size of wind turbines has become physically larger and the locations of the wind farm are usually built at remote mountainous or rough sea regions, this makes maintenance and repair works difficult (Ciang, C.C. *et al*, 2008). Therefore, *Failure Detection Systems* (*FDS*) are required to monitor the wind turbine and provide predictive maintenance. According to P. Caselitz *et al* (1994), the development of efficient *FDS* for wind turbine has to deal with three main issues:

1. The choice of operational data: *Signals* and *Sampling Rates*.
2. Research on suitable signal processing algorithms.
3. Development of efficient classification and diagnosis algorithms.

Operational data clearly reflect the health status of wind turbine; it is already routinely

acquired by the turbine's SCADA (**S**upervisory **C**ontrol **a**nd **D**ata **A**cquisition) system or CM (**C**ondition **M**onitoring) system. The major part of current work is directed to research on suitable signal processing algorithms and develop efficient diagnosis algorithms (Laszlo M. *et al*, 2009). In March 2008, the European Commission setup a project called "ReliaWind" to conduct these researches and Durham University is one of the partners (From ReliaWind, accessed on 05 Jul 2010). The Durham University currently have 1/500 data of this project, which is about 50 gigabyte SCADA data and it consists of three different data format: *SQL database*, *Access database*, and *Office Excel*. Thus, in order to conduct an efficient laboratory research, the experts are expecting a system to resolve the following problems:

1. *Massive Data Processing:* Operational data is acquired on a regular basis (e.g. SCADA data is every 10 minutes, CM data is 10 kHz) and each wind farm consists of hundreds of wind turbines. Thus, massive data processing is a challenge.
2. *Data Visualization:* Display data with human-friendly style is the most convenient way for expert to conduct the research of data analysis.
3. *Data Manipulation:* Different kinds of data format increase the difficulty of data analysis.
4. *Wind Turbine Failure Detection:* Be able to analyse the wind turbine data and provide the predictive maintenance.

A. *Objective and Deliverables*

The project aims to develop a new Computer-Aided system for assisting experts to conduct an efficient laboratory research on wind turbine data analysis. This project will focus on analysing off-line data, which is the 50 gigabyte SCADA data provided by Engineering Department, Durham University. The system should be able to process large size of data as operational data is acquired on a regular basis (e.g. SCADA data is every 10 minutes, CM data is 10 kHz) and each wind farm consists of hundreds of wind turbines. It should display data with human-friendly style and in the most convenient way for expert to conduct the research of data analysis. It should further provide data manipulation (e.g. import/export data) to enable expert to work with different data format and use failure diagnosis algorithms to analyse the wind turbine data and provide the predictive maintenance. Thus, this project should help interpret the wind farm data in the most simplest and fastest way taking into consideration that large amount of data is processed and provide wind turbine failure detection. Apart from the instructions from supervisor, this project also gain help from two experts from Engineering Department, Dr. Yanhui Feng and Dr. Yingning Qiu. The system evaluation will also invite them to evaluate the success of this project.

Project deliverables are divided into three levels: *Basic*, *Intermediate*, and *Advanced*. The details of each level are listed in Table 1 below:

TABLE 1: EXPECTED DELIVERABLES

Level	Deliverables
Basic	1. Database connection (Data come from database).
	2. Data visualization e.g. Line Plot or Scatter Plot.
	3. Basic user interactions e.g. Zoom in/out, chart properties setting, tooltips.
Intermediate	1. Support multiple data display.
	2. Data manipulation e.g. Import/Export data.
	3. Advanced user interaction e.g. save current view status.
Advanced	1. Wind turbine failure detection
	2. Mass data processing
	3. Web-based display

B. Project Plan

The project plan can be divided into two stages. The first stage is preparation, which start in March and last until the end of May. This stage can be described as choosing a project topic and studying project background. The second stage is project development. It contains the project implementation and writing project report. Below two Gantt charts describe the second stage in further details. Figure 1 show the Project Plan based on task. Figure 2 show the Project Plan based on deliverables.
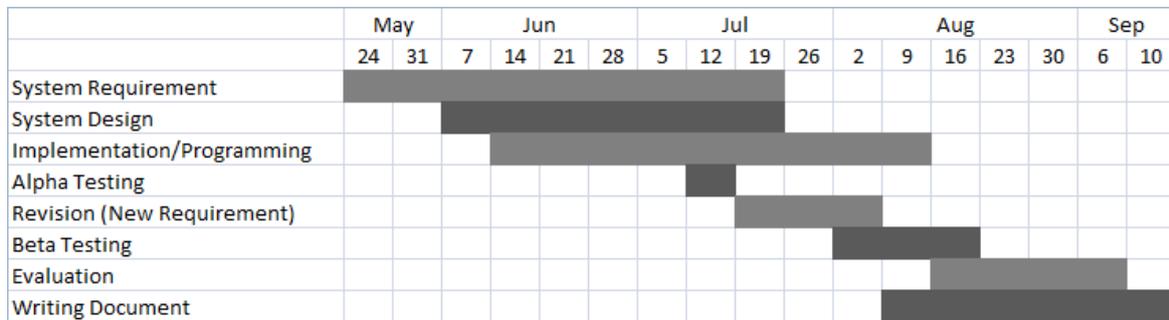
	May		Jun				Jul				Aug				Sep	
	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	6	10
System Requirement																	
System Design																	
Implementation/Programming																	
Alpha Testing																	
Revision (New Requirement)																	
Beta Testing																	
Evaluation																	
Writing Document																	

Figure 1. Project Task Plan

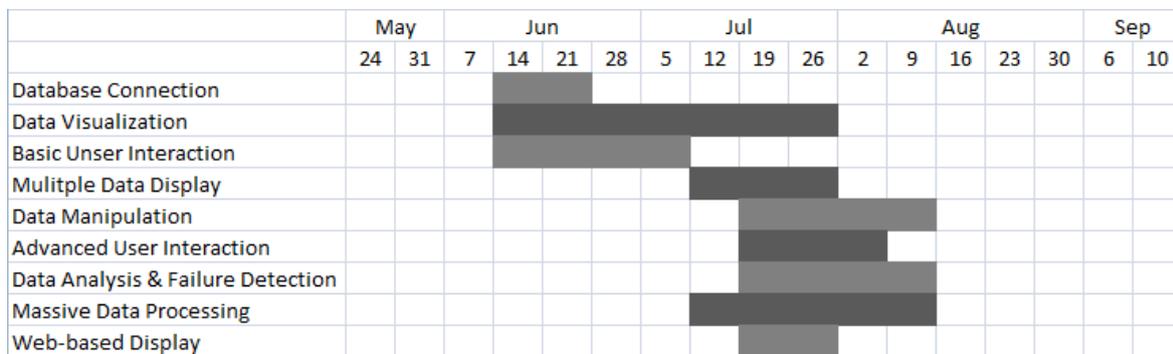		May		Jun				Jul				Aug				Sep	
	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	6	10
Database Connection																	
Data Visualization																	
Basic Unser Interaction																	
Mulitple Data Display																	
Data Manipulation																	
Advanced User Interaction																	
Data Analysis & Failure Detection																	
Massive Data Processing																	
Web-based Display																	

Figure 2. Project Deliverables Plan

C. Guide to the following Report

The remainder of the report is structured as follows: Chapter II contains a survey of existing work about the development of data analysis tools and the development of failure detection methods. Chapter III describes in detail the solution of this project. It consists of system design, choice of technology, implementation and testing. The result of this project will be discussed in Chapter IV. In Chapter V, an evaluation approach is provided and aims to gain the feedback from 2 groups of people: experts in wind energy and non-experts. Finally, the conclusions to the outcome of the projects are given in Chapter VI, along with some indications of interesting ways in which the project may be extended.

