DECLARATION OF THESIS / UNDERGRADUATE PROJECT PAPER AND COPYRIGHT

Author’s full name : LING HOW SAIN

Date of birth : 17 NOVEMBER 1987

Title : DISASTER MANAGEMENT SYSTEM: LOGISTICS SUPPORT (DMSLS)

Academic Session : 2009/2010

I declare that this thesis is classified as :

☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1972)*

☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*

☑ OPEN ACCESS I agree that my thesis to be published as online open access (full text)

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:

1. The thesis is the property of Universiti Teknologi Malaysia.
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction.

Certified by:

Signature of Supervisor:

Name of Supervisor:

871117525809

(NEW IC NO./PASSPORT NO.)
"I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of Computer Science"

Signature : 

Name : Dr. MOHD ZAIDI BIN ABD ROZAN

Date : 19th APRIL 2010
DISASTER MANAGEMENT SYSTEM:
LOGISTICS SUPPORT
(DMSLS)

LING HOW SAIN

A thesis submitted in fulfillment of the requirements for the award of the degree of
Computer Science

FAKULTI SAINS KOMPUTER DAN SISTEM MAKLUMAT
UNIVERSITI TEKNOLOGI MALAYSIA

MAY 2010
I declare that this thesis entitled “Disaster Management System: Logistics Support (DMSLS)” is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature: [Signature]
Name: LING HOW SAIN
Date: 19th APRIL 2010
ACKNOWLEDGEMENT

I would like to express my special thanks to Dr Mohd Zaidi Abd Rozan, my supervisor, who has been encouraging, guiding and supporting me towards the successfulness of this project. Without the advice from Dr Mohd Zaidi Abd Rozan, it would mean hardship in researching related fields in this project.

Secondly, I would like to thanks Mr. Shariff Idros, MPPG (Majlis Perbandaran Pasir Gudang) and PAGEMA (Pasir Gudang Emergency Mutual Aid) for providing helpful and valuable information which contributed much in the development phase of the system.

Last but not least, I would extend my appreciation to my beloved family members who have been supportive at all time financially and spiritually.
ABSTRAK

ABSTRACT

Majlis Perbandaran Pasir Gudang (MPPG) has been monitoring the industries activities in Pasir Gudang for decades. With over 44 high risk industries in this compact industrial area, disaster such as explosion, fires, etc would throw people into trauma. Pasir Gudang Emergency Mutual Aid (PAGEMA) is the identical organization under MPPG, especially established for ensuring the safety of public in Pasir Gudang. They have sets of rules and regulation to react during time of crisis. However, manual data analysis and retrieval would require a bundle of time, which may results in the lost of valuable lives and properties. Resources management is the utmost issue where current data manipulation does not support efficient data retrieval. This situation worried MPPG hence the government seek for alternate solution to replace the existing manual procedure to handle disaster outcome. Efficient data retrieval and a handful of useful data enclosed great values during disaster. This paperwork is centered on my topic which is the Disaster Management System: Logistics Support (DMSLS). I am honored to design a web-based logistic system which would handle crucial and bulky data during disaster. Technologies involved would be the Asp.Net, Microsoft Server 2005. Supportive software to make this project a success would be the Google Map, Photoshop CS3, Microsoft Visual Studio 2008, Microsoft Project 2007 etc.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACKNOWLEDGMENT</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td></td>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>TABLE OF CONTENTS</td>
<td>iv</td>
</tr>
<tr>
<td></td>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>LIST OF TABLES</td>
<td>xii</td>
</tr>
<tr>
<td></td>
<td>LIST OF ABBREVIATIONS</td>
<td>xiv</td>
</tr>
<tr>
<td></td>
<td>LIST OF APPENDICES</td>
<td>xv</td>
</tr>
</tbody>
</table>

1 INTRODUCTION

1.1 Introduction 1

1.2 Background of the Problems 3

1.3 Statement of the Problems 3

1.4 Purpose of the Project 4
1.5 Objectives of the Study 4

1.6 Scope of the Project 5

1.7 Importance of the Project 5

1.8 Summary 6

2 LITERATURE REVIEW

2.1 Introduction 7

2.2 Organization Study 8

2.3 SWOT Analysis 9

2.4 Concept of Disaster Management System 10

2.5 Comparison between Existing System (SAHANA) and DMSLS

2.5.1 Interfaces of SAHANA 13

2.6 GIS 15

2.7 Summary 17

3 METHODOLOGY

3.1 Introduction 18

3.2 Methodology of Project 19

3.2.1 Steps Involved in Prototyping Process 19

3.2.2 Evolutionary Prototype Model 21
3.3 Hardware and Software Requirements

3.3.1 Hardware Specifications and Justification

3.3.2 Software Specifications and Justifications

3.3.3 Other Supportive Software

3.4 Summary

4 SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

4.2 System Analysis

4.2.1 Organization Background and Structure

4.2.2 Functions of DMSLS

4.2.3 DMSLS from MPPG Context

4.3 User Requirements

4.3.1 Functional Requirements

4.3.2 Non-functional Requirements

4.4 Module Design

4.4.1 Retrieve organization, supplier, vehicle, volunteer public users information

4.4.2 Registration organization, supplier, vehicle and public users information

4.4.3 Coordination/GIS mapping processing
4.4.4 Disaster SMS notification

4.4.5 Report Module

4.5 Interface Design

4.6 Input and Output Design

4.6.1 Input or output specification for retrieve organization, supplier, vehicle, public users information

4.6.2 Input or output specification for Registration Module

4.6.3 Input or output specification for Coordination/GIS mapping module

4.6.4 Input or output specification for disaster SMS notification module

4.6.6 Input or output specification for report module

4.7 Summary

5 System Implementation

5.1 Introduction

5.2 Database Development

5.3 User Interface Development

5.4 Programming Development
<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Gantt Chart PSM I DMSLS</td>
<td>61</td>
</tr>
<tr>
<td>B</td>
<td>Gantt Chart PSM II DMSLS</td>
<td>62</td>
</tr>
<tr>
<td>C</td>
<td>Entity-Relationship Diagram for DMSLS</td>
<td>63</td>
</tr>
<tr>
<td>D</td>
<td>Context Diagram for Disaster Management System</td>
<td>64</td>
</tr>
<tr>
<td>E</td>
<td>Diagram level 0 Disaster Management System</td>
<td>65</td>
</tr>
<tr>
<td>F</td>
<td>DFD level 1 for DMSLS</td>
<td>66</td>
</tr>
<tr>
<td>G</td>
<td>DFD level 2 Retrieve organization, supplier, vehicle, public user information</td>
<td>67</td>
</tr>
<tr>
<td>H</td>
<td>DFD level 2 Coordination/ GIS mapping processing</td>
<td>68</td>
</tr>
<tr>
<td>I</td>
<td>DFD level 2 Disaster SMS notification</td>
<td>69</td>
</tr>
<tr>
<td>J</td>
<td>DFD level 2 Register supplier, organization and public Users’ information</td>
<td>70</td>
</tr>
<tr>
<td>K</td>
<td>DFD level 2 Report Module</td>
<td>71</td>
</tr>
<tr>
<td>L</td>
<td>Hierarchy Structure of Disaster Management System</td>
<td>72</td>
</tr>
<tr>
<td>M</td>
<td>Data flow definition table for Diagram 0 (refer Appendix E)</td>
<td>73</td>
</tr>
<tr>
<td>N</td>
<td>Database Design</td>
<td>75</td>
</tr>
<tr>
<td>O</td>
<td>PAGEMA Interview Minute</td>
<td>86</td>
</tr>
<tr>
<td>P</td>
<td>User Manual</td>
<td>90</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Subsystems of Disaster Management System</td>
<td>1</td>
</tr>
<tr>
<td>2.1</td>
<td>GIS based data processing and application development for a Hot Installation disaster management</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Main Page of SAHANA</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>Organization Registration Portion of SAHANA</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Organization Report by Sector in SAHANA</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>The database view</td>
<td>16</td>
</tr>
<tr>
<td>2.6</td>
<td>The map view</td>
<td>16</td>
</tr>
<tr>
<td>2.7</td>
<td>The model view</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Prototyping Process</td>
<td>20</td>
</tr>
<tr>
<td>4.1</td>
<td>Organization chart for the MPPG as retrieved from MPPG official website</td>
<td>27</td>
</tr>
<tr>
<td>4.2</td>
<td>Functions of DMSLS</td>
<td>28</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>4.3</td>
<td>Module design of DMSLS</td>
<td>31</td>
</tr>
<tr>
<td>4.4</td>
<td>Interface of the main page of DMSLS</td>
<td>34</td>
</tr>
<tr>
<td>4.5</td>
<td>Interface of Organization Registration Module</td>
<td>35</td>
</tr>
<tr>
<td>4.6</td>
<td>Interface of the Organization Contact Person Registration Module</td>
<td>35</td>
</tr>
<tr>
<td>5.1</td>
<td>code extraction of GIS coordination insert new disaster location sub module</td>
<td>46</td>
</tr>
<tr>
<td>5.2</td>
<td>code extraction of GIS coordination (routes from disaster location to suppliers) using JavaScript from GMap API</td>
<td>49</td>
</tr>
<tr>
<td>6.1</td>
<td>Adding breakpoint</td>
<td>51</td>
</tr>
<tr>
<td>6.2</td>
<td>Adding trace point</td>
<td>52</td>
</tr>
<tr>
<td>6.3</td>
<td>Client side input testing</td>
<td>53</td>
</tr>
<tr>
<td>6.4</td>
<td>Server side input testing</td>
<td>53</td>
</tr>
<tr>
<td>6.5</td>
<td>Output testing for account administration</td>
<td>54</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>SWOT Analysis for DMSLS</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Comparison between SAHANA and DMSLS</td>
<td>12</td>
</tr>
<tr>
<td>4.1</td>
<td>List of Input retrieve organization, supplier, vehicle, public users information</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>List of Input of the Registration Module for Organization</td>
<td>37</td>
</tr>
<tr>
<td>4.3</td>
<td>List of Input for Registration Module of the Supplier</td>
<td>38</td>
</tr>
<tr>
<td>4.4</td>
<td>List of Input of the Registration Module for the Public User</td>
<td>39</td>
</tr>
<tr>
<td>4.5</td>
<td>List of Input of the Registration Module for the Vehicle</td>
<td>39</td>
</tr>
<tr>
<td>4.6</td>
<td>Input of the Coordination/GIS mapping module</td>
<td>40</td>
</tr>
<tr>
<td>4.7</td>
<td>Output of the Coordination Module</td>
<td>40</td>
</tr>
<tr>
<td>4.8</td>
<td>Input of the Disaster SMS Notification Module</td>
<td>40</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.10</td>
<td>Input for the Report Module</td>
<td>40</td>
</tr>
<tr>
<td>6.1</td>
<td>Component testing on login module</td>
<td>55</td>
</tr>
</tbody>
</table>
**LIST OF ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMSLS</td>
<td>Disaster Management System: Logistics Support</td>
</tr>
<tr>
<td>MPPG</td>
<td>Majlis Perbandaran Pasir Gudang</td>
</tr>
<tr>
<td>PAGEMA</td>
<td>Pasir Gudang Emergency Mutual Aid</td>
</tr>
<tr>
<td>SWOT</td>
<td>Strengths, Weakness, Opportunities and Threats</td>
</tr>
<tr>
<td>PEST</td>
<td>Political, Economic, Social, and Technological analysis</td>
</tr>
<tr>
<td>EPISTEL</td>
<td>Environment, Political, Informatics, Social, Technological, and Regulatory</td>
</tr>
<tr>
<td>STEER</td>
<td>Socio-cultural, Technological, Economic, Ecological, and Regulatory</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographical Information System</td>
</tr>
<tr>
<td>ERD</td>
<td>Entity Relationship Diagram</td>
</tr>
<tr>
<td>DFD</td>
<td>Data Flow Diagram</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphic User Interface</td>
</tr>
<tr>
<td>GMap</td>
<td>Google Map</td>
</tr>
</tbody>
</table>
### LIST OF APPENDICES

<table>
<thead>
<tr>
<th>APPENDIX NO</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Grant Chart PSM I DMSLS</td>
<td>61</td>
</tr>
<tr>
<td>B</td>
<td>Grant Chart PSM II DMSLS</td>
<td>62</td>
</tr>
<tr>
<td>C</td>
<td>Entity-Relationship Diagram for DMSLS</td>
<td>63</td>
</tr>
<tr>
<td>D</td>
<td>Context Diagram for Disaster Management System</td>
<td>64</td>
</tr>
<tr>
<td>E</td>
<td>Diagram 0 Disaster Management System</td>
<td>65</td>
</tr>
<tr>
<td>F</td>
<td>DFD level 1 for DMSLS</td>
<td>66</td>
</tr>
<tr>
<td>G</td>
<td>DFD level 2 Retrieve organization, supplier, vehicle, public user information</td>
<td>67</td>
</tr>
<tr>
<td>H</td>
<td>DFD level 2 Coordination/ GIS mapping processing</td>
<td>68</td>
</tr>
<tr>
<td>I</td>
<td>DFD level 2 Disaster SMS notification</td>
<td>69</td>
</tr>
<tr>
<td>J</td>
<td>DFD level 2 Register supplier, organization and public Users’ information</td>
<td>70</td>
</tr>
<tr>
<td>K</td>
<td>DFD level 2 Report Module</td>
<td>71</td>
</tr>
</tbody>
</table>
L  Hierarchy Structure of Disaster Management System  72
M  Data flow definition table for Diagram 0 (refer Appendix E)  73
N  Database Design  75
O  PAGEMA Interview Minute  86
P  User Manual  90
CHAPTER 1

INTRODUCTION

1.1 Introduction

Disaster Management System that was built is an integrated system. It is integrated from Logistics Support, Volunteer Management Solution and Missing Person Information Solution. Figure 1.1 illustrates the subsystems of Disaster Management System.

Figure 1.1: Subsystems of Disaster Management System
A group of three of us had developed a disaster management system for MPPG with the purpose of handling disaster. The system is divided into three sub systems which are the Logistics Support, Volunteer Management Solution and Missing Person Information Solution.

Disaster Management System: Logistics Support subsystem meant to store organization, supplier, public or community and vehicles information and retrieve them for use during disaster time. Followed by Disaster Management System: Volunteer Management Solution subsystem which functions to organize volunteers’ information for disaster handling purpose. Disaster Management System: Missing Person Information Solution subsystem functions to record the details of missing and found person after the disaster strikes.

This report is specific on the Logistics Support subsystem for Disaster Management System. Disaster Management System: Logistics Support (DMSLS) provides several portions which includes registration for Government and Private Sector Organization, vehicle registration, supplier registration, Geographic Information System (GIS) for mapping, vehicle management, supplier management and SMS notification ability.

Expeditious data retrieval from loads of records can save lives and properties during disaster. MPPG has been seeking for solution over the existing manual data handling method. Disaster Management System: Logistics Support (DMSLS) is the proposed system as the answer for the quest. DMSLS is a web-based information management system which targeted on government and organization for an overall monitoring during disaster time. This system is a pilot system for the MPPG to manage possible disaster happening in Pasir Gudang.

To conclude, DMSLS has the great value to serve as the pioneer system in the whole Malaysia which deals with disaster management using logistics support.
1.2 Background of the Problems

Pasir Gudang functions as the alternate entre-port besides Singapore within South East Asia. Its crowded population draws attention to the safety issues. With over 44 high risk industries located at this golden region, a simple explosion might take hundreds and thousands of lives along with it. Current approach in data handling during emergency or disaster in MPPG is done manually, such as browsing through thousands records of files. More time consumed may result in valuable lives lost and properties damage. Obviously, MPPG is lacking technology support in handling disaster. Alternate solution for the traditional way to retrieve useful data was proposed. A logistics support system which enables efficient data retrieval for use during crisis scenario supports resources management with ease. Notification ability to concerned individual or organization about the tragedy will add value to the system.

1.3 Statement of the Problems

A few questions need to be answered before the designing phase of the system to ensure fulfillment of the user requirements. The questions are as listed below:

i. Does the system support operative data retrieval to support disaster management?

ii. Can the system save time and resources in acquiring informative data?

iii. Does the system able to notify pre-determined individuals and organizations to be informed whenever there is a disaster?

iv. Does the system keep records of resource involved during the disaster management operation?

v. Will the system support MPPG to minimize lives and properties lost during disaster?
1.4 Purpose of the Project

The purpose of this project is to develop a web-based disaster management logistics support system for MPPG. The highlighted feature of this system is the effectiveness of data connection and retrieval on disaster recovery and managing issues.

1.5 Objectives of the Project

To ensure the fullness achievement of the purpose of this project, a list of objectives is listed below as a guideline for the development of the system.

1.5.1 To establish a logistic system for MPPG (Majlis Perbandaran Pasir Gudang) which handle the catastrophic outcome during time of crisis.

1.5.2 To identify the nearest resources (supplier, vehicle, organization) available during disaster.

1.5.3 To inform the person in charge or the associated organization during disaster to get necessary help support.
1.6 Scope of the Project

Before any subsequent design and development phase of the system, the scope of the project has to be identified as to aim the arrow straight to the bullet eye. A draft of the design of the system is initiated here from features of the system as classified bellow:

i. The system developed will be able to manage informative data during crisis.

ii. The system will include a function which can send SMS to involved person or organization when disaster occurred.

iii. Registration of organization or individual who can help during disaster will be one of the core functionalities inside the system.

iv. The system will visualize the available resources (supplier, vehicle, organization) relative to the disaster location in order to find nearest resources.

1.7 Importance of the Project

This system is probably the first and only web-based system which deals with disaster management in the whole Malaysia, which as well indicates the big leap for Malaysia Government in implementation of web technologies to fight calamity. MPPG as the representative of Malaysia executive representative in Pasir Gudang will be the first party to enjoy the ease brought by web technologies. This project enables overall monitoring for vehicle resources, organization and supplier using a logistic approach. Valuable information such as the nearest supplier to the location of disaster is also produced from the system.
1.8 Summary

As a conclusion, the center of the management system is to instantly retrieve informative data to handle disaster. Consequently, PAGEMA can easily manage relevant data, saving time and resource during the process. Furthermore, this system also produces valuable data such as the shortest distance of supplier from disaster location. Besides, the system also allows notification to related individuals and organizations. Overall, the system replaces the current manual system for more effective operation during disaster time.
CHAPTER II

LITERATURE REVIEW

2.1 Introduction

Literature review is not merely a summary of existing, available information. Literature review is important such that it helps to organize, describe, evaluate, clarify and summarize the literate studies of currently available system and technologies.

In this chapter, concepts of disaster management system, disaster logistics support will be explained in details. SWOT is used as the analysis method to evaluate the features of this project. Further study was centered on the Geographical Information System (GIS) technology. A comparison between existing disaster management system in Sri Lanka (SAHANA) and the proposed DMSLS will also be discussed in the later part of this chapter.

All the supportive information of the literature review is sourced from internet web pages, e-books, conference paper in pdf format, as well as some references books. All sources will be properly listed at the reference part of the report. Through the literature review, it is hopefully the development of this system will progress to its fullness, taking advantage from existing resource and technologies.
2.2 Organization Study

Studies on organization’s background and policies might help the developer to understand the functional and non-functional requirements of the organization. Policies and procedures in the organization affect the specification of the system. Hence, a study on MPPG is crucial as to have an overall understanding of the organization.

MPPG exist as the government representative in Pasir Gudang. Having the authority over Pasir Gudang Emergency Mutual Aid (PAGEMA), MPPG strive to establish a safety environment for the public of Pasir Gudang. Currently, only manual procedure is the available resources whenever there is a disaster or emergency happening in Pasir Gudang. Coordination at the disaster scenario is done using radio and cellular. Ineffective data operation (transfer, retrieval, etc) is the main issue in the operation of MPPG.
2.3 SWOT Analysis

SWOT (Strengths, Weakness, Opportunities and Threats) is introduced by Albert Humphrey in the 1960s. SWOT is a strategic planning analysis which is used to evaluate ideas and proposition. Internal and external factors which influenced the objectives accomplishment of a project are determined in SWOT. SWOT excels over PEST (Political, Economic, Social, and Technological analysis), EPISTEL (Environment, Political, Informatics, Social, Technological, Economic and Legal) and STEER (Socio-cultural, Technological, Economic, Ecological, and Regulatory) as the chosen method to evaluate this project due to its simplicity and logical order of assessment. Table 2.1 shows the output of SWOT analysis for DMSLS.

**Table 2.1: SWOT Analysis for DMSLS**

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>- fulfill the quest for an information management system from MPPG</td>
<td>- No pilot or trial done yet</td>
</tr>
<tr>
<td>- faster &amp; energy-saving data retrieval operation</td>
<td>- Advanced technologies implementation might not be too confident</td>
</tr>
<tr>
<td>- organized data which linked between one another</td>
<td></td>
</tr>
<tr>
<td>- GIS mapping produces valuable and rigorous geographical information</td>
<td></td>
</tr>
<tr>
<td>- can be accessed from anywhere around the world</td>
<td></td>
</tr>
<tr>
<td>- Management is committed and confident</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunities</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>- pioneer project of disaster management system in the whole Malaysia</td>
<td>- new technologies might not receive good feedback from users</td>
</tr>
<tr>
<td>- Profit margins might be good</td>
<td>- users might need time to adapt to the new environment</td>
</tr>
<tr>
<td>- government seeks for the technologies integration in disaster management</td>
<td>- integration to new system might take the users by surprise</td>
</tr>
</tbody>
</table>
2.4 Concept Of Disaster Management System

H.Assilzadeh and S.B.Mansor (2004) from Institute of Advanced Technology (ITMA), University Putra Malaysia suggested three kind of system for disaster management system, namely Communication, Data Distribution and Data Management Systems. In their paperwork, they stated that \textit{"data management system functions to support value-added data and information requirements of disaster management unit and the national coordinator of disaster and relief in the country."} H.Assilzadeh and S.B.Mansor (2004) agree that the disaster management process covers the extensive, resources, equipment, skills and man power. They also see effective coordination also as one of the most demanding portion.

R.K.Singh (2004) also mentioned in the India-United States Conference that the disaster management community is moving towards more effective emerging technologies such as remote sensing, Geographic Information System (GIS) and satellite communication. Hence, it is obvious that the world is moving towards the era of technologies in disaster management. Malaysia, as one of the leading country in the South East Asia Continent, should not be left out in the moving trend of technologies integration.

Figure 2.1 describe the different application layer for a disaster management system. First and foremost, the database server layer functions as the communication management layer, providing all the data needed in an disaster application or system. GIS is the portion where mapping and coordination come into place to help decision support and give consultation for geographic matters. Lastly, figure 2.1 display a send warning message module where the end users will receive notification from the state or district authorities about a disaster happening. This layer structure basically describes a standard disaster management system as a reference to DMSLS design and analysis process.
Figure 2.1: GIS based data processing and application development for a Hot Installation disaster management.
2.5 Comparison between Existing System (SAHANA) and DMSLS

SAHANA, as described at its own home page with address http://www.sahana.lk/, is a Free and Open Source Disaster Management system developed in Sri Lanka. SAHANA is a web-based disaster system similar to DMSLS but it includes a collaboration of bigger fraction. A comparison between SAHANA and DMSLS is made specifically at the logistic fragment. Table 2.2 below describes the comparison between the logistics support between the two systems.