II. RELATED WORK

The existing work on the problems that this project addresses can be divided into 2 parts. One is the development of data analysis software. The other is the development of failure detection methods.

A. Development of Data Analysis Software

Data analysis tools assists experts in removing measurement errors from wind data sets and performs specialized statistical analysis. It will help experts to find out the suitable signal processing and diagnosis algorithms. Popular applications include *Microsoft Office Excel*, *Matlab*, *Windographer* and *WindRose*.

Microsoft Office Excel is a spreadsheet application distributed by *Microsoft*. It is versatile software, which features calculation, graphing tools, pivot tables and a macro programming language called *Visual Basic for Applications* (*VBA*). For professionals with a limited budget and they have to use the software they already have, *Microsoft Office Excel* is one of the best choices. Although this software is not specialized for wind data analysis, it provides a battery of functions to answer statistical and engineering needs (Bob B., 2008).

Matlab is not a professional software for wind data analysis either. It is developed by *MathWorks* and primarily used for numerical computing. The software allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other language, e.g. C/C++, Java and C#. As it has a powerful numerical computing capability and it also provide an optional toolbox for user to access symbolic computing capabilities (e.g. Neural Network, Financial tools and Model-based calibration), the software are widely used on the field of wind data analysis (Richard G., 2008).

Windographer is a specialized wind data analysis tools developed by Mistaya Engineering since 2006. It has been used worldwide to analyse meteorological data on wind farm. Users can view data time series, frequency distributions, remove erroneous data and perform calculations such as vertical shear, turbulence intensity and extreme gust. The latest version of *Windographer* provide flagging mechanism, which allow user highlight, classify, and filter data by applying user-defined flags to data segments. The software improves the productivity of data analysts, performs sophisticated calculations, and creates clear and attractive graphs (Windographer, accessed on 08 Aug 2010).

WindRose is another analysis tools which has been widely used on the field of wind energy development. It was initially designed for analysing the meteorological and oceanographic data for offshore oil and gas operations, but it was extended to be used on wind energy development since 1998 (Monkelien, K. *et al*., 2002).

In addition, according to Prof. Peter Tavner and Dr. Feng from Durham University, most of the wind turbine factories and some of the wind energy research institutes have their own data analysis tools, but due to the wind energy is a relative young technologies and the commercial secrets, they are not allowed to reveal, or disclose or transfer the software to another person or organization.

B. Development of Failure Detection Methods

According to A. Zaher *et al* (2009), the current achievements in the research of wind turbine failure detection focus on specific problem or component of the turbine in isolation. This is mainly due to the complex nature of each individual problem. Each of the research invokes both *Engineering* and *Computer Science* technologies. Depending on the nature of the data and the objective of the work, the existing researches can be classified as statistics, artificial intelligence (AI) or both of them.

Wind turbines exhibit correlation, which is dependent on the spatial distance. This kind of spatial relationship can be used to estimate and predict the performance of a single machine. Therefore, a tailored statistical method called "spatial analysis technique" was developed by Leany V. *et al* to monitor the wind turbine performance (Leany V. *et al*, 1999). Experts are

able to use this approach to classify a problem machine from a wind farm. A disadvantage with this approach is that it can only classify the overall health status of the wind turbine.

As the wind speed and active power output of a wind turbine provide information about the overall rotor condition, Caselitz. P., and Giebhardt J. (2005) introduced an online rotor fault prediction. The algorithm based on statistical technologies and calculates the average power output $\bar{P}$ (see equation (1) below) from every 300 samples data during a bin width of 0.5m/s wind speed (Red circle in Figure 3). Then the bandwidth of alarm limits can be calculated for each $\bar{P}$ (Red line in Figure 3). Any value exceeds the alarm limit, a problem with the rotor performance is detected and an alarming action is triggered by the system. This algorithm has been proven by field tests and gives reasonable threshold criteria.

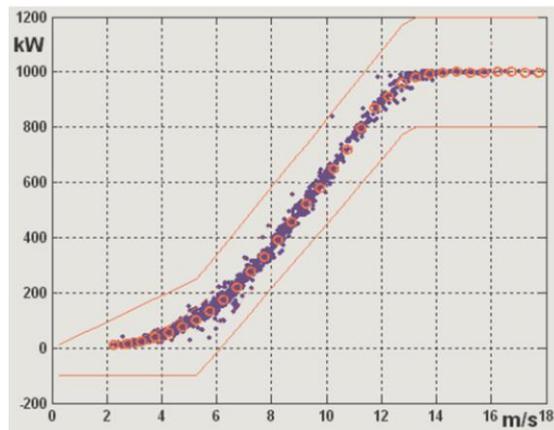$$\bar{P} = \sum_{i=1}^{300} P_i / 300 \tag{1}$$

Figure 3. Power characteristic of a wind turbine with alarm limits. Red circle is the $\bar{P}$ for each a bin width of 0.5m/s wind speed. Red line is the upper and lower alarm limit.

Another new emerging failure detection method called SIMAP, which is based on artificial intelligence techniques aimed to detect and diagnose gearbox failures. It was first introduced by Garcia M.C. et al (2006) in the conference of *Computers in Industry* in 2006. SIMAP provide a general tool oriented to the diagnosis and maintenance of wind turbine. In the real case, SIMAP is able to optimize and to dynamically adapt a maintenance calendar for a monitored wind turbine according to the real needs and operating life.

In 2007, Singh S. *et al* introduced power estimation using artificial neural network. It is used to estimate and predict power produced by each wind turbine. However, based on the estimated power output, the experts are able to identify the overall health status of each wind turbine (Singh S. *et al*, 2007).

In 2009, A. Zaher *et al* (2009) described a set of anomaly-detection techniques and their applicability to wind turbine fault identification. They also introduced a novel multi-agent system architecture that offers the opportunity to corroborate the output of the various interpretation techniques and improve the accuracy of fault detection.

C. Summary

The development of wind turbine failure detection is relatively slow; this is mainly due to the nature complex of wind energy. It would be very helpful if experts have robust data analysis software to conduct their laboratory research. However, the current development of data analysis software seems has not provided a good platform for experts to analyse wind turbine data. Both Office Excel and Matlab are not good at processing large volume of data.

Windographer and *WindRose* are currently only available for analysing the meteorological data.

III. SOLUTION

With using the problems addressed in chapter I, a Computer-Aided tool is organized with suggested solutions. This chapter begin with talking about *System Design* and follow the *Choice of Technology*. Then, the proposed *Implementation* will be introduced. Finally, the software *Testing* will be illustrated along with the expected functionalities.

A. *System Design*

The system design consists of *System Structure*, *Operational Scenarios*, *System Architecture*, *System Interface*, and *Database Design*.