Table 2.2: Comparison between SAHANA and DMSLS

<table>
<thead>
<tr>
<th>Function</th>
<th>SAHANA</th>
<th>DMSLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization registry and notification</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Supplier registry and management</td>
<td>χ</td>
<td>√</td>
</tr>
<tr>
<td>Vehicle registration and management</td>
<td>χ</td>
<td>√</td>
</tr>
<tr>
<td>GIS mapping and coordination</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Disaster Profile</td>
<td>χ</td>
<td>√</td>
</tr>
</tbody>
</table>

Note: Available (√) Not Available (χ)
2.5.1 Interfaces of SAHANA

Figure 2.2 to Figure 2.4 describes the interfaces of SAHANA.

**Figure 2.2: Main Page of SAHANA**
Figure 2.3: Organization Registration Portion of SAHANA

Figure 2.4: Organization Report by Sector in SAHANA
2.6 GIS

i. Introduction

A Geographic Information System (GIS) is the system which consists of hardware, software and geographical data for geographical information analysis purpose. GIS allows the user to understand the situation of a geographical area through visual aided data, which may be in form of maps, globes, reports and charts.

Arthur Getis, (2008) describes GIS as the interactive map which organizes data such that a person can select necessary data for a specific task. Arthur (2008) mentioned that GIS coordination can be from varies sources, such as global positioning units that attach a location coordinate (latitude and longitude) to a feature as small as a pump station. Integrating GIS into DMSLS can support decision making such as finding the nearest supplier to the disaster location or produce a map showing the route during disaster time.

ii. Implementation of GIS in DMSLS

Integration of GIS technology into any information system framework is possible, adding its advantage in functioning like an embedded portion in another system. In this case, integration of GIS into DMSLS contributes the domain reason for choosing it in the coordination function.

Basically, there are three different views of GIS, namely the database view, the map view and the model view. DMSLS might needs the database and map view most, where the information of a geographical area is represented in address form and plotted on map for easier queries, analysis and editing of information.
Figure 2.5: The database view

Figure 2.6: The map view

Figure 2.7: The model view
2.7 Summary

Throughout the studies of SWOT on DMSLS, Strength, Weakness, Opportunities and Threats of DMSLS are identified. Advantages of DMSLS over existing system SAHANA are to be excelled to produce a more powerful disaster management system as described. GIS mapping integration into DMSLS add more values to DMSLS in terms of disaster scenario coordination and mapping. Functionalities and features of DMSLS will outshine in MPPG, providing solution for disaster management using a logistics support.
3.1 Introduction

Evolutionary Prototyping was chosen as the methodology approach for DMSLS development. Reasons for choosing evolutionary prototyping as the targeted methodology will be explained in this chapter, addressing its advantages in helping the development of the system. Besides the justification for the methodology, hardware and software specifications and justification will also be discussed in the later part of this chapter. Appropriate choice in the methodology, hardware and software contribute to the successful application development.
3.2 Methodology of Project

The methodology to build up DMSLS is prototyping model approach which is a popular methodology for most programming environment which require high fulfillment of user requirements. Evolutionary prototyping methodology uses the creation of the model incrementally towards the system required by user. In the middle of the prototyping process, user reactions, suggestion, innovation is evaluated in order to come out with a better system model. The development process requires reviewing user requirement from time to time.

3.2.1 Steps Involved in Prototyping Process

Step 1: Initial analysis

In this step, basic requirements from the user is studied, comprises input and output desired by the user. Detail of the system is ignored during this step.

Step 2: Define prototype objectives

Objectives of prototype are defined so as to evaluate the performance of prototype by the end of the day.

Step 3: Specify prototype

In this step, system is defined in such a way that it can meet the user requirements. Major process and data input, output are defined together with physical system programs and database.

Step 4: Construct prototype

Implement the design from step 3 into the prototype from the well-understood part first.

Step 5: Evaluate prototype and recommend changes
Prototype constructed is presented for review. Comments and suggestions from client are analyzed and possible improvements are identified. Step 3 and 4 are repeated until the objectives of the prototyping model are achieved.

Step 6: Complete prototyping

Figure 3.1: Prototyping Process
3.2.2 Evolutionary Prototype Model

The prototype model has been discussed into quite detail as above-mentioned. The evolutionary prototype model used as the methodology of the system development aimed to build prototype in structured manner and continuously refine it. The evolutionary system prototyping came out with incomplete iterative system but modified in the process of testing, validating and re-designing. Figure 3.2 illustrates the process of evolutionary prototyping development.

In the evolutionary prototype model, an initial version of system is developed until the final version which is more of user satisfactory. Developer started off with the more understood part after the user requirements analysis (as discussed in section 3.2.1). Partial system came out from the initial phase of prototyping is sent to the user for testing purpose. Feedback from the user is analyzed and contributed towards modifying the prototype until a final prototype came out. This process of modifying is continuously and concurrently with the prototyping process.

3.3 Hardware and Software Requirements

DMSLS require specific hardware and software to be developed. DMSLS is a web-based information system which handles data during disaster time. Hardware and software requirements to develop the web application would be as specified bellow:
3.3.1 Hardware Specifications and Justification

Proper hardware specifications and justification is crucial so as to support the operation of the system. DMSLS’s performance depends greatly on the hardware of the computer. To allow the system to run smoothly without system collapse, the minimum hardware specification of the users’ computers is as listed below:

i) Processor: Intel Pentium IV and above

ii) Processor: Minimum 233 MHz and above

iii) Hard Disk: 2.1 GB and above

iv) Memory Size: 1.0 GB and above

v) Cache Size: 512 KB and above

3.3.2 Software Specifications and Justifications

Software specifications and justifications is as important (if not more important) as the hardware’s such that the developer can ensure the success in the development process. Proper specifications in software also enhance the system’s functional operations during the development phase. The software specifications and justifications is as below:

i) Operating System: Windows XP

ii) Database Server: Microsoft SQL Server 2005

iii) Server Side Scripting language: ASP.Net

ASP.Net is chosen as the scripting language for the system to build interactive and attractive user-interface, making the system more user-friendly. In addition, ASP.Net communicates with Microsoft SQL Server 2005 effectively, making it the most appropriate web-programming language in this system development. Microsoft SQL Server 2005 is chosen as the database server due to its high speed in data transfer and supportive features such as the report viewer service etc.

3.3.3 Other Supportive Software

i. Photoshop CS3

Photoshop CS3 is software for graphical design and editing. It helps in the graphical image designing for the DMSLS throughout the whole development phase. Powerful editing tools make it even popular among the professional level of designing.

ii. Microsoft Project 2007

Microsoft Project 2007 produces the Grant Chart in the designing and planning phase of the system. It helps as the monitor to the incremental of the system progress.

iii. Rational Rose

Rational Rose was chosen to develop the Unified Modeling Language (UML) for the system in analysis and modeling phase. It was thus used to develop the process dependency diagram, DFD and ERD diagram for the data description purpose. Through these diagrams, the structure of the database and dataflow in the system can be described clearly to the users and those who concerned.
3.4 Summary

Evolutionary prototyping is chosen as the methodology to develop DMSLS. Evolutionary prototyping delivered the system from an initial prototype until the final prototype. During this delivery period, concurrent activities such as specification, testing, evaluation took place to modify the prototype into a better version. Hence, evolutionary prototyping can produce higher user satisfaction. The satisfactory of the user by achieving most users’ requirements is the measure of success of a system. Evolutionary prototyping is definitely the approach to that gate.
CHAPTER IV

SYSTEM ANALYSIS AND DESIGN

4.1 Introduction

System analysis for development of DMSLS is an important phase since it identifies the requirements from users and identifies functions of the system to enhance the design phase of the system. This chapter will determine the user requirements for DMSLS mainly from the organization view, in this case the MPPG. Database design of DMSLS will be displayed using a combination of diagram, consisting of the class diagram, Entity Relationship Diagram (ERD), Data Flow Diagram (DFD) and other selected methods of graphic representation. Besides, the output and input design will also be discussed in this particular chapter. Last but not least, the Graphic User Interface (GUI) will be displayed as a preliminary model for the layout design of DMSLS. Throughout the system analysis and design phase, it is hopeful that the development of the DMSLS will be able to achieve all functional and non-functional requirements, with appropriate design interface.
4.2 System Analysis

4.2.1 Organization Background and Structure

A developer has to analyze the organization’s requirements before initiating development phase of the system. The background and structure of the MPPG help the developer to identify the requirements and expectation of MPPG in DMSLS. On the other hand, certain policies and procedures of organization have to be well understood in order to specify the system in terms of its functionality and performance.

MPPG acts as the existing government authority in Pasir Gudang to oversee all activities in Pasir Gudang including the safety of public. On the other hand, Pasir Gudang Emergency Mutual Aid (PAGEMA) also plays an important role during emergency or time of crisis. For the time being, PAGEMA only have sets of rules and procedure to be followed during disaster. PAGEMA lacks of technologies support in resources management. Usage of radio and cellular phone are the only technologies they have in scenario coordination and data or information transfer.

Figure 4.1 describe the current organization structure of MPPG. As described in the chart, MPPG does have an Information Technology Department under its supervision, functions to support MPPG using technology implementation in data manipulation. Without a doubt, MPPG is more than willing for any technology support or revolution in the handling the government job scope.
Figure 4.1: Organization chart for the MPPG as retrieved from MPPG official website.
4.2.2 Functions of DMSLS

Appendix L is the hierarchical diagram for Disaster Management System as a whole. I am concentrate in developing the Logistics Support subsystem inside the Disaster Management System.

Several functions of the Logistics Support subsystem identified are as displayed in figure 4.2 below. All the functions strive to achieve MPPG’s requirements to enhance data retrieval process during time of crisis.

![Diagram of Disaster Management System: Logistics Support]

**Figure 4.2:** Functions of DMSLS
4.2.3 DMSLS from MPPG Context

As been discussed many times earlier, MPPG has been seeking alternative for data-handling during disaster time. DMSLS will serve to effectively retrieve data from the database with relatively higher speed, saving time and energy. Besides, DMSLS also include GIS mapping and coordination to efficiently search the indicated suppliers according to the distance from the disaster scenario. It is undoubtedly a revolution in the MPPG context to implement web-technologies into saving valuable lives and enhance disaster recovery process.

4.3 User Requirements

Determining user’s requirements is an important step in analysis phase of structural approach. The MPPG’s requirements will mostly determine how DMSLS is going to fulfill. There are, however, the non-functional requirements which have to be reviewed during the analysis phase so as to avoid big scaling changes in the development phase of DMSLS. These requirements basically identified from research, literature review and also the interview session with PAGEMA. A minute is attached in Appendix O as a reference of the interview conducted.

4.3.1 Functional Requirements

i. Get useful information from a bulk of files/ data in shortest time available

Time is everything during disaster or crisis hour. Fast information retrieval is most crucial as it concerned with lives of many. DMSLS must be able to retrieve data fast, effective and reliable.

ii. Notify concerned organization, suppliers or other parties during disaster time

Organization, suppliers or concerned parties need to be notified during disaster time so they can react instantly and provides all possible means of help.

iii. Get the coordination of available resources near to the disaster location

Coordination of the nearest available resources is very valuable to MPPG. With DMSLS, MPPG does not have to manually figure out the nearest resources.
4.3.2 Non-functional Requirements

i. Product Requirement

Interface of DMSLS has to be properly designed to represent MPPG in the
government context. Simple yet attractive GUI is the requirement developer strives to
achieve.

ii. External Requirement

DMSLS have to keep the personal information of registered users to MPPG
unless with agreement from the party itself.

iii. Performance Requirement

DMSLS has to be available almost all the time because disaster might happen at
any time, anywhere.

iv. Security Requirement

Since data of certain government organization and suppliers might be
confidential, DMSLS has to be secure to avoid any form of manipulate by the third
party.
4.4 Module Design

Figure below will describe the modules available in DMSLS to co-operate in data manipulation to gain useful data. There are exactly 6 main modules inside the DMSLS as described in figure 4.3 below.