Figure 4 below shows the *System Structure*. Wind Farm is the data origin. Data is acquired regularly by turbine's SCADA system and send via network to the Computer-Aid system. The Computer-Aided system stores data and provides 4 main functionalities: *Data Processing*, *Data Manipulation*, *Data Visualization* and *Failure Detection*. Experts are able to use the Computer-Aided system to develop new failure diagnosis algorithms or provide the predictive maintenance for wind turbine. In this project, the main task is to develop the Computer-Aided system. SCADA data will be provided by Engineering Department.
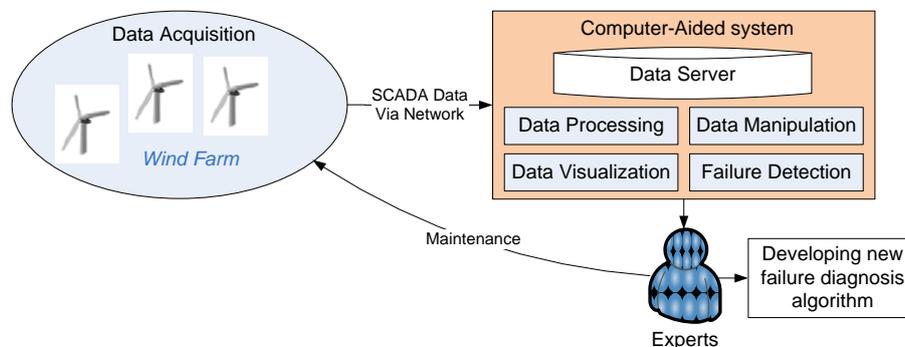

Figure 4. System Structure

Figure 5 below uses a process flow diagram to illustrate the *Operational Scenarios* of the Computer-Aid system. The system should start with *System Initialisation* and *Database Connection* checking. After that, the system will allow user to manipulate data or work with *Data Visualization*. The *User Interaction* and *Data Analysis* are provided after *Data Visualization*. Finally, the system will display the result for user.
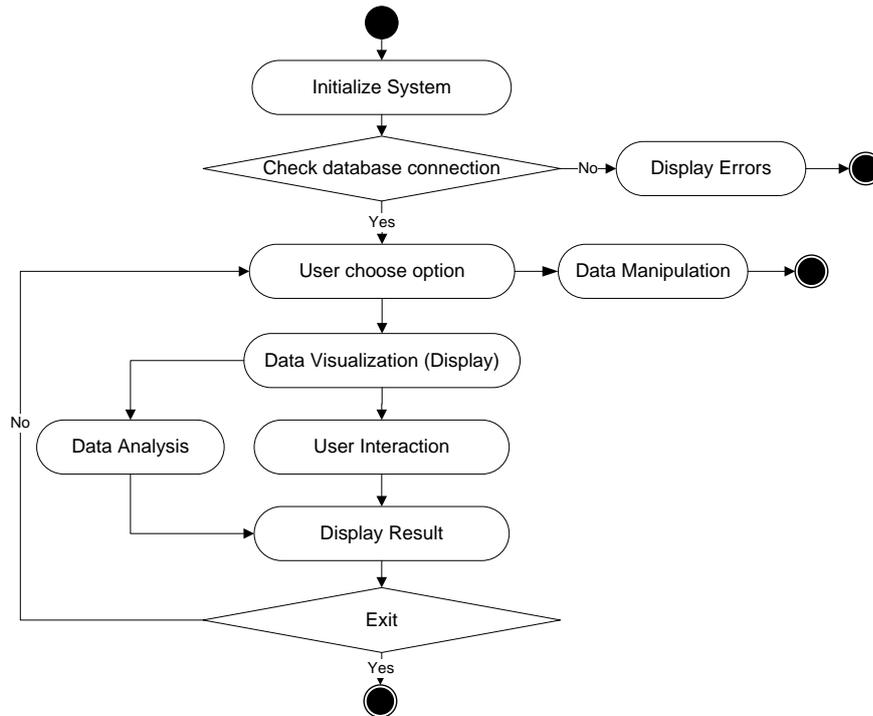
Figure 5. Operational Scenarios.

The *System Architecture* use N-tier design, which divides the whole system into 4 layers: *Presentation*, *Business Logic*, *Data Access*, and *Data Source* (see Figure 6 below). *Presentation Layer* is what user can see and interact; basically, it is the graphic user interface. *Business Logic Layer* contains business logic, validations or calculations related with the data. *Data Access Layer* contains methods that helps business layer to connect the *Data Source* and perform required actions. The main benefits of using N-tier design in this project can be summarized as following:

1. *Better code reuse:* The same logic can be invoked from many applications. *Data Access Layer* also means less repeated code to access a specific table.
2. *Easy to develop and maintain:* Centralizes business rules into a component that is easy to create, use, and re-use. This makes development and maintenance easier.
3. *Improved data integrity:* All updates have to go via *Business Logic Layer* and the *Business Logic Layer* can ensure that only valid data is allowed to be updated in the database.
4. *Good Scalability:* Provides the flexibility to add new component or separate components onto different machines.

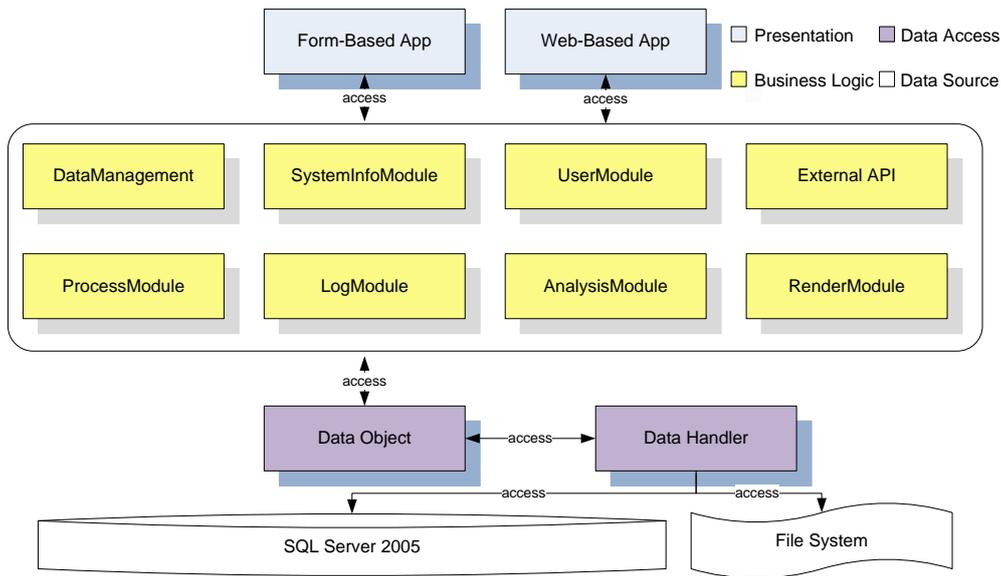The description of each module in Figure 6 is illustrated in Table 2.

Figure 6. System Architecture.

Module	Description
Form-Based App	Form-based application.
Web-Based App	Web-based application.
DataManagement	Data manipulation e.g. Data import/export.
SystemInfoModule	Store the global variables e.g. database connection string.
UserModule	Design for Web-based application and used to identify user permission
External API	Allow invoke the 3rd application to process and analysis data e.g. Matlab.
ProcessModule	Handle data processing, especially the mass data processing.
LogModule	Log of user login, system warning, and system error.
AnalysisModule	Provide data analysis and failure diagnosis solutions
RenderModule	Render data into chart.
Data Object	Initialise the data object.
Data Handler	Execute select, update, delete, and insert operations.

The *System Interface* will be designed and implemented like Figure 7 (a) below. Menus will be displayed on the top. A list of available options e.g. system control, charting options will be displayed on the left hand side. Main window shows you the data visualization and the analysis result. Figure 7 (b) is the proposed layout of the main window.
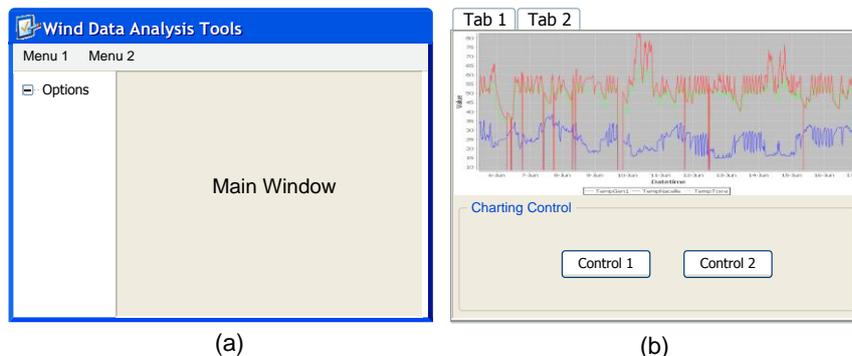

Figure 7. Proposed System Interface design. (a) layout of the system interface. (b) layout of the main window.

The *Database Design* is showed in Figure 8 below. Table *SystemLog* is used to record user login or system error. Table *ClientUser* is designed for web-based application and used

to identify user permission. The remaining tables are used to store SCADA data for different wind turbines. All of them are named like "WT_NNNN", where "WT" is short for wind turbine; "NNNN" is the turbine number. Each of "WT_NNNN" table contains 68 columns, and they stand for the 68 SCADA signals. The number of this kind of table can be varied and it depends on how many wind turbine you have in your wind farm.
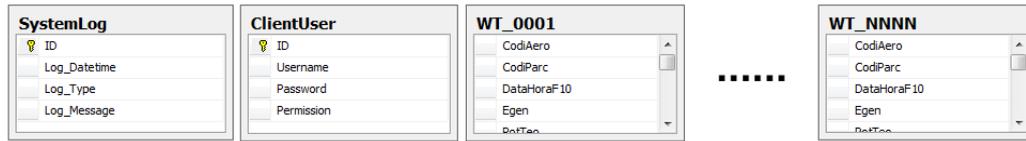

Figure 8. Database Design. Any table named like "WT_NNNN" is used to store SCADA data. "WT" is short for wind turbine, "NNNN" is the turbine number.

B. *Choice of Technology*

This project is decided to be implemented by using Java. The reasons for choosing Java are: Firstly, Java is an Object-oriented programming language; it is simple and easy to use. Secondly, Java is mature and stable; it also has rich class libraries available on the internet, this would make programming in Java become much easier. In addition, Java supports multi platform and *Remote Method Invocation* (*RMI*), this would enable massive data process in this project.

Apart from the programming language, we also need a database to store data. According to the project requirements in Chapter I, the database is required to be able to handle a large volume of off-line data (about 50G) and perform efficient data query; among the existing database options, the Microsoft SQL Server 2005 is the best choice as it have good scalability, easy manageability, and powerful administration.

C. *Implementation*

In order to make sure the system is designed and implemented properly, a sequential software development model called *Waterfall* model is involved in this project. As shown in Figure 9 below, the development process of this project is seen as flowing steadily downwards (like a waterfall) through the phases of *Requirements*, *Design*, *Implementation*, *Testing & Verification*, and *Maintenance*.
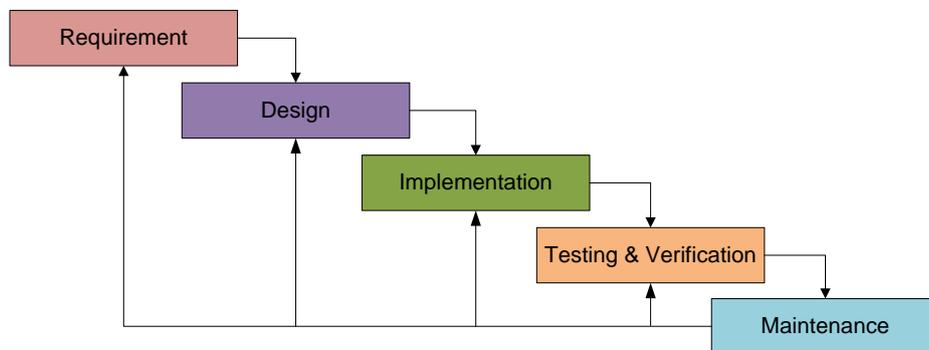

Figure 9. Waterfall model

Apart from the applied software development model, we also adopt the Version Control Strategies to keep a recorded history of the changes to source codes. The Concurrent Version System (CVS) is a version control software we used in this project to accomplish version control of the source code in *Integrated Development Environment* (*IDE*). In addition, a log book is used to record everything include the newly added functional requirements and the regular meeting with supervisor. For source code itself, we add some programming

constraints like Table 3 below:

Constraint	Sample
Naming Convention	The naming convention for identifiers in this project will follow Sun Microsystem Java communities (Sun Microsystems, Accessed on 02 Jul 2010).
File head note	Each file of the source code is forced to write a note like below: ```/* -------------------- * File Name: AverageXaxis.java * -------------------- * Author: Bindi Chen * Create Time: 27-06-2010 * Last Modify: 16-08-2010 * File Action describe: Average X axis * approach to analyse data…… */```
Comments and code	Other comments and code style should follow Code Conventions for Java Programming Language (Revised 20 Apr 1999).

The whole system is developed by using an *Integrated Development Environment* (*IDE*) called Eclipse. The reason to choose Eclipse is because it comes with many built-in features for simplifying and streamlining Java development. For example, Eclipse generates some relevant methods for component when you create it.

At the very beginning, in order to reduce the development time and improve source code quality, we import 5 external Java class libraries: *SQL Server 2005 Driver*, *JFreeChart*, *JMatlabControl*, *JCalendar*, and *iText.* The descriptions of them are listed in Table 4 below:

TABLE 4: 5 EXTERNAL JAVA CLASS LIBRARIES

Class Library	Description
SQL Server 2005 Driver	A Java Database Connectivity compliant driver that provides robust data access to Microsoft SQL Server 2005 database.
JFreeChart	A free Java chart library provided by Object Refinery Ltd. It is responsible for tasks such as drawing lines and curves, rendering charts, and handling palettes.
JMatlabControl	A free Java library that constructs the connection between Java application and Matlab.
JCalendar	A free Java date chooser bean for graphically picking a date.
iText	A free Java library that allows developer to generate PDF file.

In order to implement the system architecture like Figure 5, the Java Package is used in this project to construct every single module. For example, Package *Render* refers to the *RenderModule* in Figure 3. To build the database connection, the *SQL Server 2005 Driver* is invoked within the Package *DataHandler*.