![Module design of DMSLS](image)

**Figure 4.3:** Module design of DMSLS

4.4.1 **Retrieve organization, supplier, vehicle, public users information**

In this profile retrieval module, MPPG is able to view organization, supplier, vehicle and public users list. This module is also integrated with subsystem Volunteer Management solution in view volunteer function. MPPG need to know everything to details of the available resources or support it has during a disaster. Operation of data retrieval can be found in Appendix G.
4.4.2 Register supplier, organization, vehicle and public users’ information

Inside the registration module, information of the public, supplier, organization and vehicle is retrieved from the users and kept inside the database system. Registration module will produce details of information inputs and organize the data into specific tables. To conclude, registration module functions as the input module where valid data which comes from users’ input will be kept as tables inside the database to produce useful information output. The module is described in diagram level 2 of register module (Appendix J).

4.4.3 Coordination/ GIS mapping processing

Coordination module functions based on the GIS technologies. GIS mapping inside the system can produce the nearest available supplier registered inside the system. As an example, if MPPG needs to know the nearest can food supplier to location X, coordination module will calculate the distance between all registered suppliers to location X and finally gives the location of the suppliers according to nearest distance. Volunteer in subsystem volunteer management solution will also retrieve map from this module to get their route to the disaster or supplier location. Process specification of this module can be found at Appendix M while the diagram level 2 of coordination module is in Appendix H.

4.4.4 Disaster SMS notification

Send SMS module of DMSLS is triggered by MPPG whenever there is a disaster happened. SMS which contains information concerned with the catastrophe will be send via DMSLS to mobile hand phones of registered organization, suppliers and even the public. MPPG can also specify the SMS to be sent to certain users group only. Send SMS module is highly recommended to be include inside the system due to drastic actions required in time of crisis. For example, the Red Crescent can prepare well the medical equipments right after they received SMS reporting about a severe incident in
Pasir Gudang. Their fast response can save lives of many during the hard time. Appendix I describes this module in detail.

4.4.5 Report Module

Last but not least, the logistic report is included to make the disaster information handling process as a whole. This module will produce required report to MPPG according to information of supplier, organization, and vehicle. Hence, MPPG can always have an overall monitoring of its available resources. Further explanation can be found in Appendix K.

4.5 Interface Design

Interface design of DMSLS is initiated as the model of GUI for the system. Interface design of DMSLS has to be suitable since it will represent the image of MPPG as the authority in Pasir Gudang. First phase of interface design for DMSLS will be integrated from time to time since amendments have to be made in order to come out with interactive, user-friendly, effective and harmony interface for DMSLS. Below are some figures showing samples of the GUI for DMSLS.
Figure 4.4: Interface of the main page of DMSLS
**Figure 4.5:** Interface for Organization Registration Module

**Figure 4.6:** Interface of the organization contact person registration module
4.6 Input and Output Design

Input and output design of DMSLS directly related to the modules available inside the system. Some modules require users’ input while others produce output for the users. Following are tables describing the input and output design of DMSLS in an easier understanding way. The tables will be listed according to the modules available as discussed earlier.

4.6.1 Input or output specification for retrieve organization, supplier, vehicle, public users information

Table 4.1: List of Input retrieve organization, supplier, vehicle, public users information

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization ID</td>
<td>Organization ID functions as the primary key to retrieve identical organization information</td>
</tr>
<tr>
<td>Supplier ID</td>
<td>Supplier ID functions as the primary key to retrieve identical supplier information</td>
</tr>
<tr>
<td>Public ID</td>
<td>Public ID functions as the primary key to retrieve identical public users’ information</td>
</tr>
<tr>
<td>Vehicle ID</td>
<td>Vehicle ID functions as the primary key to retrieve identical vehicle information</td>
</tr>
</tbody>
</table>
### 4.6.2 Input or output specification for Registration module

**Table 4.2**: List of Input of the Registration module for Organization

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization Login Username</td>
<td>Username of the organization to login the system.</td>
</tr>
<tr>
<td>Organization Login Password</td>
<td>Password of the organization to login the system.</td>
</tr>
<tr>
<td>Organization Name</td>
<td>Name of the registered organization.</td>
</tr>
<tr>
<td>Organization registered number</td>
<td>The registered number of the organization as with the government.</td>
</tr>
<tr>
<td>Organization Address</td>
<td>Address of the organization.</td>
</tr>
<tr>
<td>Organization Description</td>
<td>A brief description of the organization and how it can be of help to MPPG during disaster time.</td>
</tr>
<tr>
<td>Contact Person Name</td>
<td>Name of the person as representative of the organization.</td>
</tr>
<tr>
<td>Contact Person HP Number</td>
<td>Hand phone number of the contact person which the SMS notification will be delivered by DMSLS.</td>
</tr>
<tr>
<td>Contact Person Email</td>
<td>Email account of the contact person which the email notification will be delivered by DMSLS.</td>
</tr>
<tr>
<td>Contact Person Address</td>
<td>Address of the contact person as a basic record of that particular person</td>
</tr>
</tbody>
</table>
**Table 4.3:** List of Input for Registration module of the Supplier

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier Login Username</td>
<td>Username of the supplier to login system.</td>
</tr>
<tr>
<td>Supplier Login Password</td>
<td>Login Password for the supplier.</td>
</tr>
<tr>
<td>Supplier Name</td>
<td>Name of the registered supplier.</td>
</tr>
<tr>
<td>Supplier registered number</td>
<td>The registered number of the supplier as with the government.</td>
</tr>
<tr>
<td>Supplier Address</td>
<td>Address of the supplier.</td>
</tr>
<tr>
<td>Supplier Category</td>
<td>Category which the supplier fall into.</td>
</tr>
<tr>
<td>Supplier Description</td>
<td>A brief description of the supplier and logistic it can provide to MPPG during disaster time.</td>
</tr>
<tr>
<td>Contact Person Name</td>
<td>Name of the person as representative of the Supplier.</td>
</tr>
<tr>
<td>Contact Person HP Number</td>
<td>Hand phone number of the contact person which the SMS notification will be delivered by DMSLS.</td>
</tr>
<tr>
<td>Contact Person Email</td>
<td>Email account of the contact person which the email notification will be delivered by DMSLS.</td>
</tr>
<tr>
<td>Contact Person Address</td>
<td>Address of the contact person as a basic record of that particular person</td>
</tr>
</tbody>
</table>
**Table 4.4**: List of Input of the Registration module for the Public User

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Login Username</td>
<td>Login Username of the public to the system.</td>
</tr>
<tr>
<td>Public Login Password</td>
<td>Password used by the public user to login the system.</td>
</tr>
<tr>
<td>Public Name</td>
<td>Name of the user.</td>
</tr>
<tr>
<td>Public Address</td>
<td>Address of the public.</td>
</tr>
<tr>
<td>Public HP Number</td>
<td>Hand phone number of the public user which the SMS notification will be delivered by DMSLS.</td>
</tr>
<tr>
<td>Public Email</td>
<td>Email account of the public user which the email notification will be delivered by DMSLS.</td>
</tr>
</tbody>
</table>

**Table 4.5**: List of Input of the Registration module for the vehicle

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Registered No</td>
<td>Official registered Number of the vehicle.</td>
</tr>
<tr>
<td>Vehicle Category</td>
<td>Category of the vehicle.</td>
</tr>
<tr>
<td>Vehicle Capacity</td>
<td>Capacity of the vehicle whenever applies.</td>
</tr>
<tr>
<td>Vehicle Weight</td>
<td>Approximate weight of the vehicle.</td>
</tr>
<tr>
<td>Vehicle Description</td>
<td>A brief description of the vehicle.</td>
</tr>
<tr>
<td>Vehicle Owner Name</td>
<td>Name of the Owner to the vehicle.</td>
</tr>
<tr>
<td>Vehicle Contact Person Name</td>
<td>Name of the contact person (usually will be the same as the owner to the vehicle).</td>
</tr>
<tr>
<td>Vehicle Contact Person HP Number</td>
<td>Hand Phone number of the contact person.</td>
</tr>
<tr>
<td>Vehicle Address</td>
<td>The address where usually the vehicle will most probably be.</td>
</tr>
</tbody>
</table>
### 4.6.3 Input or output specification for Coordination/ GIS mapping module

**Table 4.6:** Input for the Coordination/ GIS mapping module

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location of disaster</td>
<td>Location where the disaster happened, in longitude and latitude format.</td>
</tr>
<tr>
<td>Supplier Category</td>
<td>Category of Supplier to be searched.</td>
</tr>
</tbody>
</table>

**Table 4.7:** Output for the Coordination module

<table>
<thead>
<tr>
<th>Output</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Supplier</td>
<td>List of Supplier in order of nearest distance to the disaster scenario.</td>
</tr>
<tr>
<td>Address of the Supplier</td>
<td>Addresses of the respective suppliers.</td>
</tr>
</tbody>
</table>

### 4.6.4 Input or output specification for Disaster SMS Notification Module

**Table 4.8:** Input for the disaster SMS notification module

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Content</td>
<td>The content of the SMS/ Email to be sent to indicated receivers.</td>
</tr>
<tr>
<td>Receivers list</td>
<td>List of the indicated receivers to certain SMS/ Email.</td>
</tr>
</tbody>
</table>

### 4.6.5 Input or output specification for report module

**Table 4.10** Input for the report module

<table>
<thead>
<tr>
<th>Input</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle list</td>
<td>Registered vehicles in the system</td>
</tr>
<tr>
<td>Organization list</td>
<td>Registered organizations in the system</td>
</tr>
<tr>
<td>Supplier list</td>
<td>Registered suppliers in the system</td>
</tr>
</tbody>
</table>
4.7 Summary

To conclude this chapter, the system requirements of DMSLS have been well identified. Five modules in DMSLS, namely Login/Logout, Registration, Coordination, SMS/Email and Report Module serve to achieve all the requirements from MPPG. Samples of interface design for DMSLS as shown targeted to interact with the users conveniently. The input and output design justification at the later of the chapter helps the designer to have a clearer view of the function of each module.
CHAPTER V

System Implementation

5.1 Introduction

This chapter intends to explain deeper the development processes of the DMSLS based on the previous discussion in chapter 4. The system development processes includes several parts namely database development, user interface development, programming development and system module development. Installation of visual studio 2008 and other required software is a prerequisite for the system development. Database is developed in Microsoft SQL Server 2005 according to the input table as discussed in chapter 4. Tables in database are designed according to the identified input forms. Consequently, user interface development and programming development are done accordingly based on the system modules.
5.2 Database Development

Database in the system is developed using Microsoft SQL Server 2005 which works perfectly in Visual Studio 2008 with Asp.Net project files. During the development phase of database tables, the input forms identified earlier in chapter 4 is studied further. Each table in the database was assigned a primary key which work as a foreign key in other tables it is related with. The primary key works so as to uniquely identify a row of data inside the table for retrieval. Besides, each data was assigned specific type and size based on the data it stores. List of table can be found in Appendix N.

5.3 User Interface Development

Disaster Management System we developed is an integrated system. Hence, the user interface is standardized so as to give a harmonized outlook. The design of user interface is being done to make the system user friendly and reserve its usage simplicity at the same time. Microsoft Visual Studio 2008 and Adobe Photoshop CS3 were the main software used to develop the user interface of DMS.
5.4 Programming Development

There are 5 modules in the DMSLS which are registration module, information retrieval module, SMS notification module, GIS coordination module, and Report module. GIS coordination module will be discussed further in this chapter due to its complexity.

5.4.1 GIS coordination module

The GIS coordination module in DMSLS is developed using API from Google Map (GMap). The programming part was done in Microsoft Visual Studio 2008 using JavaScript which work together with vb.net for the required output. JavaScript is a client side programming language while vb.net is running at server side. Since GMap basically support JavaScript from a wider range, the system developer needed to configure the communication between these two different languages.

Figure 5.1 shows the code extraction of GIS coordination insert new disaster location sub module where the server store the disaster location determined by the user. The user will first click on the map (a marker will be placed synchronously with the click event) indicating the disaster location. The latitude and longitude of the marker is gained from the map once the user click the submit button. The code is programmed such that it will check whether the Reference ID data input and allow the insertion of new disaster location only if the Reference ID is not found in the Disaster_Location table.

Example of JavaScript from GMap API is shown in figure 5.2 where the routes between disaster location and supplier are illustrated in the system. The javascript will get two arrays from visual basic .net code behind, namely supp_points[ ] and supplier_name[ ] which stored the supplier points and supplier names accordingly. The JavaScript will then get the disaster location from the textbox by the id Disaster_Lat and Disaster_Lng. Disaster_Lat store the latitude of the disaster location while Disaster_Lng store the longitude of the disaster location. The client side coding will then generate a loop of waypoints which store the routes between array of supplier points and the disaster location. Inside this loop, info window will be generated for each marker of
each supplier marked on the map. After this looping is terminated, the JavaScript will draw the waypoints (array of routes) on the map, indicating the routes from suppliers to the disaster location. The distance is thus visualized. The markers are added accordingly for each supplier and also for the disaster location.

```vbnet
Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnAdd.Click
Dim Record As Boolean
Dim act As Boolean
act = MsgBox("Are you sure you want to insert new disaster location?", vbYesNo, "Confirm Action")
If (act) Then
    Using myConnection As New SqlConnection(ConfigurationManager.ConnectionStrings("ConnectionString").ConnectionString)
        Const SQL4 As String = "SELECT * FROM [Disaster_Location] WHERE ([Ref_ID] = @Ref_ID)"
        Dim myCommand4 As New SqlCommand(SQL4, myConnection)
        myCommand4.Parameters.AddWithValue("@Ref_ID", RefID.Text.ToString)
        myConnection.Open()
        Dim myReader4 As SqlDataReader = myCommand4.ExecuteReader
        If myReader4.Read() Then
            Record = True
        End If
    End Using
    myConnection.Close()
    If (Record = False) Then
        MsgBox("No previous record!")
    Else
        MsgBox("There is record on disaster on this reference ID!")
        Response.Redirect("../MPPG/GMapRouteDistance.aspx")
    End If
End If
Const SQL3 As String = "Insert into Disaster_Locat(on Lat,Lon,Date,Disaster_Desc,Ref_ID) Values (@Lat,@Lon,@Date,@description,@Ref_ID)"
Dim myCommand3 As New SqlCommand(SQL3, myConnection)
myCommand3.Parameters.AddWithValue("@Date", dateDisaster.Text.ToString)
myCommand3.Parameters.AddWithValue("@Ref_ID", RefID.Text.ToString)
Dim result3 As Integer
myConnection.Open()
result3 = myCommand3.ExecuteNonQuery()
If (result3) Then
    MsgBox("Your location successfully inserted!")
End If
myConnection.Close()
Const SQL As String = "SELECT * FROM [Disaster_Location] where [Ref_ID]=@Ref_ID"
Dim myCommand As New SqlCommand(SQL, myConnection)
Dim x, y As Double
myCommand.Open()
myCommand.Parameters.AddWithValue("@Ref_ID", RefID.Text.ToString)
Dim myReader As SqlDataReader = myCommand.ExecuteReader
If myReader.Read() Then
    x = myReader("Lat")
    y = myReader("Lon")
```
Figure 5.1: code extraction of GIS coordination insert new disaster location sub module

```vbnet
GMap1.Width = "600"
GMap1.Height = "450"
GMap1.setCenter(New Subgurim.Controles.GLatLng(1.461234, 103.878123), 12,
Subgurim.Controles.GMapType.GTypes.Normal)
Dim GIcon As New Subgurim.Controles.GIcon
GIcon.image = "http://gmaps-samples.googlecode.com/svn/trunk/markers/green/blank.png"
Dim MarkerOption As New Subgurim.Controles.GMarkerOptions()
MarkerOption.icon = GIcon
MarkerOption.draggable = True
Dim oMarker As New Subgurim.Controles.GMarker(New Subgurim.Controles.GLatLng(x, y), MarkerOption)
GMap1.addGMarker(oMarker)
GMap1.addListener(New Subgurim.Controles.GListener(oMarker.ID,
Subgurim.Controles.GListener.Event.dragend, "function(){
var ev = new
serverEvent('dragend', 0);


"& oMarker.ID & ".getPoint().lat();


"& oMarker.ID & ".getPoint().lng();


()})
'Keyboard navigation
GMap1.enableGKeyboardHandler = True
'Mouse wheel zoom
GMap1.enableHookMouseWheelToZoom = True
'Add map type
GMap1.addMapType(Subgurim.Controles.GMapType.GTypes.Physical)
'Map Type control
GMap1.addControl(New
Subgurim.Controles.GControl(Subgurim.Controles.GControl.preBuilt.MapTypeControl)
)
GMap1.enableRotation = True
Else

End If
myConnection.Close()
End Using
End If
End Sub
```
<%@ Page Language="VB" Debug="true" MasterPageFile="~/masterpage/Logistic.master" AutoEventWireup="false" CodeFile="GMapRouteDistance.aspx.vb" Inherits="MPPG_GMapRouteDistance" title="Untitled Page" %>

<% @ Register assembly="GMaps" namespace="Subgurim.Controles" tagprefix="cc1" %>
<% @ Register assembly="AjaxControlToolkit" namespace="AjaxControlToolkit" tagprefix="cc1" %>

<asp:Content ID="Content1" ContentPlaceHolderID="content" Runat="Server">

<head>
<script src="http://maps.google.com/maps?file=api&amp;v=2&amp;sensor=true&amp;key=ABQIAAAAbX5c_xOgilOyuvez35DsJhRzRs4soj9rw9WME87m-_RR9yP1hQa80X8sCWiDZkY0dZkA1cBMHBQ"
type="text/javascript"></script>

<script type='text/javascript' language='javascript'>
var map; var panel; var dir; var geopoint=new Array(); var geostring="";
var marker; var count=0; var lat=""; var lon=""; var route=new Array();
var sepname=new Array(); var directions; var zoomLevel = 11;
var infoPoints = []; var routePoints = []; var currentLatLng;

function displayStatus(msg) {
    document.getElementById("message").innerHTML = msg;
}

function createMarker(latLng, tag) {
    var marker = new GMarker(latLng);
    GEvent.addListener(marker, "click", function () {
        marker.openInfoWindowHtml(tag);
    });
    return marker;
}

function InfoPoint(lat, lng, tag) {
    this.latLng = new GLatLng(lat, lng);
    this.tag = tag;
}

function readPoints(location, info_window) {
    var supp_point = new Array();
    supp_point = location.split(',');
    var disaster_lat=document.getElementById('<%=DisasterLat.ClientID%>').value;
    var disaster_lng=document.getElementById('<%=DisasterLng.ClientID%>').value;
    var waypoints = [
    {lat: disaster_lat , lng: disaster_lng , tag: "Disaster Location" },
    {lat: supp_point[0] , lng: supp_point[1], tag: info_window }
    ];
    for (var i = 0; waypoints.length > i; i++) {
        var info_point;
        var lat = waypoints[i].lat;
        var lng = waypoints[i].lng;
        var tag = waypoints[i].tag;
        if (tag.length > 0) {
            info_point = new InfoPoint(lat, lng, tag);
            infoPoints.push(info_point);
        }
        routePoints.push(new GLatLng(lat, lng));
        if (i == (waypoints.length - 1)) {
            currentLatLng = info_point.latLng;
        }
    }
}
</script>
</head>

<body>
</body>
</asp:Content>
// Draw the directions polyline.
function directionsOverlay() {
  var directionsPolyline = directions.getPolyline();
  map.addOverlay(directionsPolyline);
}

function initialize() {
  if (GBrowserIsCompatible()) {
    //map3 is the map for plotting new disaster location in part A
    var map3 = new GMap(document.getElementById("map"));
    map3.centerAndZoom(new GPoint(103.878123,1.461234), 4);
    map3.addControl(new GSmallMapControl());
    GEvent.addListener(map3, 'click', function(overlay, point) {
      if (overlay) {
        map3.removeOverlay(overlay);
      } else if (point) {
        var marker = new GMarker(point);
        map3.addOverlay(marker);
        document.getElementById("latbox").value=point.y;
        document.getElementById("lonbox").value=point.x;
        document.getElementById('<%=txtlat.ClientID%>').value=point.y;
        document.getElementById('<%=txtlon.ClientID%>').value=point.x;
      }
    });
    //map is for the map for calculate the GDirection and distance
    map = new GMap2(document.getElementById("map_canvas"));
    map.addControl(new GLargeMapControl());
    map.addControl(new GMapTypeControl());
    map.addControl(new GScaleControl());
    map.enableScrollWheelZoom();
    map.disableDoubleClickZoom();
    map.enableContinuousZoom();
    map.addControl(new GOverviewMapControl(new GSize(100, 100))); //map overview control
    //panel = document.getElementById("my_div");
    map.setCenter(new GLatLng(1.461234,103.878123),14);
    map.addMapType(G_PHYSICAL_MAP);
    var k=1;
    var point_disaster = document.getElementById('<%=tempDisaster.ClientID%>').value;
    panel = document.getElementById("my_div");
    directions = new GDirections();
    // Draw the polyline when the load event is signaled. The polyline would be
drawn automatically if the GMap2 object were passed to new GDirections but we
don't want the default markers.
    GEvent.addListener(directions, "load", directionsOverlay);
    // Read the waypoints.
    for (i=1;i<=supp_points[0];i++) {
      readPoints(supp_points[i],supplier_name[i]);
    }
    // Center on the current location.
    map.setCenter(currentLatLng, zoomLevel);
    // Display the route. Use an option to preserve the map center and zoom level
    which have already been set. Another option to get polyline even though no
    GMap2 object passed to GDirections.
    directions.loadFromWaypoints(routePoints, {preserveViewport: true, getPolyline: true});
Figure 5.2: code extraction of GIS coordination (routes from disaster location to suppliers) using JavaScript from GMap API

5.5 Summary

In conclusion, the database, user interface and programming development is vital for DMSLS to function accordingly. Development of database has to be data-driven and able to produce valuable information during retrieval. Besides, user interface of the system should be user friendly and reserves its simplicity. The programming development section must produces the required modules output as determined by the user.
CHAPTER VI

System Testing

6.1 Introduction

System testing is performed to DMSLS in order to evaluate the outcome from this system. Procedures for system testing and the findings of it are discussed in this chapter. Program debugging, input and output testing, and user satisfaction testing are reviewed in this chapter too. These steps are carried out so as to access the system and improve it due to any dissatisfactory.

6.2 Program Debugging

Program debugging is performed in the process of finding and fixing the errors in the system. This process took place during the coding stage to allocate and fix coding which cause violation. Microsoft Visual Studio 2008 debugger is used to perform the debugging process. The developer used breakpoints and trace points for debugging purpose. Breakpoints tell the developer that an application should break or pause execution at a certain point. Trace points are a new debugger feature in Visual Studio. A trace point is a breakpoint with a custom action associated with it. Hence, when a trace
point is hit, the specified trace point action will be performed instead of breaking program execution.

Figure 6.1 shows the way to insert a breakpoint in the source view. A red circle will appear when the developer click on the narrow grip strip at the far left of source code tab, indicating a breakpoint has been added.