As we mentioned in Chapter I, the Data Visualization is the most conventional way to display data with human-friendly. With using *JFreeChart* (see Table 4 above), Data Visualization becomes much easier. The *JFreeChart* also allow display multiple data series within a single chart, this enable us to implement the multiple dataset display function. The current solutions to Data Visualization have been proposed to have the following 6 different charts:

TABLE 5: 6 DIFFERENT CHART SOLUTIONS FOR DATA VISUALIZATION

Chart Solution	Description
Multiple Dataset	A line plot, which allow users to display as many data series as they want.
Combined X axis	A line plot, which has 3 sub-charts that use the same x axis.
Difference Chart	A line plot, which allow user to choose 2 different datasets and compare their difference.
Scatter Plot	A Scatter plot, which design for specific purpose. e.g. a scatter plot for Wind Speed versus Power Output.
Combined Y axis	A scatter plot, which has 3 sub-charts that use the same y axis.
Time Windows	A scatter plot, which allow user to manually define the domain of X axis and Y axis.

6 Chart Solutions have been proposed to provide the Data Visualization. However, it is not enough as user might need a flexible way to control or interact with the chart. Thus, real time user interactions are proposed in Table 6 below for all chart solutions.

TABLE 6: REAL TIME USER INTERACTIONS

User Interaction	Description	Solution
Zoom in/out	Real time user interaction to zoom in/out the chart.	Use JFreeChart (see Table 4) existing solutions. However, the domain of X axis needs to be reconstructed if it is datetime. For example: *dd-MMM-yyyy* zoom in to *hh:mm:ss*.
Chart Movement	Real time user interaction to move the chart to left/right/up/down.	Use JFreeChart existing solutions.
Chart Properties setting	Setup label name, domain name, colour, font type and so on.	Use JFreeChart existing solutions.
Save current view status	Save the user current view status of the chart	1. Use iText (see Table 4) to convert the chart panel into PDF file. 2. Use Java system draw library to convert the chart panel into PNG file. 3. Use Java system cache library to screenshot the chart panel and store in cache. Then you are allowed to paste on any software.
Tooltips	Display a tooltips when user mouse over the chart	Use Java text library to provide tooltips.
Highlight	Highlight specific date of data within Scatter Plot.	Provide date picker for user to choose the specific date. Then, render a new data layer over existing chart panel.
Display/highlight data point	Make data point visible within Line plot	A checkbox is provided for user to display data point. Then, render a new data layer over existing chart panel.

Another challenge is wind turbine failure detection. The ReliaWind research group in Engineering Department at Durham University already found 4 confident algorithms to diagnose the wind turbine. So this project will focus on studying the 4 existing confident algorithms and providing the relevant system solutions. Table 7 lists you the 4 confident algorithms with the descriptions and the system solutions.

TABLE 7: 4 CONFIDENT ALGORITHMS

Algorithms	Description	System Solution
TempGen1 - TempGenRodA > 0	The temperature difference between generator and driven end generator should larger than zero.	Use Difference Chart to detect the wind turbine failure.
TempTorr - TempGenRodA > 0	The temperature difference between non-driven end generator and driven end generator should larger than zero.	Use Difference Chart to detect the wind turbine failure.
(TempGen1 - TempNacelle) vs PotMed	Temperature difference of generator and nacelle has the linear relationship with power output	Use linear regression to find out the linear equation, and then setup a healthy zone to detect wind turbine failure.
PotMed vs VelMed	Power output has a curve relationship with wind speed.	Divide the curve into 3 parts. For each part, we use linear regression to find out the relevant linear equation, and then setup a healthy zone to detect wind turbine failure.

In addition, the massive data processing is a challenge. In this project, we proposed the following 2 solutions to solve the massive data processing problem.

1. *Dedicated Database Server:* According to Microsoft JDBC 1.1 help document, the general form of the JDBC connection URL is

```
jdbc:sqlserver://serverName;instanceName:portNumber;property=value[;property=value]
```

where **serverName** is the IP address of the server to connect to. This would allow us to construct a dedicated server for database.

2. The *External API* module in this project allows us to invoke the function from the 3rd application. This module is allowed to be extended by adding more external API. Currently, Matlab API is provided in this project and it is implemented by using *JMatlabControl* (see Table 4). The connection URL for JMatlabControl is based on *Java Remote Machine Invocation* (*RMI*), so it would definitely allow us to construct the Distribute Computing.

Finally, the *DataManagement* module will provide data manipulations. It allow user to import data from CSV file and save current operational data into a CSV file. Both of them will be implemented by using Java system IO control and database query command.

D. Testing

Prior to testing, any bugs and errors need to be resolved and kept as a record. Figure 10(a)(b) below showed the message received when an exception handling error occurs in two different execute modes. The worst errors are intermitting errors as they have no definitive cause and no system error suggestions. This would cost lot of time for a programmer to identify the cause and resolve it.
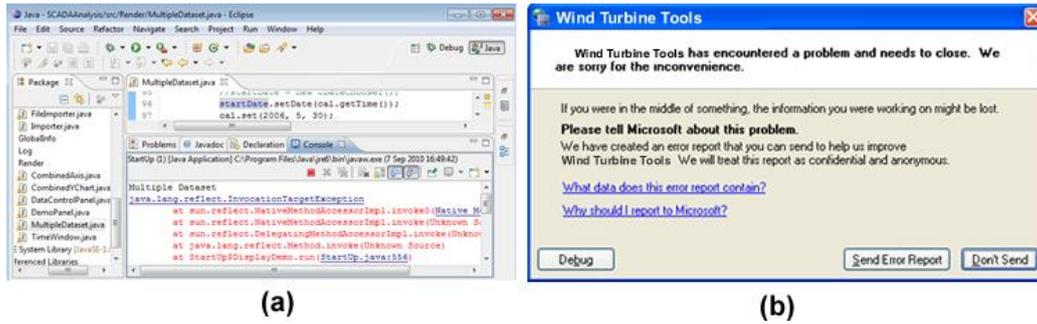
Figure 10. Error exception, (a) error message return from Ellipse, (b) is the system collapse caused by unknown reasons.

Then, before testing, we also need to allow this system to connect the network, otherwise, a windows security alert will display like Figure 11 below. To setup the permission, simply add the exception on your firewall setup or click "Unblock" button from the popup alert.

Figure 11. Windows Security Alert, click "Unblock" to allow the system to connect the network or add the exception on your firewall setting.

The testing features of functionality testing are visible. The functionality testing for this system is shown as below in Table 8:

TABLE 8: FUNCTIONALITY TESTING

Functionality	Input	Output	Expected Output (Yes/No)
Database connection	User defines database name, username, and password	Windows to confirm the successful or not.	Yes
6 different chart solutions (See Table 5 for details)	User clicks the chart solution and selects data to display.	Data Visualization	Yes
User Interactions (See Table 6 for details)	Real time user interaction to every chart.	Response to user interaction immediately	Yes
4 algorithms and detect wind turbine failure (See Table 7 for details)	User chooses relevant chart solution and selects data to analyse.	Highlight Error data and list on a table	Yes
Data import/export	User import data from CSV file or save operational data into a CSV file.	Data been imported or a CSV file has been created.	Yes

IV. RESULTS

The results of the proposed solutions in previous chapter can be summarized in Table 9 below:

TABLE 9: RESULTS

Deliverable	Results	Outcome (Yes/Partly/No)
Data Visualization	Provide 6 different Chart solutions. 1) Multiple Dataset 2) Combined X axis 3) Difference chart 4) Scatter Plot 5) Combined Y axis 6) Time Window	*Yes*
Database connection	Data come from Microsoft SQL 2005. System also allow user to setup database connection and choose dataset.	*Yes*
User interactions	System support the following user interaction: 1) Use mouse wheel to zoom in/out 2) Use mouse to drag an area to zoom in and click "zoom out" to zoom out 3) Allow press "Ctrl" and use the mouse to move the chart to left/right/up/down. 4) Allow setup chart properties e.g. colour, font style, chart orientation...... 5) Display tooltips when user mouse over the chart. 6) Allow user to highlight specific date of the data in Scatter Plot. 7) In Line Plot, system allow user to display/highlight the data point. 8) Save current view status as PNG file. 9) Save current view status as PDF file. 10) Copy current view status.	*Yes*
Multiple data display	The following Charts solution support multiple data display: 1) Multiple Dataset 2) Combined X axis 3) Difference chart	*Yes*
Data manipulation	Allow to import data from CSV file and save the operational data into a CSV file.	*Yes*
Data analysis	Provide the solution of the following 4 algorithms and detect wind turbine failure: 1) TempGen1-TempGenRodA > 0 2) TempTorre-TempGenRodA > 0 3) (TempGen1-TempNacelle) vs PotMed 4) PotMed vs VelMed	*Yes*
Data processing	2 solutions are provided to process massive data: 1) Assign an IP to *serverName* to construct a Dedicated Database Server. 2) External Matlab API is based on Java RMI, thus we should be able to build the distributed computing.	*Yes*
Web-based display	Got some progress but face lots of challenges. The web-based application need to follow the W3C standard to avoid browser compatibility problem. In addition, the chart and animation on web is much difficult than windows-based application.	*Partly*

As we can see from Figure 12 (a) to (f), the final product of the system successfully uses 6 different chart solutions to implement Data Visualization. With supporting robust user interactions list in Table 6, the data visualization in this system provides the following benefits:

1. Robust real-time user interactions e.g. zoom in/out, tooltips, and chart movement.
2. Easy to load and refine data, by simply choose dataset from dropdown list and define the data span (Start Date and End Date).
3. Every chart is featured with save current view status. Thus, user can easily grab the screenshot without the help of the screen capture software.
4. Multiple Dataset chart (Figure 12 (a)) allow users to display as many dataset as they want. It enable expert to have a good understanding of the relationship between different data.
5. Combined X axis chart (Figure 12 (b)) is really good for expert to analyse several data with different domains of Y axis.
6. Difference Chart (Figure 12 (c)) shows the value difference between 2 data. The positive value difference is showed in green; negative value difference is showed in red.
7. Scatter Plot (Figure 12 (d)) featured highlight function, which allow user to highlight the specific date of the data.
8. Combined Y axis chart (Figure 12 (e)) is designed for displaying the wind speed and power output. It shows 3 different periods of data in 3 sub-charts individually.
9. Time Window (Figure 12 (f)) allow user to customize the domain of X and Y axis. It is also featured highlight function.
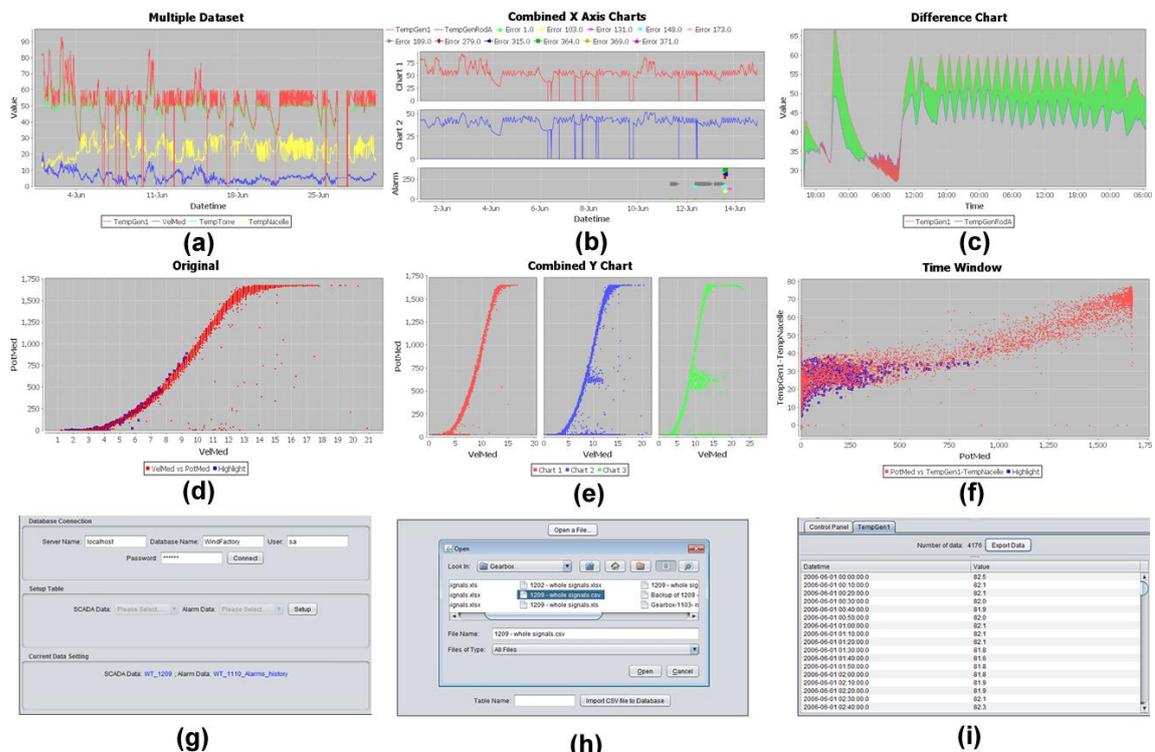

Figure 12. Graphic user interface, (a) Multiple Dataset, (b) Combined X axis, (c) Difference Chart, (d) Scatter Plot, (e) Combined Y axis, (f) Time Window, (g) Database setup, (h) Data Import, (i) Data Export.

Apart from the 6 Chart solutions, the system is also featured database setup and data import/export (Figure 12 (g)(h)(i)). Database setup allows user to assign an IP address for *serverName*, and then construct the dedicated database server. Data Import allow user to

import data from CSV file into the database. Data Export is enabled at anywhere which has data display, it allow user to export the operational data into a CSV file.

The system also successfully implemented the solutions of the 4 confident failure detection algorithms. Figure 13 (a) and (b) show you the screenshot of the algorithm 1 and the algorithm 2. Both of them use the Difference Chart, and any failure will be showed in red. For algorithm 3, Figure 13 (c) below shows that the Linear Regression approach successfully calculate the linear equation for the input data. By setup upper and lower healthy bandwidth, the system identified a healthy zone and marked it in light green. Then, any data beyond the healthy zone will be treated as turbine failure and highlighted in blue. The Algorithm 4 claimed the wind turbine power output has a curve relationship with the wind speed. The solution is to divide the curve into 3 separate lines, where each line can use the similar linear regression approach to find the failure data. Figure 13 (d) shows the result of the solution of Algorithm 4.
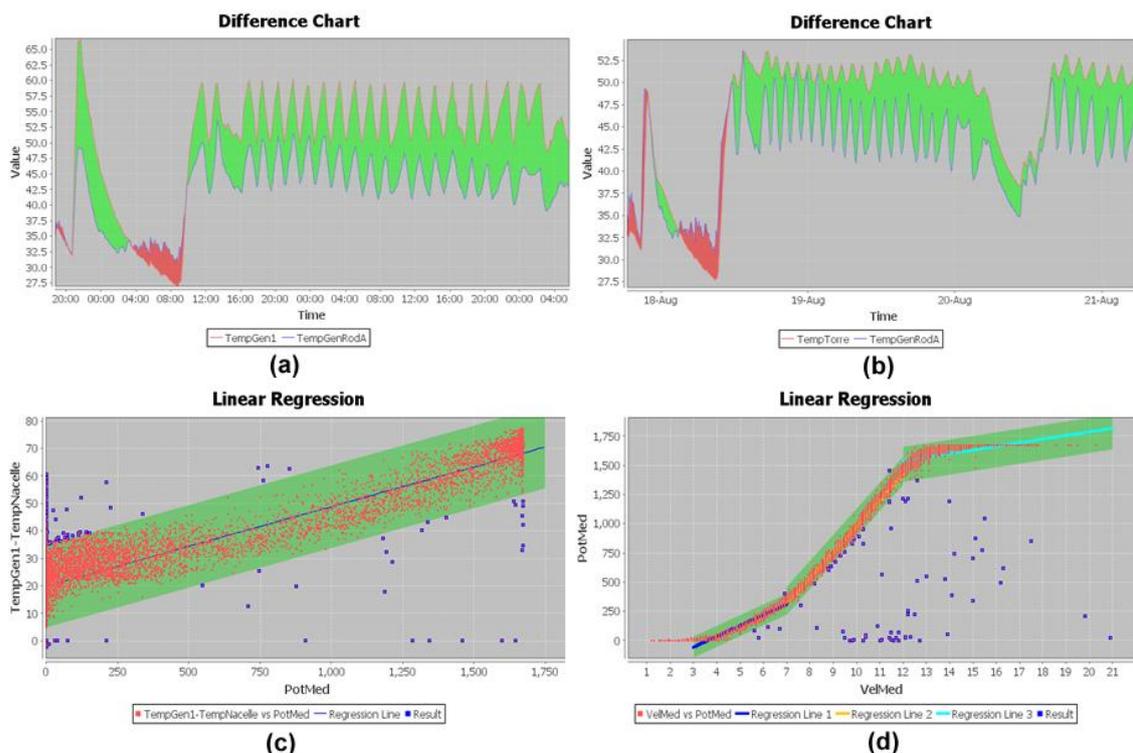

Figure 13. Wind Turbine Failure Detection, (a) Algorithm1: TempGen1-TempGenRodA > 0, red area is turbine failure. (b) Algorithm2: TempTorre-TempGenRodA > 0, red area is turbine failure. (c) Algorithm3: (TempGen1-TempNacelle) vs PotMed, blue dot are failure data (d) Algorithm4: PotMed vs VelMed, blue dot are failure data.

In order to solve the massive data processing problem, we proposed 2 solutions in previous chapter. The first solution, *Dedicated Database Server* is been test successfully and works pretty well. However, the second solution, *Matlab API with Java RMI* only works when the dataset contains less than 1000 data. A Matlab matrix initialization error happens when there is more than 1000 data been transferred to Matlab via RMI from another PC. We are trying to investigate the problem and find out a suitable solution, but due to the limited time, we have not got chance to solve the Matlab communication problem.

Finally, the development of the web-based application has some progress, but faces lot of technical difficulties. The main technical difficulties can be summarized as following: 1) Browser compatibility. 2) Large Data transmission and many client requests will slow down the application. 3) Web-based charting and animation are much difficult than form-based

application. 4) Web-based application has to consider security issues, especially for commercial use.

V. EVALUATION

This chapter explains the method used to evaluate the success of the project. According to ISO/IEC 9126-1 (2001) document, the evaluation of the software quality should identify 6 main characteristics, namely: *Functionality*, *Reliability*, *Usability*, *Efficiency*, *Maintainability*, and *Portability*. *Functionality* evaluates the achievement of the system in meeting its objectives. It is the essential purpose of the system. *Reliability* defines the capability of the system to maintain its service under defined conditions or defined period of time. *Usability* refers to the ease of use for a given function. *Efficiency* concerns with the relationship between the system performance and the resources usage. *Maintainability* identifies the ease of the system modification. *Portability* refers to how well the software can adopt to changes in its environment.

In order to evaluate above 6 characteristics of the system, two forms are designed to aim to gain the feedback. The first form aim to evaluate the success of the system in meeting its objectives, the proposed form is showed in Table 10 below:

TABLE 10: EVALUATION FORM ONE

Level	Deliverable	Outcome (Completed / Partly / No)
Basic	D1. Database Connection (Data come from database).	
Basic	D2. Data Visualization e.g. Line Plot, Scatter Plot.	
Basic	D3. User Interactions e.g. Zoom in/out, tooltips	
Intermediate	D4. Support multiple data display.	
Intermediate	D5. Import/Export data.	
Intermediate	D6. Advanced user interaction e.g. save current view status.	
Advanced	D7. Data analysis and failure detection	
Advanced	D8. Mass data processing	
Advanced	D9. Web-based display	

The second form is designed to evaluate the 6 system characteristics. The proposed form is showed in Table 11 below:

TABLE 11: EVALUATION FORM TWO

Question	Satisfaction (Excellent / Good / Fair / Poor)
Q1. Functionality (e.g. Meet requirements, Function Suitability)	
Q2. Reliability (e.g. frequency of system failures)	
Q3. Usability (e.g. Software Interface and operability)	
Q4. Efficiency (e.g. Time Behaviour and resource usage)	
Q5. Maintainability (e.g. Easy to change or extend)	
Q6. Portability (e.g. Migrate to Mac or install on another PC)	
Q7. Overall Performance	
Q8. Any problem? Please write down.	

With using above 2 forms, the process of the evaluation was taken place in 2 different groups of people: experts in wind energy and non-experts. For the first group, we invited 2

research assistants from Engineering Department: Dr. Yanhui Feng and Dr. Yingning Qiu. They installed the software on their PC and used it for several days. Dr. Yanhui Feng knew about this software but never used it before. Dr. Yingning Qiu was fully informed about all the functionalities. For the second group, we invited 8 non-experts and asked them to run the software. After several days' evaluation, the overall satisfaction can be summarized in Figure 14 below:
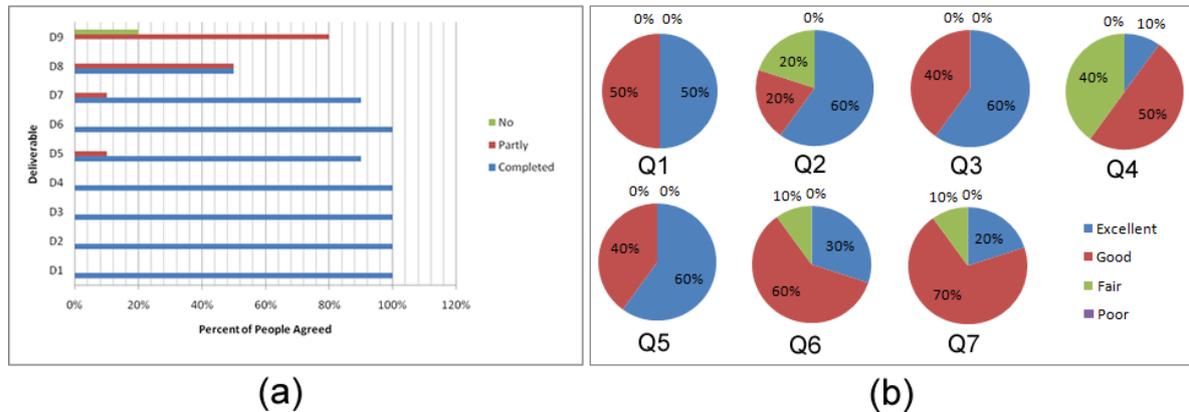

Figure 14. Overall satisfaction. (a) Summary of form 1. (b) Summary of form 2

Figure 14 (a) shows the completion rate of this project. Almost all of the people think deliverable 1 to 7 are completed. More than 50% of people think deliverable 8 and deliverable 9 are not completed. Figure 14 (b) shows the satisfaction rate of the software quality. The evaluation result of the 6 system characteristics and overall performance can be summarized as following:

Q1. Functionality – Excellent: 50% of participants rated it as good and 50% of participants rated as excellent. It means all the functionalities are suitable in this system.

Q2. Reliability – Good: Although there are 20% of participants think the reliability is fair, there still have 60% of participants think it is excellent. In overall, there were no major failures which stopped the programme.

Q3. Usability – Excellent: 60% of people think the usability is excellent. The remaining 40% of people rated it as good.

Q4. Efficiency – Fair: There are about 40% of participants rated it as fair. Some people mentioned problems with poor performance when the system is processing a large volume of data.

Q5. Maintainability – Excellent: The system architecture use N-Tier design, so it is easy to maintain.

Q6. Portability – Good: This system was implemented by using Java and Java support cross platform. The only problem is the installation cost lot of time.

Q7. Overall Performance – Good: About 90% of people rated overall performance as good or excellent.

In overall, the evaluation showed that the system has good quality, but obviously needs some improvements and modifications. The system efficiency is the worst characteristic which is the result of using large size of data. Although system allows us to construct a dedicated database server and invoke the 3rd software to manage the data process, the efficiency of current system still was not good as we expected.

VI. CONCLUSION

A. *Project Summary*

The aim of this project is to develop a Computer-Aided system for assisting expert to conduct an efficient wind turbine data analysis. The system is expected to provide data visualization, data manipulation, data processing, and wind turbine failure detection.

The development of this project was divided into several steps. The first step is discussed in Chapter I, which introduced the project background and addressed the problems. A proposed project deliverables and a project plan were also defined within this step. The second step studied the existing works on the problem that this project addressed. A summary of the study is given in Chapter II. Then, the next step is concerning with the system design and implementation. The system design discussed an operational scenario of the system, and then proposed the design of system architecture, system interface and database structure. Through analysing the available technologies, Java and Microsoft SQL database were chose to develop this system. The implementation described the applied software engineering model, version strategies and the solutions to the project deliverables. After implementation, a testing is giving to debug the whole system and ensure that the implementation meets the user needs. Finally, the system was evaluated by giving 2 evaluation forms to 2 groups of people: experts in wind energy and non-experts.

It was satisfying to see that this system completed almost all of the deliverables. 6 different Chart solutions with robust user interactions do enable experts to have an efficient laboratory research. 4 failure diagnosis solutions are able to diagnose the wind turbine and provide predictive maintenance. The final product also provided two solutions to resolve massive data processing problem.

B. *Limitations of the System and Future Work*

Although the implementation of this system has finished almost all of the deliverables, it does not mean the result is perfect. Due to the restriction of the technologies and the development of wind energy, the limitations of the system can be summarized as following:

1. The system can not work on-line. The system is initially designed for analysing off-line historical data.
2. 4 wind turbine diagnosis solutions are far from enough.
3. Existing solutions does not completely resolve the massive data processing problem.
4. Web-based application is partly completed.

As the wind energy has strong potential to contribute a much higher proportion of the electricity needs, this system is worth to investigate and further work on it (Tavner P., 2008). Firstly, it would be interesting if the system can work with on-line data. Secondly, the system requires more wind turbine diagnosis solutions and artificial intelligence can be a good solution. Thirdly, data mining can be used to find useful information from turbine data and distributed computing need to be further developed to resolve the massive data processing problem. In addition, a web-based application is need because it would enable a large group of experts working collaboratively, and then accelerate the progress of wind energy development. Finally, it would be benefit if the system has real-time intelligent management for the dynamically increasing huge volumes of data and also be able to respond quickly in failure diagnosis environment.

REFERENCES

A. Zaher, S.D.J. McArthur and D.G. Infield, (2009) "Online wind turbine fault detection through automated SCADA data analysis", *Wind Energy*, **12**(6), pp. 574-593.

Bob B., (2008) "Introduction To Microsoft Office Excel 2007", *CiCS, University of Sheffield*, July 2008

Caselitz. P., and Giebhardt J. (2005)"Rotor condition monitoring for improved operational safety of offshore wind energy converters", *Journal of Solar Energy Engineering – Transactions of the ASME 2005*; **127**: pp. 253-261.

Ciang, C.C., Lee, J.-R., and Bang, H.-J. (2008) "Structural health monitoring for a wind turbine system: a review of damage detection methods", *Meas. Sci. Technol.* **19** 122001.

Garcia M.C., Sanz-Bobi M.A., and Del Pico J. (2006) "SIMAP: Intelligent System for Predictive Maintenance; Application to the health condition monitoring of wind turbine gearbox", *Computers in Industry*; **57**: pp. 552-568

ISO/IEC 9126-1 (2001) "Software Engineering – Product Quality", *International Organization for Standardization*, Ed.1: pp.7-12

Laszlo M., Zsolt V., Gabor E., and Andras K. (2009) "AI supported maintenance and reliability system in wind energy production", *International Symposium on Method of AI, AI-METH, Poland*, pp 13-17.

Leany V., Sharpe D.J., and Infield D.G., (1999) "The applicability of spatial analysis techniques to monitoring wind farms", In *Proceedings of the European Wind Energy Conference*, **1**: pp. 216-219.

Monkelien, K. and Murrel, T.L., (2002) "Windrose, PC software for wind data analysis", *OCEANS, A Partnership of Marine Interests Proceedings*. vol.2 pp. 699-702

P. Caselitz, J. Giebhardt, and M. Mevenkamp, (1994). "On-line fault detection and prediction in Wind Energy Converters", *Proceedings of the EWEC*, Thessaloniki, pp. 623-627.

ReliaWind. Available at: http://www.reliawind.eu/ . (Accessed: 05 Jul 2010)

Richard G. (2008) "Wind Farm – Simulation Power Systems", *Computers in Industry*; **56**: pp. 124-128.

Windographer. "What's new in Windographer 2.0", Available at: http://windographer.com/ . (Accessed: 08 Aug 2010)

Singh S., Bhatti T.S., and Kothari D.P., (2007) "Wind power estimation using artificial neural network", *Journal of Energy Engineering;* **133**(1): pp. 46-52

Sun Microsystems (1999) "Code Conventions for Java Programming Language", *Sun Microsystems* press. California, US

Sun Microsystems, Naming Conventions, Available at
http://www.oracle.com/technetwork/java/codeconventions-135099.html , Access on 02 Jul 2010.

Tavner, P., (2008), "Wind power as a clean-energy contributor", *Energy Policy* , **36**(12): pp. 4397-4400