**Figure 6.1**: Adding breakpoint

Trace point can be added in the source view from breakpoint by right click on the existing breakpoint and select “when hit” tab. Then the developer needed to check the checkbox to print a message when that breakpoint was traced in debugging mode. Figure 6.2 illustrates the process of adding a trace point.
6.3 Input Testing

Input testing took place for the purpose of validate the input keyed in or value chosen by the user for each input field. Available input fields in the system include textbox, dropdown list, text area and others. Input testing took place at both client side and server side. The client side involved the process using validation control before the input data was actually sent to the database at server side. The server side input testing took place using visual basic code behind to validate the input data. Figure 6.3 illustrates the process of client side input testing while figure 6.4 illustrates the process of server side input testing.
Figure 6.3: Client side input testing

Figure 6.4: Server side input testing
### 6.4 Output Testing

Output testing of the system is required to verify the desired output from the input data. Output testing described whether the server had processed the input data as programmed. User’s satisfactory basically mainly influenced by output testing since a big portion of the system performance evaluation is cored on how data is presented inside the system. An example of output testing can be found at figure 6.5 where output testing of account administration is done. User of different type registered at the main page of logistics support sub module. If the server processed the input data accordingly, the output of user details should be displayed in the account administration page.

![Figure 6.5: Output testing for account administration](image-url)
### 6.4.1 System Testing

System testing in DMSLS is done to validate the system work to meet user’s requirements and module functionality. The system testing is done using component testing and integration testing.

i. **Component Testing**

This method is used on single unit component so as to ensure that the module work correctly independently. Table 6.1 illustrates login module as an example of component testing.

**Table 6.1**: Component testing on login module

<table>
<thead>
<tr>
<th>No</th>
<th>Condition Being Tested</th>
<th>Expected Result</th>
<th>Actual Result</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter wrong username or password or both</td>
<td>Error message prompt out</td>
<td>Similar</td>
<td>Successful</td>
</tr>
<tr>
<td>2</td>
<td>Enter the correct username and password</td>
<td>Login successful</td>
<td>Similar</td>
<td>Successful</td>
</tr>
</tbody>
</table>

ii. **Integration Testing**

Integration testing is carried out in the subsystem of DMSLS between individual modules which also work with other modules. Besides, integration testing also been done during integration between the 3 subsystems of DMS to ensure that the technology value is delivered accordingly.
6.5 Summary

In general, the testing processes of DMSLS have been briefed and done to ensure that the system deliver the right functionalities and user requirements. This phase is crucial so as to incorporate usability of DMSLS in disaster handling issues. DMSLS is said to be a success to achieve the objectives set earlier in the start off of the project.
CHAPTER VII

Conclusion

7.1 Introduction

This chapter intends to conclude the project in a concrete and organized approach. In this project, developer aims to develop a web-based logistics support system for MPPG which stores and retrieve logistics information during disaster. Objectives of the projects are achieved. Advantages, disadvantages and recommendation of project are discussed in the later part of the chapter so as to review the system from those three criteria.
7.2 Advantages of Project System

Through DMSLS, MPPG is able to stores and retrieve logistics information using computerized support. DMSLS also visualizes distance between resources (suppliers, vehicles, organizations) and disaster location. Lastly, DMSLS allows MPPG user to send SMS notification to concerned persons in disaster rescue and recovery process.

In a nutshell, DMSLS replaces the existing manual data handling process with computerized web-based information system thus saving time and lives during disaster time.

7.3 Disadvantages of Project System

The disadvantages of the system are:

i. Record tracking of suppliers, organizations and public users who had been helping during disaster time is not provided by DMSLS. This is due to the time limitation of the developer in developing the project system.

ii. Calculation between many resources points is not available inside the system. This is due to the restriction in Google Map API which does not allow such programming coding.
7.4 Recommendations of Project System

Finally, it is recommended such that the project study should be continued in the following advancement fields:

i. Include record tracking of suppliers, organizations, and public users with involving resources (such as vehicles).

ii. Explore further in the available Google map API documentation or other map API.

7.5 Summary

In a nutshell, DMSLS possess great values from MPPG context since it provides effective, reliable data retrieval. DMSLS also support MPPG using SMS and GMap API. Despite the challenges met, it’s worthwhile to develop DMSLS as the pioneer system in the whole Malaysia for disaster management through logistics support.


Appendix A: Gantt Chart PSM I DMSLS
Appendix B: Gantt Chart PSM II DMSLS

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Planning for PSM II</td>
<td>32 days</td>
<td>Thu 1/2/18</td>
<td>Mon 4/1/18</td>
</tr>
<tr>
<td>2</td>
<td>System Development</td>
<td>45 days</td>
<td>Nov 12/1/09</td>
<td>Fri 12/19</td>
</tr>
<tr>
<td>3</td>
<td>Developing Prototype</td>
<td>45 days</td>
<td>Nov 12/1/09</td>
<td>Fri 12/19</td>
</tr>
<tr>
<td>4</td>
<td>Testing and Implementation</td>
<td>14 days</td>
<td>Mon 2/20/10</td>
<td>Thu 2/25/10</td>
</tr>
<tr>
<td>5</td>
<td>System Integration</td>
<td>5 days</td>
<td>Mon 2/2/10</td>
<td>Fri 2/7/10</td>
</tr>
<tr>
<td>6</td>
<td>User Documentation</td>
<td>4 days</td>
<td>Thu 2/20/10</td>
<td>Fri 2/22/10</td>
</tr>
<tr>
<td>7</td>
<td>Project Plan Report</td>
<td>7 days</td>
<td>Sat 2/9/14</td>
<td>Mon 2/18/14</td>
</tr>
<tr>
<td>8</td>
<td>Project Report Requirement</td>
<td>5 days</td>
<td>Mon 3/1/10</td>
<td>Wed 3/3/10</td>
</tr>
<tr>
<td>9</td>
<td>PM Plan Report</td>
<td>1 day</td>
<td>Mon 3/9/10</td>
<td>Mon 3/9/10</td>
</tr>
<tr>
<td>10</td>
<td>Project Presentation</td>
<td>1 day</td>
<td>Wed 3/11/10</td>
<td>Fri 3/13/10</td>
</tr>
<tr>
<td>11</td>
<td>Project Progression</td>
<td>2 days</td>
<td>Sat 3/28/10</td>
<td>Mon 3/30/10</td>
</tr>
<tr>
<td>12</td>
<td>Project Analysis</td>
<td>1 day</td>
<td>Tue 3/3/10</td>
<td>Fri 3/6/10</td>
</tr>
<tr>
<td>13</td>
<td>Final Report</td>
<td>12 days</td>
<td>Mon 3/25/10</td>
<td>Thu 4/12/10</td>
</tr>
<tr>
<td>14</td>
<td>Final Report Corrected</td>
<td>7 days</td>
<td>Mon 3/29/10</td>
<td>Tue 4/2/10</td>
</tr>
<tr>
<td>15</td>
<td>Gantt Chart Document</td>
<td>5 days</td>
<td>Wed 4/4/10</td>
<td>Wed 4/11/10</td>
</tr>
<tr>
<td>16</td>
<td>Gantt Chart</td>
<td>1 day</td>
<td>Mon 4/9/10</td>
<td>Mon 4/11/10</td>
</tr>
</tbody>
</table>
Appendix C: Entity-Relationship Diagram for DMSLS
Appendix D: Context diagram for Disaster Management System

- MPPG Admin
  - Supplier information
  - Organization, supplier, public, vehicle, volunteer, missing person, found person information, disaster location

- Disaster Management System
  - Public user, vehicle
  - Volunteer
    - Volunteer personal details, skills, restrictions, preferences, and availability duration
    - Most suitable disaster type and disaster resources
    - Resources, missing person, found person
    - Supplier information
    - Organization, vehicle

- Organization
  - Supplier, disaster notification
  - Organization, vehicle, disaster notification

- Supplier
  - Visitor
    - Resources, missing person, found person

- Public user
  - Public user, vehicle

- Visitor
Appendix E: Diagram level 0 of Disaster Management System

1.0 Logistics Support

2.0 Volunteer Management Solution

3.0 Missing Person Information Solution

4.0 Generate Report

GIS Information
Organization
Supplier
Public user
Volunteer
Vehicle
Disaster report
VictimsDetails
VolunteerProfile
AdminInfo
ForumInfo

MPPG Admin
Public user
Volunteer
Supplier
Visitor
Organization
Appendix F: DFD level 1 for DMSLS

- 1.1 Retrieve organization, supplier, vehicle, public information
- 1.2 Coordination/ GIS mapping processing
- 1.3 Disaster SMS notification
- 1.4 Register Supplier, Organization, vehicle and public users’ information
- 1.5 Report Module

Flow diagram:
- MPPG admin
- Request report
- Organization
- Profile information
- Volunteer
- Location map
- Supplier
- Supplier information
- Nearest resources coordination information
- Disaster scenario Requested profile information
- Coordinate of disaster scenario
- Request report
- Public user
- Vehicle
- Disaster Profile/
- Public information
- Notification

Legend:
1.1 Retrieve organization, supplier, vehicle, public information
1.2 Coordination/ GIS mapping processing
1.3 Disaster SMS notification
1.4 Register Supplier, Organization, vehicle and public users’ information
1.5 Report Module

Notifications:
- Request report
- Request report
- Notification
- Notification
- Notification
Appendix G: DFD level 2 Retrieve organization, supplier, vehicle, public information
Appendix H: DFD level 2 Coordination/ GIS mapping processing

2.1 Coordination/ GIS mapping processing
   - Disaster Scene coordinate
   - Distance resources and disaster scene
   - Visualized route mapping
   - Distance resources and disaster location

2.2 Calculate distance between resources and the scene
   - Disaster Scene coordinate

2.3 Produce map to disaster location
   - Disaster/ resources location map

MPPG admin

volunteer
Appendix I: DFD level 2 Disaster SMS notification

1. MPPG admin
   - Disaster news

2. 3.1
   - Trigger notification
   - Sms disaster news

3. 3.2
   - Send notification through SMS gateway
   - Supplier, organization, public hp no

4. Supplier
5. Organization
6. Public user
Appendix J: DFD level 2 Register Supplier, Organization and public users’ Information

Supplier

4.1
Register Supplier and vehicle Information

Supplier details, suppliers’ vehicle

4.2
Register Organization and vehicle Information

Organization details, organizations’ vehicle

Organization and vehicle information

4.4
Insert Supplier, Organization, public users’ information into system

Processed Supplier, organization, public, vehicle information

Supplier and vehicle information

Organization and vehicle information

Public user

4.3
Register public users and vehicle Information

Public details, publics’ vehicle

Public user and vehicle information

vehicle

supplier

organization

Public user
Appendix K: DFD level 2 Report Module

MPPG admin

Requested real time report
Request report

5.1 Process report

Profile information

organization
Supplier
Public user
Vehicle
<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Data flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPPG admin</td>
<td>Logistics Support</td>
<td>Organization, supplier, public user, vehicle, disaster location, GIS information</td>
</tr>
<tr>
<td>MPPG admin</td>
<td>Volunteer management system</td>
<td>Volunteer application result</td>
</tr>
<tr>
<td>MPPG admin</td>
<td>Missing person information system</td>
<td>Missing person, found person information</td>
</tr>
<tr>
<td>MPPG admin</td>
<td>Generate report</td>
<td>Request report</td>
</tr>
<tr>
<td>Logistics Support</td>
<td>MPPG admin</td>
<td>Organization, supplier, public user, vehicle, GIS information</td>
</tr>
<tr>
<td>Volunteer management system</td>
<td>MPPG admin</td>
<td>Volunteer profile info, volunteer skill disaster list</td>
</tr>
<tr>
<td>Missing person information system</td>
<td>MPPG admin</td>
<td>Missing person, found person</td>
</tr>
<tr>
<td>Generate report</td>
<td>MPPG admin</td>
<td>Organization, supplier, public user, vehicle, disaster location, missing person, found person, volunteer profile info, volunteer skill disaster list</td>
</tr>
<tr>
<td>Public user</td>
<td>Logistics Support</td>
<td>Vehicle, public user information</td>
</tr>
<tr>
<td>Public user</td>
<td>Missing person information system</td>
<td>Missing person, found person</td>
</tr>
<tr>
<td>Logistics Support</td>
<td>Public user</td>
<td>Vehicle, public user information</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Missing person information system</td>
<td>Public user</td>
<td>Missing person, found person</td>
</tr>
<tr>
<td>Volunteer</td>
<td>Volunteer management system</td>
<td>Volunteer profile, volunteer skill</td>
</tr>
<tr>
<td>Missing person information system</td>
<td>Volunteer</td>
<td>Missing person</td>
</tr>
<tr>
<td>Logistics Support</td>
<td>Volunteer</td>
<td>Disaster location map, vehicle, supplier, organization</td>
</tr>
<tr>
<td>Supplier</td>
<td>Logistics Support</td>
<td>Supplier information</td>
</tr>
<tr>
<td>Logistics Support</td>
<td>Supplier</td>
<td>Supplier, notification</td>
</tr>
<tr>
<td>Organization</td>
<td>Logistics Support</td>
<td>Organization, vehicle information</td>
</tr>
<tr>
<td>Logistics Support</td>
<td>Organization</td>
<td>Organization, vehicle information, notification</td>
</tr>
<tr>
<td>Visitor</td>
<td>Volunteer management system</td>
<td>Volunteer application</td>
</tr>
<tr>
<td>Volunteer management system</td>
<td>Visitor</td>
<td>Volunteer application result</td>
</tr>
<tr>
<td>Missing person information system</td>
<td>Visitor</td>
<td>Missing person, found person</td>
</tr>
<tr>
<td>Volunteer management system</td>
<td>Logistics Support</td>
<td>Volunteer profile, volunteer skill disaster list</td>
</tr>
<tr>
<td>Logistics Support</td>
<td>Volunteer management system</td>
<td>Disaster location map, supplier, organization profile</td>
</tr>
<tr>
<td>Missing person information system</td>
<td>Volunteer management system</td>
<td>Missing person, found person</td>
</tr>
</tbody>
</table>
## Appendix N: Database Design

### Table Login_Logistic

Table Description: Store the username, password, user types of system users

Primary Key: ID

Foreign Key: TypeID, Type

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Username</td>
<td>Varchar</td>
<td>15</td>
<td>Username</td>
</tr>
<tr>
<td>Password</td>
<td>Varchar</td>
<td>15</td>
<td>Password</td>
</tr>
<tr>
<td>Type</td>
<td>Varchar</td>
<td>10</td>
<td>Foreign key for user type</td>
</tr>
<tr>
<td>TypeID</td>
<td>Int</td>
<td>-</td>
<td>Foreign key for user type</td>
</tr>
<tr>
<td>Approve</td>
<td>Varchar</td>
<td>20</td>
<td>Approval to login into system</td>
</tr>
</tbody>
</table>
**Table Org**

Table Description: Store the organization profile information

Primary Key: Org_ ID

Foreign Key: -

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org_ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Org_Name</td>
<td>Varchar</td>
<td>30</td>
<td>Organization name</td>
</tr>
<tr>
<td>Org_Reg_No</td>
<td>Varchar</td>
<td>30</td>
<td>Organization Registered Number</td>
</tr>
<tr>
<td>Org_Add</td>
<td>Text</td>
<td>-</td>
<td>Organization Address</td>
</tr>
<tr>
<td>Org_Desc</td>
<td>Text</td>
<td>-</td>
<td>Organization Description</td>
</tr>
</tbody>
</table>
**Table Contact Org**

Table Description: Store the contact person information of organization

Primary Key: Contact_ID

Foreign Key: OrgID

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact_ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Contact_Name</td>
<td>Varchar</td>
<td>30</td>
<td>Contact Person Name</td>
</tr>
<tr>
<td>Contact_HP</td>
<td>Varchar</td>
<td>15</td>
<td>Contact Person HP Number</td>
</tr>
<tr>
<td>Contact_Add</td>
<td>Text</td>
<td>-</td>
<td>Contact Person Address</td>
</tr>
<tr>
<td>Contact_Email</td>
<td>Varchar</td>
<td>30</td>
<td>Contact Person Email</td>
</tr>
<tr>
<td>OrgID</td>
<td>Int</td>
<td>-</td>
<td>Foreign key for Organization</td>
</tr>
</tbody>
</table>
**Table OrgGMap**

Table Description: Store the latitude and longitude of organization coordination

Primary Key: ID

Foreign Key: Org_ID

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Org_ID</td>
<td>Int</td>
<td>-</td>
<td>Foreign key for Organization</td>
</tr>
<tr>
<td>Latitude</td>
<td>Float</td>
<td>-</td>
<td>Organization Latitude coordination</td>
</tr>
<tr>
<td>Longitude</td>
<td>Float</td>
<td>-</td>
<td>Organization Longitude coordination</td>
</tr>
</tbody>
</table>
**Table Public_User**

Table Description: Store the public user profile information

Primary Key: Public_ID

Foreign Key: -

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public_ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Public_Name</td>
<td>Varchar</td>
<td>30</td>
<td>Public user name</td>
</tr>
<tr>
<td>Public_Add</td>
<td>Text</td>
<td>-</td>
<td>Public user address</td>
</tr>
<tr>
<td>Public_Hp</td>
<td>Varchar</td>
<td>15</td>
<td>Public user HP Number</td>
</tr>
<tr>
<td>Public_Email</td>
<td>Varchar</td>
<td>30</td>
<td>Public user email</td>
</tr>
</tbody>
</table>
### Table Disaster_Location

Table Description: Store the disaster location coordination

Primary Key: ID

Foreign Key: -

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Lat</td>
<td>Float</td>
<td>15</td>
<td>Disaster location Latitude coordination</td>
</tr>
<tr>
<td>Lon</td>
<td>Float</td>
<td>15</td>
<td>Disaster location Longitude coordination</td>
</tr>
<tr>
<td>Date</td>
<td>Varchar</td>
<td>15</td>
<td>Disaster happening date</td>
</tr>
<tr>
<td>Disaster_Desc</td>
<td>Text</td>
<td>-</td>
<td>Disaster description</td>
</tr>
<tr>
<td>Ref_ID</td>
<td>Varchar</td>
<td>15</td>
<td>Disaster reference ID</td>
</tr>
</tbody>
</table>
**Table Supp**

Table Description: Store the supplier profile information

Primary Key: Supp_ID

Foreign Key: -

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supp_ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Supp_Name</td>
<td>Varchar</td>
<td>30</td>
<td>Supplier name</td>
</tr>
<tr>
<td>Supp_Reg_No</td>
<td>Varchar</td>
<td>30</td>
<td>Supplier Registered Number</td>
</tr>
<tr>
<td>Supp_Add</td>
<td>Text</td>
<td>-</td>
<td>Supplier Address</td>
</tr>
<tr>
<td>Supp_Desc</td>
<td>Text</td>
<td>-</td>
<td>Supplier Description</td>
</tr>
</tbody>
</table>
### Table ContactSupp

Table Description: Store the contact person information of supplier

Primary Key: Contact_ID

Foreign Key: Supp_ID

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact_ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Contact_Name</td>
<td>Varchar</td>
<td>30</td>
<td>Contact Person Name</td>
</tr>
<tr>
<td>Contact_HP</td>
<td>Varchar</td>
<td>15</td>
<td>Contact Person HP Number</td>
</tr>
<tr>
<td>Contact_Add</td>
<td>Text</td>
<td>-</td>
<td>Contact Person Address</td>
</tr>
<tr>
<td>Contact_Email</td>
<td>Varchar</td>
<td>30</td>
<td>Contact Person Email</td>
</tr>
<tr>
<td>Supp_ID</td>
<td>Int</td>
<td>-</td>
<td>Foreign key for Supplier</td>
</tr>
</tbody>
</table>
**Table SuppGMap**

Table Description: Store the latitude and longitude of supplier coordination

Primary Key: ID

Foreign Key: Supp_ID

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Supp_ID</td>
<td>Int</td>
<td>-</td>
<td>Foreign key for supplier</td>
</tr>
<tr>
<td>Latitude</td>
<td>Float</td>
<td>-</td>
<td>Supplier Latitude coordination</td>
</tr>
<tr>
<td>Longitude</td>
<td>Float</td>
<td>-</td>
<td>Supplier Longitude coordination</td>
</tr>
</tbody>
</table>
**Table Vehicle**

Table Description: Store the registered vehicle information inside the system

Primary Key: Vehicle_ID

Foreign Key: Cont_ID, Cont_Type

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle_ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Vehicle_Reg_No</td>
<td>Varchar</td>
<td>15</td>
<td>Vehicle Registered Number</td>
</tr>
<tr>
<td>Vehicle_Category</td>
<td>Varchar</td>
<td>30</td>
<td>Vehicle Category</td>
</tr>
<tr>
<td>Vehicle_Capacity</td>
<td>Varchar</td>
<td>30</td>
<td>Vehicle Capacity</td>
</tr>
<tr>
<td>Vehicle_Weight</td>
<td>Varchar</td>
<td>30</td>
<td>Vehicle Weight</td>
</tr>
<tr>
<td>Vehicle_Add</td>
<td>Text</td>
<td>-</td>
<td>Vehicle Address</td>
</tr>
<tr>
<td>Vehicle_Desc</td>
<td>Text</td>
<td>-</td>
<td>Vehicle Description</td>
</tr>
<tr>
<td>Cont_ID</td>
<td>Int</td>
<td>-</td>
<td>Foreign key for Contact Person ID</td>
</tr>
<tr>
<td>Cont_Type</td>
<td>Varchar</td>
<td>15</td>
<td>Foreign Key for Contact Person Type</td>
</tr>
<tr>
<td>From_Date</td>
<td>Varchar</td>
<td>30</td>
<td>Available from date</td>
</tr>
<tr>
<td>Till_Date</td>
<td>Varchar</td>
<td>30</td>
<td>Available till date</td>
</tr>
</tbody>
</table>
**Table VehicleGMap**

Table Description: Store the latitude and longitude of the vehicle registered inside the system

Primary Key: ID

Foreign Key: Vehicle_ID

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Data Type</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Int</td>
<td>-</td>
<td>Auto increment primary key</td>
</tr>
<tr>
<td>Vehicle_ID</td>
<td>Int</td>
<td>-</td>
<td>Foreign key for vehicle</td>
</tr>
<tr>
<td>Latitude</td>
<td>Float</td>
<td>-</td>
<td>Latitude coordination of vehicle address</td>
</tr>
<tr>
<td>Longitude</td>
<td>Float</td>
<td>-</td>
<td>Longitude coordination of vehicle address</td>
</tr>
</tbody>
</table>
Appendix O: PAGEMA Interview Minute

Date: January 15, 2010
To: Pasir Gudang Emergency Mutual Aid (PAGEMA)
Subject: Interview Report in the field of Disaster

Introduction
The purposes of this interview are:
   i. to collect information for better understanding of logistic, participation of volunteer and also missing people process during disaster.
   ii. to find out the criteria of health and food safety during a flood.
   iii. to obtain actions that should be taken during fire as well as the criteria of action taken.

Interview Summary
   • Process of Logistic
      - Stage manager and logistic manager are responsible in logistic process. It is subject by the Incident Command System (ICS).
      - Issue of logistic: Strategic Area
      - Current Method of Logistic Operation:
         (a) established operation room
            - Municipal Stadium
            - Post Incident Site
            - Post Disaster Control
         (b) Figure 1.1 shows the operation procedure of disaster happens in any factory.
- **Relations with Resources Supplier**

  (a) Memorandum Of Understanding (MOU)

  - In the history of Act MOU, PAGEMA has the right to acquire any necessary resources from communities during disaster. Expenses of resources will be paid to communities according to the prescribed insurance.

  (b) Internal Organisation

  - High-risk factories
  - Government Organisation (i.e. fireman, police)

- **Process of volunteer participation**

  - Community does not help anything during disaster. Only PAGEMA’s members such as RELA, ST.JOHN and JPA3 members are volunteers.

  - If any disaster happens, MPPG will call or SMS PAGEMA to ask for help from their members. In addition, PAGEMA usually ask help from their organizations’ members in manual way which is by calling them to tell the details about the disaster’s location and the victims.

  - PAGEMA did launch their website which contains information about “Emergency Preparedness” and “How to protect yourself” during disaster.
- There is no term and condition to become a volunteer in MPPG and PAGEMA. This is because only their well-trained members will be the volunteers during disaster.
- MPPG and PAGEMA help the organizations who are registered under them only. Most of the organizations are located in Pasir Gudang areas which is the high-risk areas in Johor.
- There is no formal system in managing volunteers’ information in MPPG and PAGEMA.
- PAGEMA focused in Port (chemical emergencies & gas release).
- Volunteers need to use PPE in helping disaster’s victims and to move them from the disaster’s location.

- **Process of missing people**
  - Normally, JPA is take part in finding missing people.
  - Victims of disaster will be gathered at assembly area of municipal stadium.

- **Health and food safety during flood**
  - Cup noodles, biscuits, fried noodles will be given to victims. The type of food given to victims is selected based on the budget of PAGEMA.
  - Normally, the headman of village will record the needed food for PAGEMA.
  - In past flood, most of the villagers (victims) will prepare the cooking tools and so on.
  - So far, helicopter is not use in relocate victims to a safe place. PAJERO, lorry is the selected vehicle to move the victims.
  - Boiled the water, store the water in bottle water and food in container in order to keep the water and food safely.
  - Community gets the information of health and food safety from the internet, booklet, experience, and media.
  - To prevent the disease happens during flood, vaccines, method for prepare the food and water and also method for keeping personal hygiene will be given to victims.
- Government health centre is fully responsible in take care the health of victims. And PAGEMA will always stand by for emergency case. For instance, they receive any call for sending victim to hospital.

- Appropriate actions during fire
  - We can interview Tuan Hamid (Johor State) or Mr. Salleh (Pasir Gudang area) for getting more information.
  - Mr. Roslan will try to obtain the number of Pasir Gudang Communities.
  - Normally, scenario and video will be given in drill. Drill is participating by high-risk industries in Pasir Gudang.
  - As an expert, they will inform their family members and giving some instruction when any disaster occurs in Pasir Gudang. They will ask their family members do not panic, inform neighbor and also take the important personal belongings.
  - According to Mr Roslan, the communities of Pasir Gudang understood the risk of their living area. But he is not so sure whether communities know the actions that should be taken during fire.
  - Mr Roslan will return the pilot questionnaire to Dr Zaidi during this few weeks.
Appendix P: USER MANUAL

USER MANUAL
DISASTER MANAGEMENT SYSTEM: LOGISTICS SUPPORT

May, 2010
1.0 GENERAL INFORMATION

1.1 SYSTEM OVERVIEW

- **System Title:** Disaster Management System: Logistics Support
- **Purpose of the system:** The purpose of this project is to develop a web-based disaster management logistic system for MPPG. The highlighted feature of this system is the effectiveness of data connection and retrieval on disaster recovery and managing issues.

- **System Category:**
  - Major Applications
  - Retrieve organization, supplier, vehicle, public users information Module:
    - Information of organization, supplier, vehicle can be retrieved for usage during disaster time.
  - Register Supplier, Organization, vehicle and public users’ information
    - Supplier, organization, vehicle and public users’ information is registered into the system for usage upon request.
  - Coordination/ GIS mapping processing
    - Google Map API is used to process and visualize the coordination of disaster location and distance to available resources (suppliers, organizations, vehicles).
  - Disaster notification
    - Notification of disaster is done by integrating SMS API into the system. SMS can be sent from the website upon request.
  - Report Module
    - Information of suppliers, organizations, and vehicles can be transformed into report inside the system. Report is generated upon request at real time.
2.0 SYSTEM SUMMARY

2.1 USER ACCESS LEVELS

There are 4 types of users in DMSLS namely MPPG, Supplier, Organization, and Public Users.

3.0 GETTING STARTED

3.1 REGISTER

Before any operation being performed using the system, a user have to register at the Login_Logistic.aspx page. A user can choose to register as organization, supplier or public user.

For example the user clicks on the “Register As Public User” hyperlink. A popup window will show up for the registration process of the user.
3.2 LOGGING ON

To access the system, a user needs to logging into the system. The four types of users will each be directed to identity master page. Different master page will provide different menu dashboards at the left side bar of the system where the user can access functions from. A message box will popup indicate the login successful if username and password can be found in the database.

Else if the username or password inserted is not found in the database, an error message box popup.
3.3 SYSTEM MENU

3.3.1 REGISTER MODULE

Supplier, organization and public users can register their login account at the Login_Logistic.aspx. Organization and supplier then will login and register their contact person information. Menu at the side bar of the masterpage can be found. User can click on the required link for the function. Organization can register contact person by clicking “Add Contact Person” menu.

Location of vehicle can be registered using Google Map API integrated inside the DMSLS. Organization user can click on the “Vehicle List”, a list of registered vehicles will be shown
<table>
<thead>
<tr>
<th><strong>Vehicle Profile</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle ID</td>
<td>1</td>
</tr>
<tr>
<td>Register Number</td>
<td>JHL2123</td>
</tr>
<tr>
<td>Category</td>
<td>lory</td>
</tr>
<tr>
<td>Capacity</td>
<td>2</td>
</tr>
<tr>
<td>Weight</td>
<td>1000</td>
</tr>
<tr>
<td>Address</td>
<td>51, Jln Aslan, Tmn Kepok</td>
</tr>
<tr>
<td>Description</td>
<td>2006 engine</td>
</tr>
<tr>
<td>valid from date</td>
<td>3/3/2010</td>
</tr>
<tr>
<td>valid till date</td>
<td>3/12/2010</td>
</tr>
<tr>
<td>Contact ID</td>
<td>1</td>
</tr>
<tr>
<td>Contact Type</td>
<td>Org</td>
</tr>
<tr>
<td>Edit Delete</td>
<td></td>
</tr>
<tr>
<td>Vehicle Mapping</td>
<td>View</td>
</tr>
</tbody>
</table>

12
Click on the “View” hyperlink under the “Vehicle Mapping”. Organization then can view and update the vehicle location as illustrated below:

Click on the map on a location and it will provide you with the latitude and longitude in the input boxes below. To remove a point, click it again.

Lat: 1.461184400633518
Lon: 103.88843550877577
3.3.2 RETRIEVE INFORMATION MODULE

Profile information of each users can be retrieved at their identical masterpage after they login. Every users can update their profile information. MPPG hold the key to administer all available information inside the database. MPPG user can view a list of organization, supplier, public users and disaster location by clicking at the identity menu at side bar. For example, MPPG can click on the “Supplier List” to view a list of registered suppliers inside the system.
3.3.3 DISASTER NOTIFICATION MODULE

SMS API was integrated in the system. MPPG user can send SMS notification to public user, supplier or organization by clicking “Send SMS” menu at side bar. MPPG user will select the users to be sent SMS. Follow up, MPPG user will type the text message to be sent in the text area provided. Sent Result with value bigger than 0 indicate the message has been sent successfully. The message will be sent to the selected user after the “Submit” button is clicked. Example of sending SMS to the public user is illustrated as below:

<table>
<thead>
<tr>
<th>Public ID</th>
<th>Name</th>
<th>Address</th>
<th>HP No</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 1</td>
<td>zhaosaint</td>
<td>ktr</td>
<td>60168161042</td>
<td><a href="mailto:zhaosaint@hotmail.com">zhaosaint@hotmail.com</a></td>
</tr>
<tr>
<td>Select 7</td>
<td>Dr Zaidi</td>
<td>utma</td>
<td>60179114820</td>
<td>email.com</td>
</tr>
</tbody>
</table>

Destination No: 60168161042, 60179114820,

Text Msg:

Submit

Sent Result:
3.3.4 COORDINATION/ GIS MAPPING MODULE

Inside this module, MPPG user is able to overview resources with the help of GMap mapping. To view available resources, MPPG user has to click on the “Google Map” menu at side bar. Registered organization, supplier and vehicles will be marked accordingly as shown in example below:

- Green Marker ----> Vehicle location
- Blue Marker ----> Supplier location
To visualize the route distance between the available resources and the disaster location, MPPG user has to click on the “Calculate Distance” menu at side bar. There are several steps to be followed inside this sub module. Firstly, a MPPG user can define a new disaster location by just clicking on the map and input the required data.

Step 1: Define the disaster location

A New disaster location

Click the ‘Update location’ button below to update your location!

- Latitude: 1.461020421223182
- Longitude: 103.89628887173514
- Date: 3/12/2010
- Reference ID: A003
- Description: Fire disaster

Update Location
Next is to select the disaster location from the database by just clicking on the specific row of the disaster location table. The location of the disaster will be visualized on the map under the disaster location table.

<table>
<thead>
<tr>
<th>ID</th>
<th>Lat</th>
<th>Lon</th>
<th>Date</th>
<th>Description</th>
<th>Reference ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>1.46571382709526</td>
<td>103.895439148473</td>
<td>3/6/2010</td>
<td>banjar</td>
<td>001</td>
</tr>
<tr>
<td>22</td>
<td>1.444336603861149</td>
<td>103.86106873062</td>
<td>3/13/2010</td>
<td>banjar</td>
<td>002</td>
</tr>
<tr>
<td>24</td>
<td>1.47114798635575</td>
<td>103.879943848879</td>
<td>3/5/2010</td>
<td>explosion</td>
<td>004</td>
</tr>
<tr>
<td>25</td>
<td>1.48969838820891</td>
<td>103.90895173113</td>
<td>3/4/2010</td>
<td>Fire</td>
<td>009</td>
</tr>
</tbody>
</table>
| 26 | 1.47558963569334  | 103.901988028956 | 3/24/2010 | jalan besar | jln112
MPPG user then has to choose which logistic category to be visualized (supplier, vehicle or organization).

The system will then visualize the route distance between resources and the disaster location.
To calculate distance between points, user has to define the two points by clicking on the map. A small green marker will appear indicating a point defined.
Click on the “Calculate Distance” button to calculate the disaster and click on the “reload” button to reload the map. The distance and direction between the two points will be shown at the right hand side of the map.
3.3.5 ADMINISTRATION MODULE

MPPG owns the right to administer the users of the system. MPPG achieve this by clicking on the “Account Administration” menu at side bar. Only the user with the “Approve” column having the value “Approved” is being allowed to login into the system. MPPG can monitor and modified the value to control access of users.

User List

<table>
<thead>
<tr>
<th>ID</th>
<th>Username</th>
<th>Password</th>
<th>Type</th>
<th>TypeID</th>
<th>Approve</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Public</td>
<td>Public123</td>
<td>Public Use</td>
<td>1</td>
<td>Approved</td>
</tr>
<tr>
<td>5</td>
<td>Organization</td>
<td>Org331</td>
<td>Org</td>
<td>1</td>
<td>Approved</td>
</tr>
<tr>
<td>6</td>
<td>Supplier</td>
<td>Supp432</td>
<td>Supp</td>
<td>1</td>
<td>Approved</td>
</tr>
<tr>
<td>7</td>
<td>choosant</td>
<td>123</td>
<td>MPPG</td>
<td>1</td>
<td>Not Approved</td>
</tr>
<tr>
<td>9</td>
<td>Benji</td>
<td>ben45</td>
<td>Org</td>
<td>33</td>
<td>Approved</td>
</tr>
</tbody>
</table>

3.3.6 LOGOUT SYSTEM

To logout from the system, the user needed only to click on the “Logout” hyperlink at the side bar. Session will be cleared and user need to login to access the system again. A message box will prompt out indicating the logout successful.
4.0 REPORTING

MPPG user can also view the logistics report by selecting the “Generate Report” menu link button. An example of vehicle report according to user type is shown below: