THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Master of Engineering (Engineering Management)
DEGREE

Engineering Management	Industrial Engineering
FIELD	**DEPARTMENT**

TITLE: The Development of Web-based GIS Application for Facility Location Problem

NAME: Mr. Kasiphop Prasatsisuparp

THIS THESIS HAS BEEN ACCEPTED BY

________________________ **THESIS ADVISOR**
(Mr. Pornthep Anussornnitisarn, Ph.D.)

________________________ **THESIS CO-ADVISOR**
(Associate Professor Kongkiti Phusavat, Ph.D.)

________________________ **DEPARTMENT HEAD**
(Associate Professor Kongkiti Phusavat, Ph.D.)

APPROVED BY THE GRADUATE SCHOOL ON ________________________

________________________ **DEAN**
(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

THE DEVELOPMENT OF WEB-BASED GIS APPLICATION FOR FACILITY LOCATION PROBLEM

KASIPHOP PRASATSISUPARP

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Engineering (Engineering Management)
Graduate School, Kasetsart University
2010

Kasiphop Prasatsisuparp 2010: The Development of Web-based GIS Application for Facility Location Problem. Master of Engineering (Engineering Management), Major Field: Engineering Management, Department of Industrial Engineering. Thesis Advisor: Mr. Pornthep Anussornnitisarn, Ph.D. 71 pages.

GIS (Geographic Information System) is quite popular. Researcher studies GIS worked with Operational Research problem. This research proposes initial idea to create a web application using free map information from Google Map to solve the Operation Research problem. Normally, the map information is not distributed free of charge and takes time to request. Facility Location Problem is selected and proposed the solution how to develop the web-based GIS application using map information from Google Map which is free and can be requested online anytime.

____________________	____________________	___ / ___ / ___
Student's signature	Thesis Advisor's signature	

ACKNOWLEDGEMENTS

I would like to grateful thank to my thesis advisor, Mr. Pornthep Anussornnitisarn, Ph.D. for his suggestion, encouragement and enthusiasm during study and research work. This research was supported and partially funded by the International Graduate Program in Industrial Engineering. Finally, I am especially appreciated my parent, my sister and my girl friend for their continuing encouragement.

Kasiphop Prasatsisuparp
February 2010

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF ABBREVIATIONS

GIS	=	Geographic Information System
CFLP	=	Capacitated Facility Location Problem
UFLP	=	Uncapacitated Facility Location Problem
API	=	Application Programming Interface
XML	=	Extension Markup Language
AMPL	=	A Modeling Language for Mathematical Programming
ASP.NET	=	Active Server Page . NET
ER	=	Entity-Relationship

THE DEVELOPMENT OF WEB-BASED GIS APPLICATION FOR THE FACILITY LOCATION PROBLEM

INTRODUCTION

Nowadays, GIS (Geographic Information System) is quite popular. GIS allows us to view, understand question, interpret and visualize data in many ways that reveal relationships, patterns and trends in the form of maps, globes, reports and charts. Many articles research in GIS in different ways. Nonetheless, there are not many visualize applications that are provided in Operational Research. The minimum requirement to create a GIS for Operational Research is map source. Fortunately, there are many free map sources from internet such as Google Earth, Google Map, Yahoo Map, etc.

There are many Operational Research problems such as Transportation problem, Facility Location problem, Traveling Salesman problem, Shortest Path problem, Vehicle routing problem, etc. To make problems related in GIS, the problem should be formulated in term of longitude, latitude, distance, population or information which GIS can provide, etc. The important information from GIS is latitude and longitude. Most of Operational Research problem use them as fixed information in the formula. The table below shows the example of Operational Research problem.

Table 1 Example of Operational Research problem

Problem	Fixed Information	Variable
Transportation	- Shipping cost per km from each plant to each customer - Distance from each plant to each customer - Customer Demand - Plant capacities	- Number of units shipped from each plant to each customer
Facility Location Problem	- Fixed or installation cost - Cost of manufacturing one unit of product at a location - Cost of shipping	- Open/Close variable for Factory and Customer

Table 1 (Continued)

Problem	Fixed Information	Variable
The Plant-Location Problem	- Shipping cost per km per product unit from Plants to Retail stores - Distance from Plants to Retail stores - Plant Capacities - Fixed costs	- no/yes decision for each plant - Number of units shipped from Plants to Retail Stores
The Travaling-Saleman	- Location of each node	
Shortest Path	- Location of each node	
Vehicle Routing	- Job that assign on trunk - Position that job occupied in tour - Number of trunk - Actual number of locations - Upper bound of number of location visited daily - Distance between node - Truck capacity	- Node assigned to trunk

The Facility Location problem is selected to study in this research due to it uses fully main fixed information from the map source such as latitude, longitude, distance and there are no research about this problem in our faculty. The facility location problem can be classified roughly in two types which are simple facility location problem and capacitated facility location problem (CFLP). For the simple facility location problem, some researcher called uncapacitated facility location problem (UFLP).

The CFLP is much more difficult to solve than the UFLP – and much less integer-friendly. Although the issue has never been investigated in depth, the results of Heller (1985) and Heller *et al.* (1989) suggest that the extent to which the CFLP can be solved without much resort to branch-and-bound is dependent on the tightness of the capacity constraints.

This research proposes the CFLP solution of implementing application connected to the well-known source map “Google Map”. It is free to access and get information. The web application technology has been selected for this research and the data for the problem has been kept in database system. To solve the problem, the fixed information and variable must be imported from the application and solve the problem by external application as Figure 1.

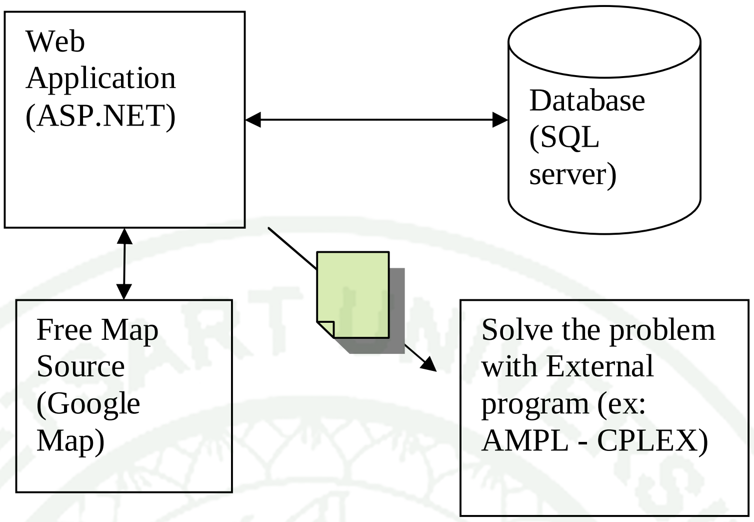

Web Application (ASP.NET)
Database (SQL server)
Free Map Source (Google Map)
Solve the problem with External program (ex: AMPL - CPLEX)

Figure 1 The environment and flow control of proposed solution

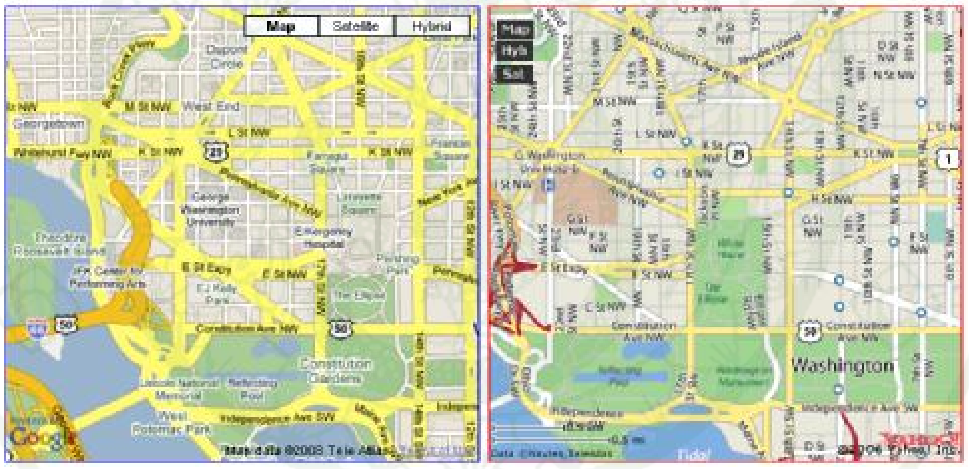


Figure 2 Google Map and Yahoo Map comparison

Google Map is a free web mapping service application and technology provided by Google that powers many map-base services including the Google Maps website. It offers street maps, a route planner, position of place and an urban business locator for numerous countries around the world, etc. Google Maps feature a map that users can zoom by using the mouse wheel or its control and drag the object. User can enter address, ZIP code, point (latitude and longitude) to quickly find the position on the map. Moreover, the Google Maps can generate driving direction between any pair of locations in some countries.

For external program as Figure 1, A Modeling Language for Mathematical Programming, AMPL, is a comprehensive, powerful, algebraic modeling language for problem in linear, nonlinear, and integer programming. AMPL is based upon modern modeling principles and utilizes and advanced architecture providing flexibility most other modeling systems lack. AMPL's solver interface supports linear, nonlinear, and mixed integer models. This interface is rich enough to support many of the features used by advanced solvers to improve performance and solution accuracy, such as piecewise-linear constructs, representation of network problem, and automatic differentiation of nonlinear functions. You can choose a solver for a particular problem by giving its executable filename in the AMPL option solver. This research uses CPLEX, MINTO as solver. It is designed to solve linear programs, as well as the integer program. Integer programs may be pure (all integer variables) or mixed (some integer and some continuous variables); integer variables may be binary (taking values 0 and 1 only) or may have more general lower and upper bounds.

OBJECTIVES

Main Objective

Our purposes are:

1. To study the facility location problem and select the general problem for web application development.
2. To study the free map source.
3. To propose the solution of application to solve the facility location problem

Scopes

We assume the following conditions:

1. We study only the facility location problem.
2. The research is for proposed the idea for further development.

Benefits

Propose the initial idea to develop operational research application with free map resource.

LITERATURE REVIEW

Facility Location problem (FLP)

The general facility location problem is given a set of facility locations and set of customers who are served from the facilities to find which facilities should be opened and which customers should be served from which facilities. The objective is to minimize the total cost of serving all the customers. The figure below shows a graphical representation of general facility location problem.

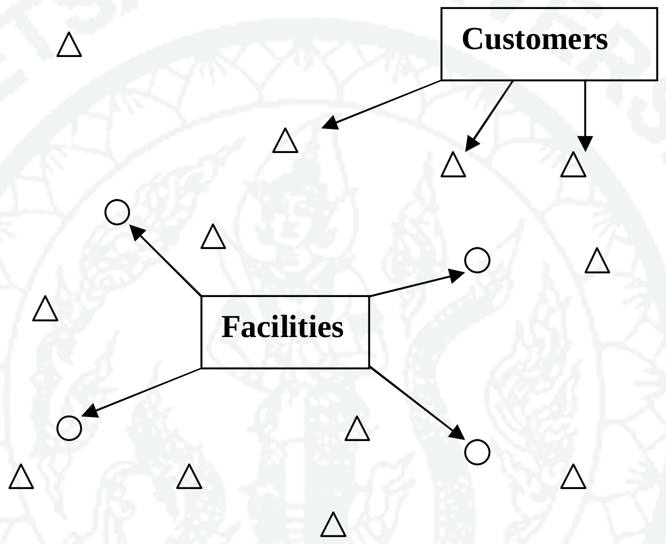


Figure 3 Facilities and Customers of facility location problem

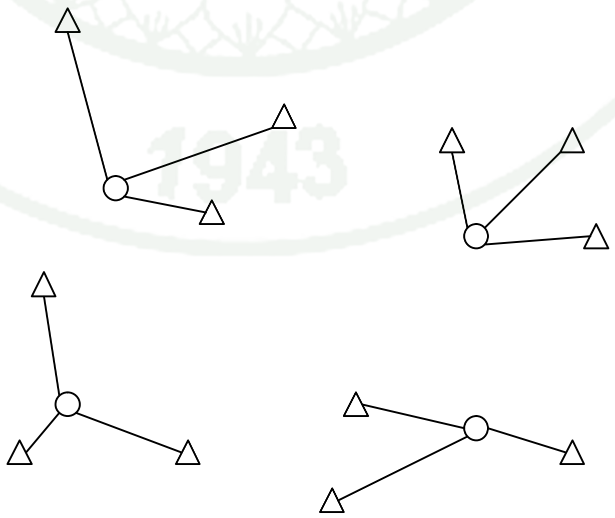


Figure 4 Possible solution of facility location problem

The “simple” or UFLP represents the foundation on which all other facility location problem are based. The first formulation was developed by Balinski (1965). The problem is given a set of facility locations and set of customers who are served from the facilities. The objective is to find the set of facilities to meet all of the demand and minimize cost. The fixed cost of opening a facility, manufacturing costs, unit production cost and shipment costs per unit was concerned in the formula. For UFLP problem is “integer-friendly”. The problem can be solved using linear programming with little or no resort to branch and bound. Moris (1978) found that 96% of test problems he solved using linear programming alone resulted in all-integer solutions. Krarup and Pruzan (1983) describe this to the “single-assignment” feature of the problem. This means that, because facilities have no capacity constraints, there is an optimal solution in which each demand is supplied by exactly one facility. Because of Demand is not split. UFLP can be solved using standard mixed-integer programming (or often just linear programming). The example of uncapcitated facility location is as Figure below.

$$\begin{aligned}
\text{Min} \quad & \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J}(t_{ij}+e_i)d_j x_{ij} \\
\text{s.t.} \quad & \sum_{i\in I} x_{ij} \geq 1 \quad \forall j \in J, \\
& x_{ij} \leqslant y_i \quad \forall i \in I,\ j \in J, \\
& x_{ij} \geqslant 0 \quad \forall i \in I,\ j \in J, \\
& y_i \in \{0,1\} \quad \forall i \in I,
\end{aligned}$$

Figure 5 UFLP formulation

The capacitated facility location problem (CFLP) is a well-known combinatorial optimization problem. It consists in deciding which facilities to open from a given set of potential facility locations and how to assign customers to those facilities. The objective is to minimize total fixed and shipping costs. Constraints are that each customer’s demand must be satisfied and that each plant cannot supply more than its capacity if it is open. The CFLP partially corrects for one the major drawbacks of the UFLP. As the name suggests, it substitutes a limited (or capacitated) supply from each potential facility for the unlimited supply available in the UFLP. In some versions of the CFLP, only upper limits are placed on the output of a facility. In other versions, both upper and lower limits are used. The CFLP is much more difficult to solve than the UFLP and much less integer-friendly. Although the issue has never been investigated in depth, the results of Heller (1985) and Heller *et al.* (1989) suggest that the extent to which the CFLP can be solved without much resort to branch-and-bound is dependent on the tightness of the capacity constraints.

$$
\begin{aligned}
\text{Min} \quad & \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{i \in I} x_{ij} \geqslant 1 \quad \forall j \in J, \\
& x_{ij} \leqslant y_i \quad \forall i \in I,\ j \in J, \\
& \sum_{j \in J} d_j x_{ij} \leqslant s_i y_i \quad \forall i \in I, \\
& x_{ij} \geqslant 0 \quad \forall i \in I,\ j \in J, \\
& y_i \in \{0, 1\} \quad \forall i \in I,
\end{aligned}
$$

Figure 6 CFLP formulation

$\boldsymbol{f_i}$ is the fixed or installation cost of facility ***i***
$\boldsymbol{e_i}$ is the cost of manufacturing one unit of product at location ***i***
$\boldsymbol{x_{ij}}$ is a decision variable ranging from zero to one denoting the fraction of the demand at ***j*** to be supplied with production at location ***i***
$\boldsymbol{t_{ij}}$ is the cost of shipping one unit of product from location ***I*** to demand point ***j***
$\boldsymbol{y_i}$ is an a decision variable equal to one if the candidate site is selected for installing a facility, zero otherwise
I and ***i*** are the set and index of candidate facility locations
J and ***j*** are the set and index of demand locations
$\boldsymbol{s_i}$ is the upper bound on output of a facility opened at i
$\boldsymbol{c_{ij}}$ has been substituted for $(\boldsymbol{t_{ij}}+\boldsymbol{e_i})\boldsymbol{d_j}$

AJAX Technology

AJAX (Asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client-side to create interactive web application. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. AJAX is not a new programming language, but a new technique to creating better, faster, and more interactive web application. With AJAX, a JavaScript can communicate directly with the server, with the XMLHttpRequest object. With this object, a JavaScript can trade data with a web server, without reloading the page. AJAX uses asynchronous data transfer (HTTP requests) between the browser and the web server, allowing web page to request small bits of information from the server instead of the whole pages.

Techniques for the asynchronous loading of content data back to the mid 1990s. Java applets were introduced in the first version of the java language in 1995. These allow compiled client-side code to load data asynchronously from the web server after a web page is loaded. In 1996, Internet Explorer introduced the IFrame element to HTML, which also enables this to be achieved. In 1999, Microsoft created the XMLHTTP ActiveX control in Internet Explorer 5, which is now supported by Mozilla, Safari and other browsers as the native XMLHttpRequest object. However, this feature only became widely known after being used by Gmail (2004) and Google Maps (2005).

The term Ajax has come to represent a broad group of web technologies that can be used to implement a web application that communicates with a server in the background, without interfering with current sate of the page. The following technologies are reguired. XHTML and CSS are for presentation layer. The Document Object Model (DOM) is for dynamic display of and interaction with data. XML and XSLT are for the interchange, and manipulation and display of data. The XMLHttpRequest object is for asynchronous communication and the important technology is JavaScript to bring these technologies together.

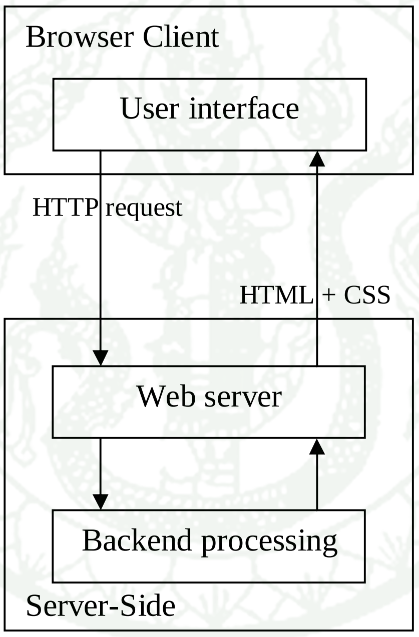


Figure 7 Classic web application model

The above Figure shows how classic web application model operates. The Brower sends the HTTP request to Web server. Then, it operates backend processing such as inserting a new record of data or calculating data, etc. After that, it sends HTML and CSS back to Browser. Therefore, the Browser page must reload every times that client sends information to the server for processing.

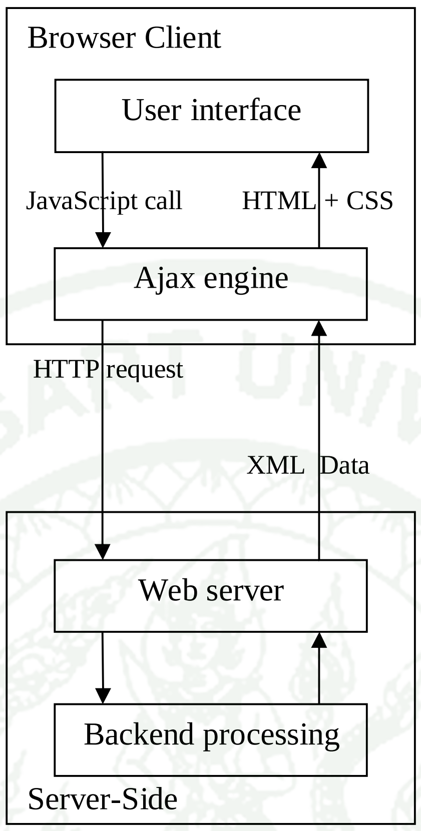


Figure 8 Ajax web application model

The above Figure 8 shows how Ajax requests information from web server. It uses JavaScript to call information by sending HTTP request to web server. For JavaScript, it is client-side script so it is able to send data to web server and change display in webpage without reloading. After processing data, web server will send XML data back to Browser client. Browser client will receive XML data and process information. Then, use JavaScript to change display in webpage.

More detail in Figure below, instead of loading a webpage, at the start of the session, the browser loads an Ajax engine written in JavaScript and usually tucked away in a hidden frame. This engine is responsible for both rendering the interface the user sees and communicating with the server on the user's behalf. The Ajax engine allows the user's interaction with the application to happen asynchronously independent of communication with the server. So the user is never staring at a blank browser window and an hourglass icon, waiting around for the server to do something.

Classic web application model (synchronous)

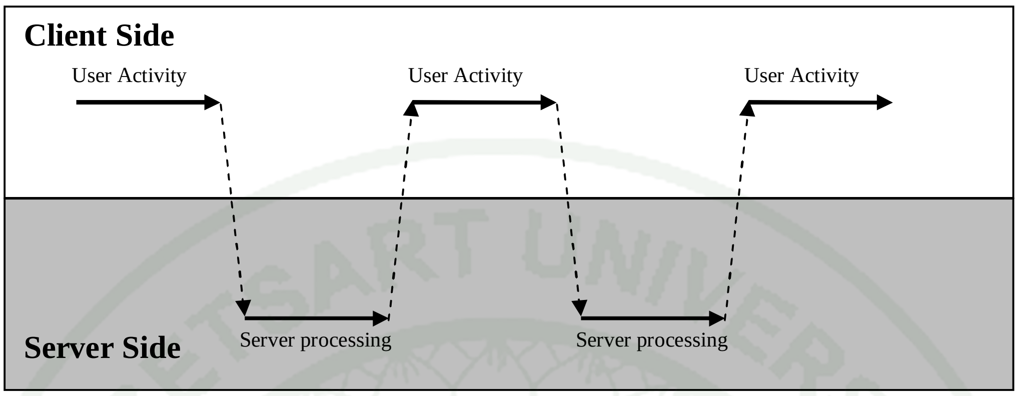


Figure 9 Classic web application model (synchronous)

Ajax web application model (asynchronous)

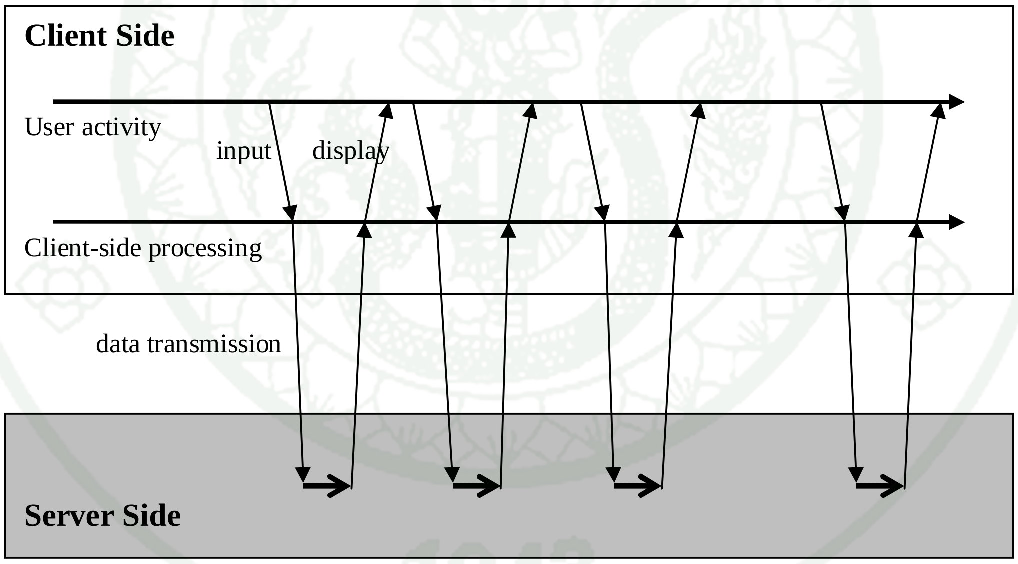


Figure 10 Ajax web application model (asynchronous)

Google Map & Google Map API

Google Maps is a basic web mapping service application and technology provided by Google, free for non commercial use that powers many map-based services including the Google Maps website, Google Ride Finder, Google Transit and maps embedded on the third party websites via the Google Maps API. It offers street maps, a route planner for traveling by foot, car, or public transport and urban business locator for numerous countries around the word. According to on of its creator, Google Maps is “a way of organizing the world’s information geographically”.

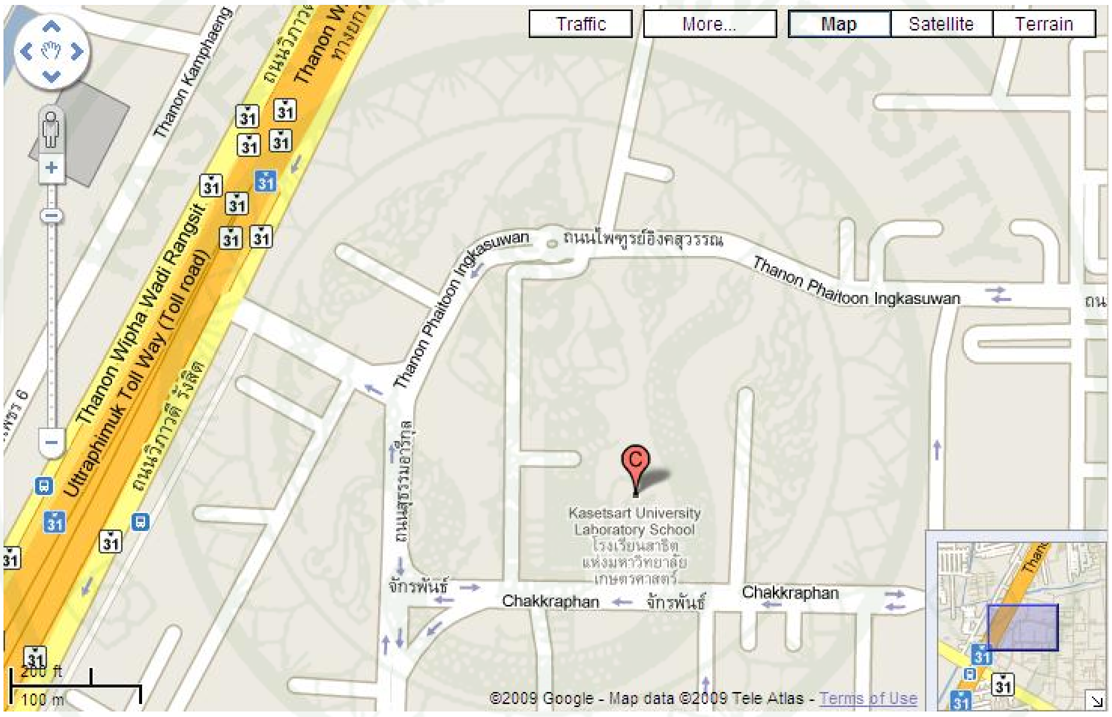

Traffic
More...
Map
Satellite
Terrain
Thanon Kamphaeng
Thanon Wipha Wadi Rangsit
Uttraphimuk Toll Way (Toll road)
Thanon Phaitoon Ingkasuwan
Thanon Phaitoon Ingkasuwan
Kasetsart University Laboratory School
Chakkraphan
Chakkraphan
200 ft
100 m
©2009 Google - Map data ©2009 Tele Atlas - Terms of Use

Figure 11 Google Map Example

Like many other Google web applications, Google Maps uses JavaScript extensively. As the user drags the map, the grid squares are downloaded from the server and inserted into the page. When a user searches for a business, the results are downloaded in the background for insertion into the side panel and map; the page is not reloaded. Locations are drawn dynamically by positioning a pin on top of the map images. As Google Maps is coded almost entirely in JavaScript and XML, some end users have reverse-engineered the tool and produced client-side scripts and server-side hooks which allowed a user or website to introduce expanded or customized features into the Google Maps interface.

Using the core engine and the map/satellite images hosted by Google, such tools can introduce custom location icons, location coordinates and metadata, and even custom map image sources into the Google Maps interface. Google created the Google

Maps API to allow developers to integrate Google Maps into their websites with their own data points. It is a free service, and currently does not contain ads, but Google states in their terms of use that they reserve the right to display ads in the future.

By using the Google Maps API, it is possible to embed the full Google Maps site into an external website. Developers are required to request an API key, which is bound to the website and directory entered when creating the key. The Google Maps API key is no longer required for API version 3. Creating a customized map interface requires adding the Google JavaScript code to a page, and then using JavaScript functions to add points to the map. The Google Maps API lets you embed Google Maps in your own web pages with JavaScript. The API provides a number of utilities for manipulating map (just like on the maps.google.com web page) and adding content to the map through a variety of services, allowing you to create robust maps applications on website.

AMPL and CPLEX programming

A Modeling Language for Mathematical Programming, AMPL, is a comprehensive, powerful, algebraic modeling language for problem in linear, nonlinear, and integer programming. AMPL is based upon modern modeling principles and utilizes and advanced architecture providing flexibility most other modeling systems lack.

AMPL's solver interface supports linear, nonlinear, and mixed integer models with no built-in size limitations. This interface is rich enough to support many of the features used by advanced solvers to improve performance and solution accuracy, such as piecewise-linear constructs, representation of network problem, and automatic differentiation of nonlinear functions. You can choose a solver for a particular problem by giving its executable filename in the AMPL option solver. This research uses CPLEX solver.

CPLEX is designed to solve linear programs, as well as the integer program. Integer programs may be pure (all integer variables) or mixed (some integer and some continuous variables); integer variables may be binary (taking values 0 and 1 only) or may have more general lower and upper bounds.

For problems that contain integer variables, CPLEX uses a "branch & cut" approach. The optimizing algorithm maintain a hierarchy of related linear programming sub-problem, referred to as the search tree and usually visualized as branching downward:

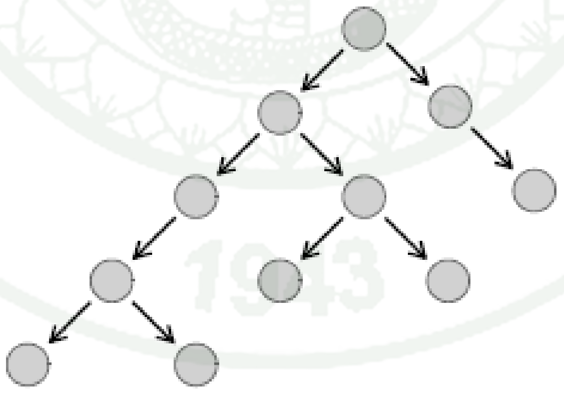


Figure 12 CPLEX Mixed Integer Algorithm: The Search Tree

Branch and cut (sometimes written as branch-and-cut) is a method of combinatorial optimization for solving integer linear programs, that is, linear programming problems where some or all the unknowns are restricted to integer values. The method is a hybrid of branch and bound and cutting plane methods.

The method solves the linear program without the integer constraint using the regular simplex algorithm. When an optimal solution is obtained, and this solution has a non-integer value for a variable that is supposed to integer, a cutting plane algorithm is used to find further linear constraints which are satisfied by all feasible integer points but violated by the current fractional solution. If such an inequality is found, it is added to the linear program, such that resolving it will yield a different solution which is hopefully "less fractional". This process is repeated until either an integer solution is found (which is then known to be optimal) or until no more cutting planes are found.

At this point, the branch and bound part of the algorithm is started. The problem is split into two versions, one with the additional constraint that the variable is greater than or equal to the next integer greater than the intermediate result, and one where this variable is less than or equal to the next lesser integer. In this way new variables are introduced in the basis according to the number of basic variables that are non-integers in the intermediate solution but which are integers according to the original constraints. The new linear programs are then solved using the simplex method and the process repeats until a solution satisfying all the integer constraints is found. During the branch and bound process, further cutting planes can be separated, which may be either global cuts, i.e., valid for all feasible integer solutions, or local cuts, meaning that they are satisfied by all solutions fulfilling the side constraints from the currently considered branch and bound sub-tree.

AMPL has Student Edition and Professional Edition. The AMPL Student Edition consists of the AMPL software together with the MINOS and CPLEX solvers. It is limited to 300 variables and 300 constraints and objectives, and does not support user-defined functions, but is otherwise identical to the full professional edition. The professional edition is the full-featured, unrestricted version. Numbers of variables and constraints are limited only by available computer resources.

NEOS Server Solver

NEOS allows us to choose a solver for the type of optimization problem that researcher would like to solve online. Developers and researchers are able to submit job to NEOS Solvers. The job will process in NEOS Server for a while. Finally, the result can be requested and send back to our application using XML-RPC protocol. NEOS runs as XML-RPC server to communicate with client for submitting and retrieving jobs. The server can communicate with clients writing in Python, Perl, PHP, C, C++, Java, Ruby, .NET and probably other languages as well.

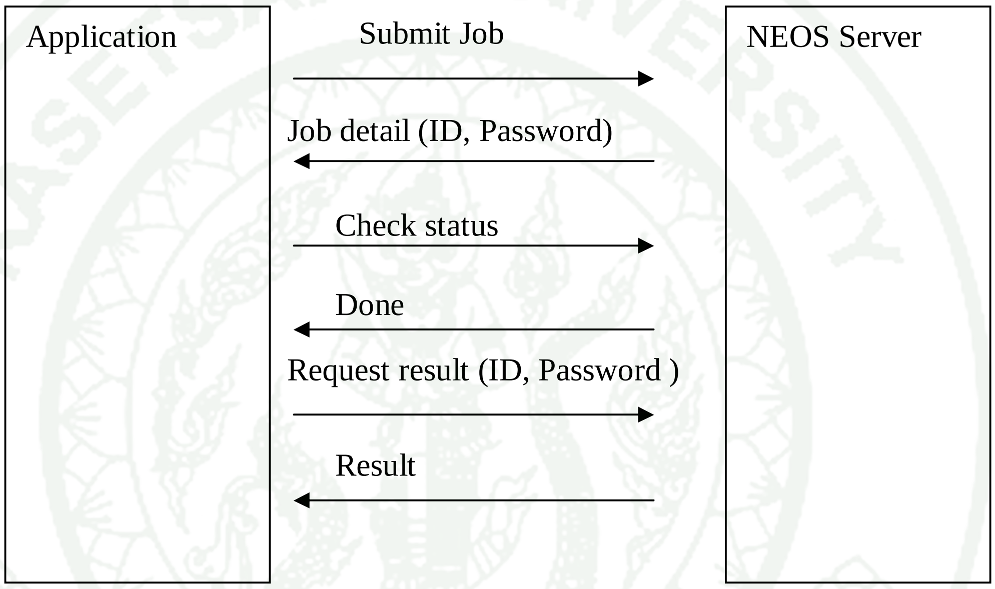


Figure 13 Application Flow for NEOS Server

The application must submit the query in AMPL format using NEOS API via XML-RPC protocol. Actually, the format depends on the solvers. The MINTO solver is selected to solve the mixed integer problem and it requires only AMPL input. After submitting the request, the NEOS Server will send back the job information which is job id and password. The application must check the job status if it finishes. The 'Done' status will send back if the job result completes successfully. Then, the application can request result and identify result using the job id and password.

MATERIALS AND METHODS

High Level Design

FLP Web application (C#.net) runs on the Internet Information Server (IIS) as web server. The Google API has been used to get and display map information. Certain, variable information and result are contained in database with four tables which are Customer, Factory, Distance and Project tables. The web application accesses those data using Data Access Layer component which is able to switch to any database such as SQL Server, Oracle, etc.

In the FLP Web application, distance can be calculated from many ways such as Point to Point or driving distance. The Point to Point method just calculates distance directly from first point to second point. For driving distance, the Google Map provides function to get distance. Google Map also provides information about road for example: one-way or two-way road. It uses this kind of information to calculate routing. The more information Google Map provided, the more accurate distance calculated from Google API.

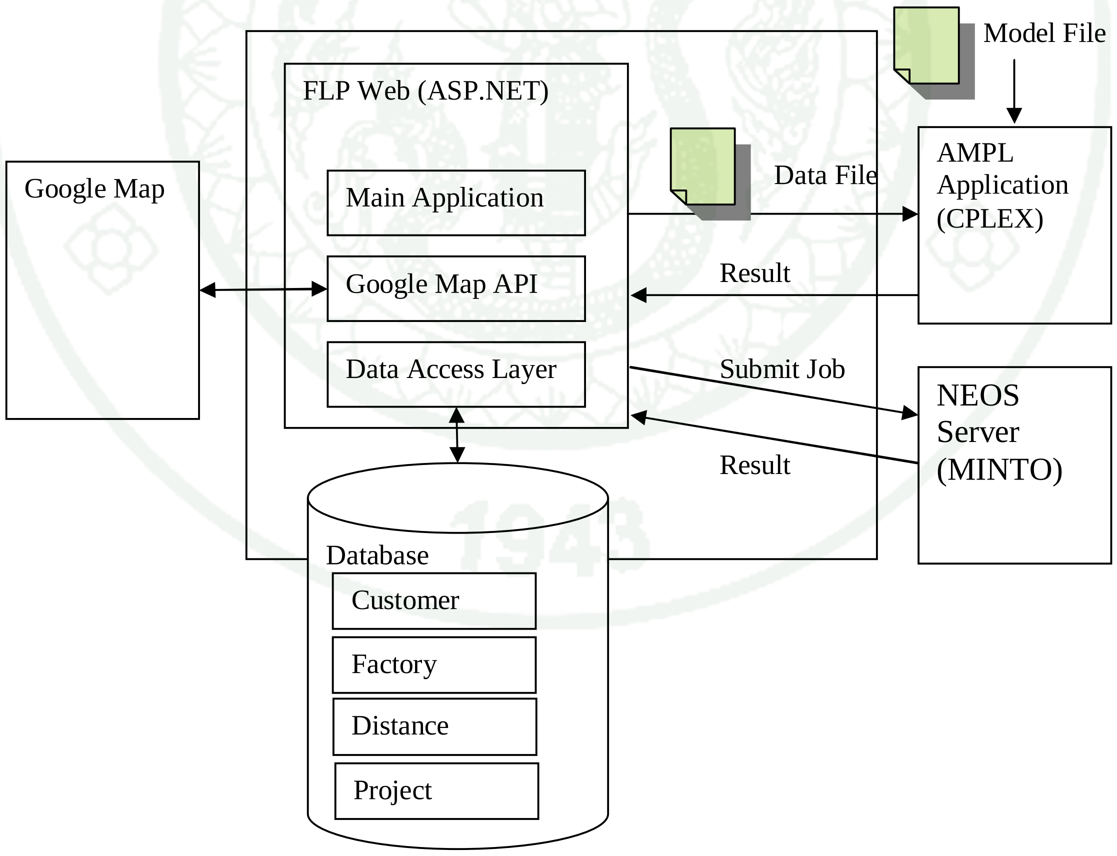


Figure 14 High-level Design

After collecting data and ready for Facility Location problem model, then the program will send data file to AMPL for finding the optimal solution against model file that has been prepared in the web application. This application provides two methods to solve problem. First, the local AMPL application with CPLEX has been used. Second, the online solver called NEOS solvers is implemented with MINTO solver. The AMPL both online and offline will solve the problem and return the result back in text. Finally, the web application will extract the result from that text and keep the result in database.

Facility Location Problem Model

The Capacitated Facility Location problem (CFLP) has been selected for this research. Transportation cost has been added in the formula. The proposed solution is implemented the application with simple formula for more understanding and for further development. The total cost of the formula is as below:

```
minimize Total_Cost: (sum {i in I} f[i] * y[i]) + (sum{i in I, j in J} (t[i,j] + e[i]) *
d[j] * x[i,j]);
```

The objective is to minimize total cost above. The Total cost consists of summary of Installation Cost of selected facilities, cost of shipping cost and product cost. See more detail in AMPL's script model as follows:

```
set I;
set J;

param f {i in I};
param t {i in I,j in J};
param e {i in I};
param d {j in J};

param s {i in I};

var y {i in I} integer >=0, <=1;
var x {i in I,j in J};

minimize Total_Cost: (sum {i in I} f[i] * y[i]) + (sum{i in I, j in J} (t[i,j] + e[i]) *
d[j] * x[i,j]);

subject to Con1 {j in J}: sum {i in I} x[i,j] >= 1;
subject to Con2 {i in I,j in J}: x[i,j] <= y[i];
subject to Con3 {i in I,j in J}: x[i,j] >= 0;
subject to Con4 {i in I}: sum {j in J} (d[j] * x[i,j]) <= (s[i] * y[i]);
```

I and **i** are the set and index of candidate facility locations
J and **j** are the set and index of demand locations
f[i] is the fixed or installation cost of facility **i**
e[i] is the cost of manufacturing one unit of product at location **i**

x[i,j] is a decision variable ranging from zero to one denoting the fraction of the demand at **J** to be supplied with production at location **i**
t[i,j] is the cost of shipping one unit of product from location **i** to demand point **j** (In this research, it will be generated from distance multiplied by transportation cost per unit)
y[i] is an a decision variable equal to one if the candidate site is selected for installing a facility, zero otherwise
d[j] is customer demand at location **j**
s[i] is the upper bound on output of a facility opened at **i**

The formulation must have the four constraints:

Con1,Con3 : is decision variable ranging from zero to one.
Con2 : ensure that **x[i,j]** should be zero if that facility **i** is not selected.
Con4 : ensure that all products are supplied by facility must not be greater than the upper bound of a facility opened at **i**

Technology Selection

The technology has been selected in this research with the following criteria:

1) It is not too expensive
2) Easy to understand, development and maintenance

Table 2 Technology Selection

Technology	Description
Google Map	Map source
ASP.NET (C#)	Development Language
Microsoft Visual Studio	Development Toolkit
MS SQL Server	Database Management System
AMPL	A Mathematical Programming Language
ILOG CPLEX, MINTO	Integer programming solver.
NEOS Server	Online Solver

The development tool and C#.net in Visual Studio have been selected due to it is quite easy to development and maintenance. The developer is able to download the Microsoft Visual Studio Express edition for free. For database, the proposed application provides the data access layer so the database is able to be any database management system such as MS SQL Server, Oracle, DB2 or MS Access, etc. However, the application has been tested in MS SQL Server Express Edition which is free of charge. To get the longitude and latitude, the Google Map must be used. Actually, there are many map internet sources but Google Map has been selected because it is well-known map source and provides us API (Application Programming

Interface). Therefore, the researcher can be implemented our own problem connecting to Google Map easily. Finally, there are two types of solver “Local” using local AMPL application and “Online” using NEOS Server. This research selects AMPL with CPLEX, MINTO as integer problem solver to solve the problem.

Materials

The following hardware and software are what the researcher uses for developing and testing the proposed application.

1. Hardware

1.1 Intel Pentium M processor 1.73GHz
1.2 10 GB free disk space.
1.3 1 GB RAM
1.4 10/100 Ethernet LAN Connection

2. Software

2.1. Windows XP Professional
2.2. Internal Information Services 6.0 or later (IIS 6.0 or later)
2.3. Visual Studio 2005 SP1 (C#.net) with Javascript
2.4. A Mathematical Programming Language (AMPL)
2.5. ILOG CPLEX system 10.0
2.6. Microsoft SQL Server 2005
2.7. Internet Explorer 7 or 8

Database Design

First, the Entity-Relationship should be created. With the Facility Location problem, there are Customer and Factory objects. The relation of them is "distance" between them. Additional, this application should be able to save project so Project object must also be in ER diagram. See more detail in ER diagram below:

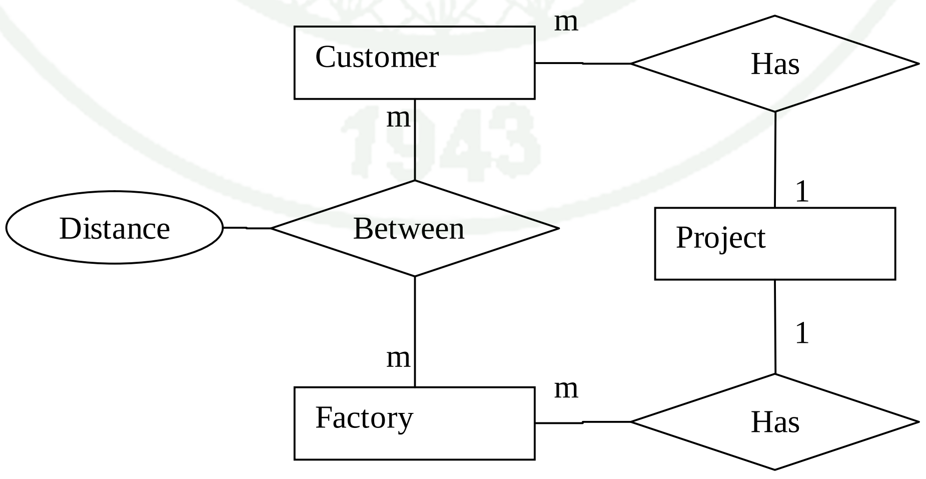


Figure 15 Entity-Relationship Diagram

After ER diagram created, it must be normalized by using database normalization. The normalization result is as below:

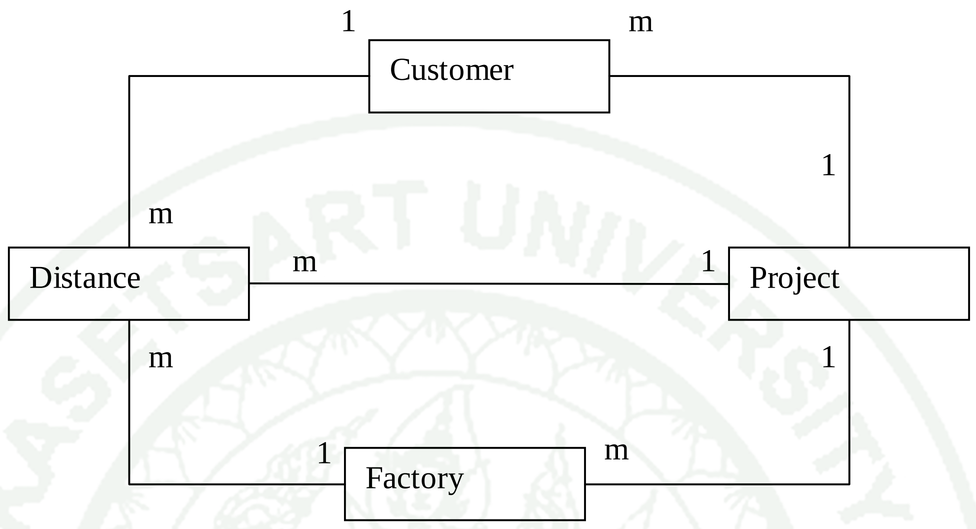


Figure 16 Entity-Relationship Diagram with normalization

The Figure above has been normalized. The researcher adds the relation between Distance and Project to implement the software easier. The attributes and foreign keys in each object are as follows:

Factory

<u>fac_id</u>, fac_name, fac_longitude, fac_latitude, fac_fixedcost, fac_unitcost, fac_transportcost, fac_capacity, fac_open, *project_id*

Customer

<u>cust_id</u>, cust_name, cust_longitude, cust_latitude, cust_demand, *project_id*

Distance

<u>dist_id</u>, distance, demand_percent, *project_id, fac_id, cust_id*

Project

<u>project_id</u>, project_name, project_description, project_totalcost, project_rawresult, project_neos_id, project_neos_password, project_neos_status

Figure 17 Attributes and foreign key for each object for database design

Each object represents each table that will be created. Therefore, the Factory, Customer, Distance and Project will be created. Every attributes in each object will be created as columns. See the final result and more detail in Appendix section.

Distance Calculation

Facility Location problem in this research needs distance between each facility and customer. In the proposed solution, there are distance information from Google Map and point-to-point calculation. From Google Map, it provides information called driving distance. The driving distance is like we drive a car from each facility to each customer in real road. For point-to-point calculation, the distance is calculated from one point to another point by using mathematic using spherical law of cosines theory.

The FLP Web application gets distance from Google Map for driving distance and stored- procedure for point-to-point calculation. After that, the distance information will be archived in Distance table.

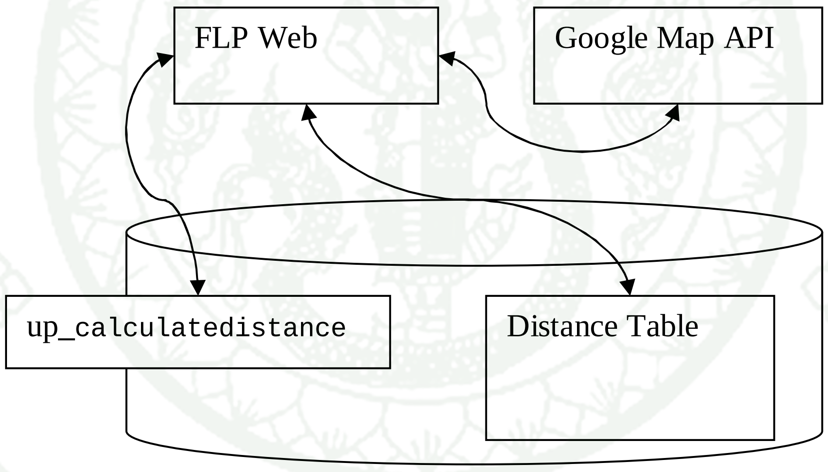


Figure 18 Calculate Distance

Using Google Maps or other online sites to get the distance between two points will work for most everyday situations, but if you have to calculate the distances of multiple (sometimes thousands) of distances programmatically there is an easy solution by using "Spherical law of cosines" theory and calculating them in SQL Stored-procedure.

Spherical law of cosines:

```
ACOS(SIN(RADIANS(@lat1))*SIN(RADIANS(@lat2))+COS(RADIANS(@lat1))
*COS(RADIANS(@lat2))*COS(RADIANS(@lon2-@lon1)))*6371
```

The stored-procedure has been created to calculate all distances between each customer and each factory. The main application passes one parameter Project ID. Then, the stored-procedure will generate distance for all pairs of factories and customers in that project.

```
CREATE PROCEDURE [dbo].[up_calculatedistance]
      (
            @project_id int
      )
AS
BEGIN

      UPDATE DISTANCE SET distance =
ACOS(SIN(RADIANS(fac_latitude))*SIN(RADIANS(cust_latitude))+COS(RADI
ANS(fac_latitude))*COS(RADIANS(cust_latitude))*COS(RADIANS(cust_long
itude)-RADIANS(fac_longitude)))*6371
      FROM  Customer, Factory, Distance
      WHERE Customer.Cust_id= Distance.Cust_id
      AND Factory.Fac_id = Distance.Fac_id
      AND Distance.Project_id = @project_id

      RETURN
END
```

Actually, there are many methods to calculate distance between two points. However, the research uses simple methods to calculate.

Calculate Result with AMPL

The application will be complex and need too many efforts to create our own operational research problem's solver. Therefore, this research uses third party application which is AMPL instead. Facility Location Problem web application (FLP Web) sends the data file to AMPL. The AMPL uses the static model file which created for this proposed web application using CPLEX solver for integer programming problem. Then, it will send the information back to FLP Web. The result from AMPL is text so the application must be translate to be more user-friendly.

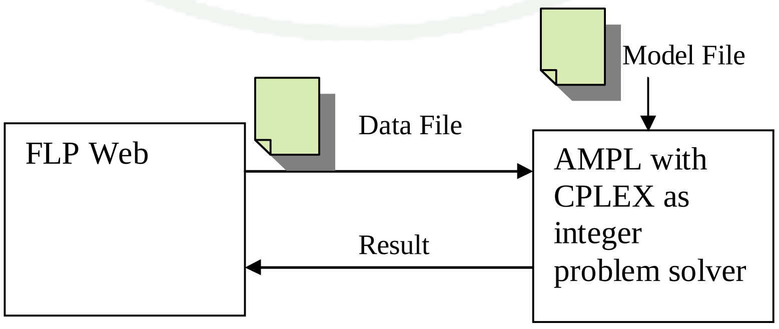
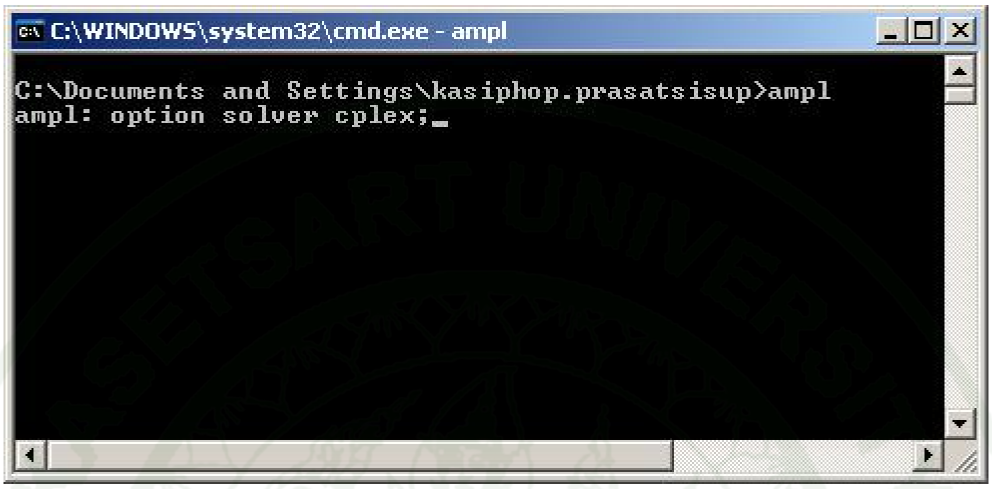


Figure 19 Get result from AMPL

The web application calls AMPL by using command prompt application as background process as shown below and run the AMPL command in that command prompt.



```
C:\Documents and Settings\kasiphop.prasatsisup>ampl
ampl: option solver cplex;_
```

```
option solver cplex;
model CPLEX.mod data CPLEX.dat solve;
display Total_Cost;
display y;
display x;
```

Figure 20 Call AMPL in command prompt and run AMPL Command

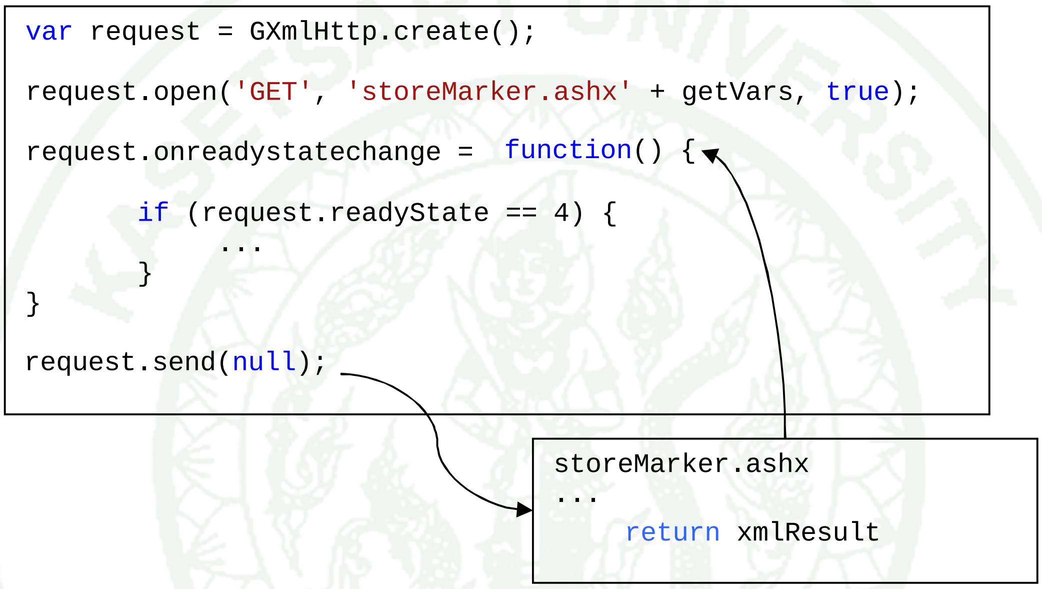
The AMPL command above runs for:

Option solver cplex : solve the problem using integer programming solver (CPLEX)
Model CPLEX.mod : use model file named CPLEX.mod.
Data CPLEX.dat : use data file named CPLEX.dat
Solve : solve the problem with the specific model and data.
Display Total_Cost : Show the result of Total Cost
Display y : Show the result of selected facilities.
Display x : Show demand supplied by facilities to customers

The result from "display" command in command prompt will be captured and sent to application.

Web Application

FLP Web Application should be implemented by using AJAX. The Figure below show how the application works with JavaScript to implement Asynchronous technology. After running `request.send(null)` command, the process jumps to run `storeMarker.ashx` which specific in `request.open()`. This process works as background. After `storeMarker.ashx` process finishes, it will get back to call function specified in `onreadystatechange`.



```
var request = GXmlHttp.create();

request.open('GET', 'storeMarker.ashx' + getVars, true);

request.onreadystatechange =  function() {

       if (request.readyState == 4) {
            ...
       }
}

request.send(null);
```

```
storeMarker.ashx
...
    return xmlResult
```

Figure 21 How JavaScript work in asynchronous model

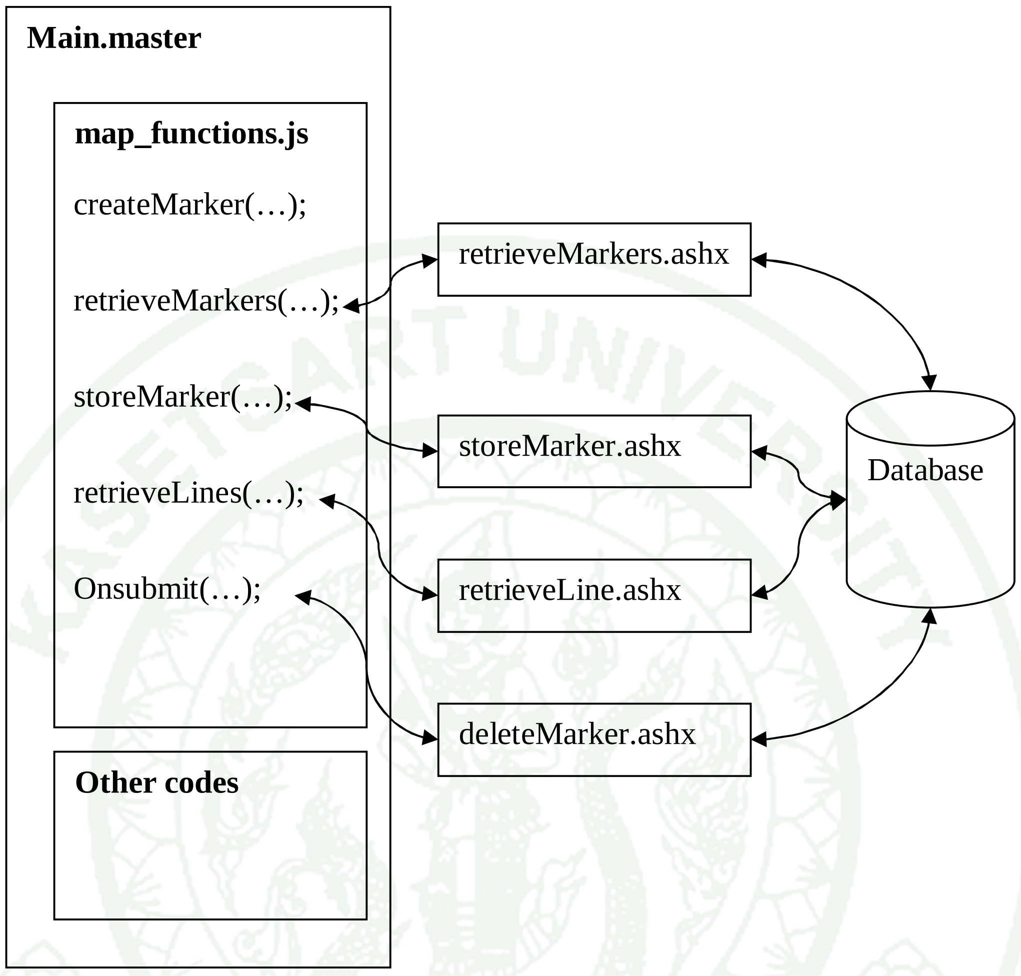


Figure 22 Function and File diagram

The application also has features to create, retrieve, store and delete markers. The Figure above shows the rough diagram which action and file are related together. For example, retrieveMarker() function will be called at the initial page. Next, it will be called storeMarker.ashx as asynchronous background process. If process completes, the result which is implemented in XML format will be sent back to its calling function.

Before generating the total cost optimal result of facility location problem, the client must fill mandatory input of facility, customer and distance information. The usage work flow shows in Figure below.

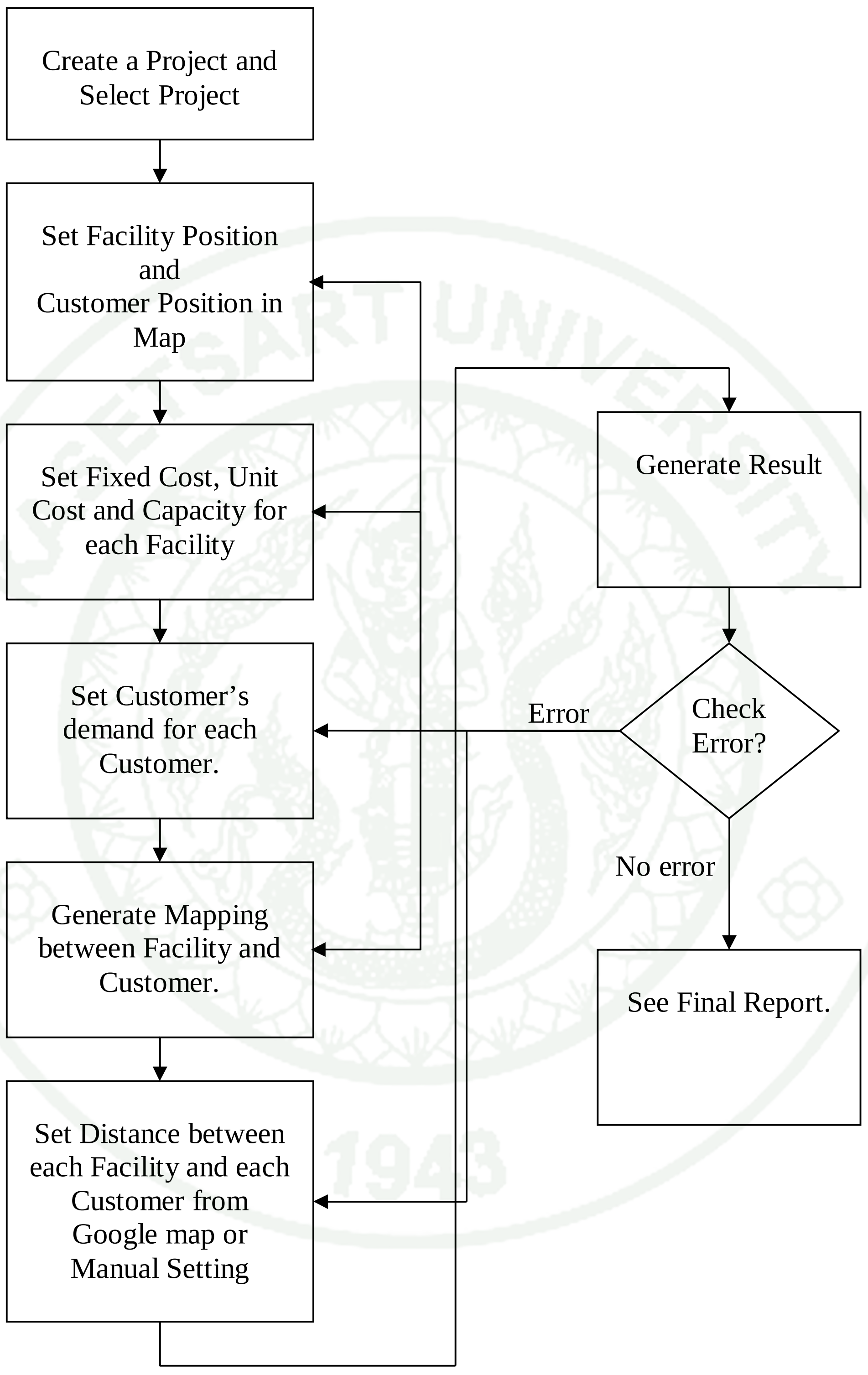
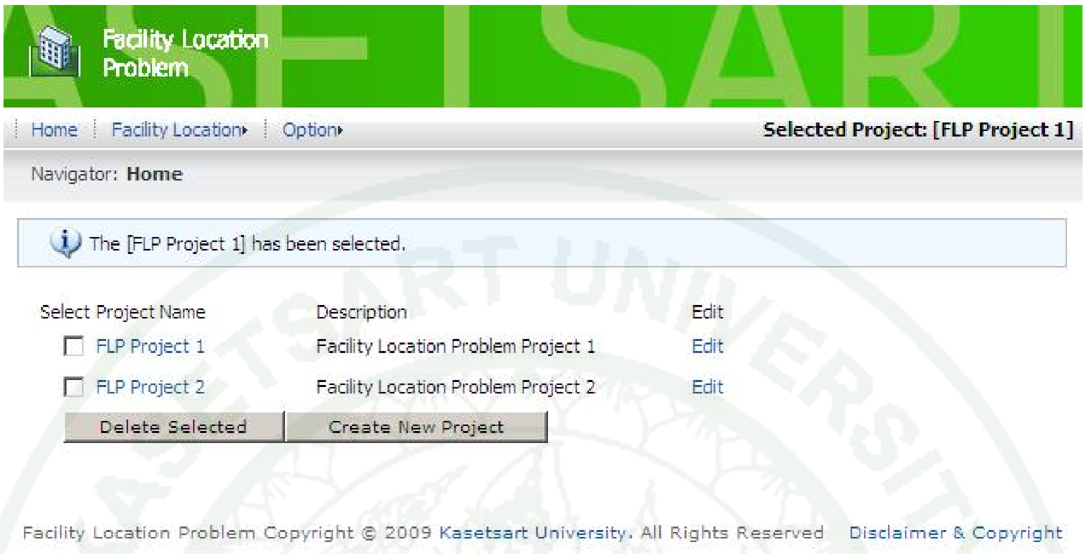

Create a Project and Select Project
Set Facility Position and Customer Position in Map
Set Fixed Cost, Unit Cost and Capacity for each Facility
Set Customer's demand for each Customer.
Generate Mapping between Facility and Customer.
Set Distance between each Facility and each Customer from Google map or Manual Setting
Generate Result
Check Error?
Error
No error
See Final Report.

Figure 23 Usage Flow Chart

The application is able to save and edit project.



Selected Project: [FLP Project 1]

The [FLP Project 1] has been selected.

Select Project Name	Description	Edit
☐ FLP Project 1	Facility Location Problem Project 1	Edit
☐ FLP Project 2	Facility Location Problem Project 2	Edit

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

Figure 24 Project List

The facility and customer must be located in the application first.

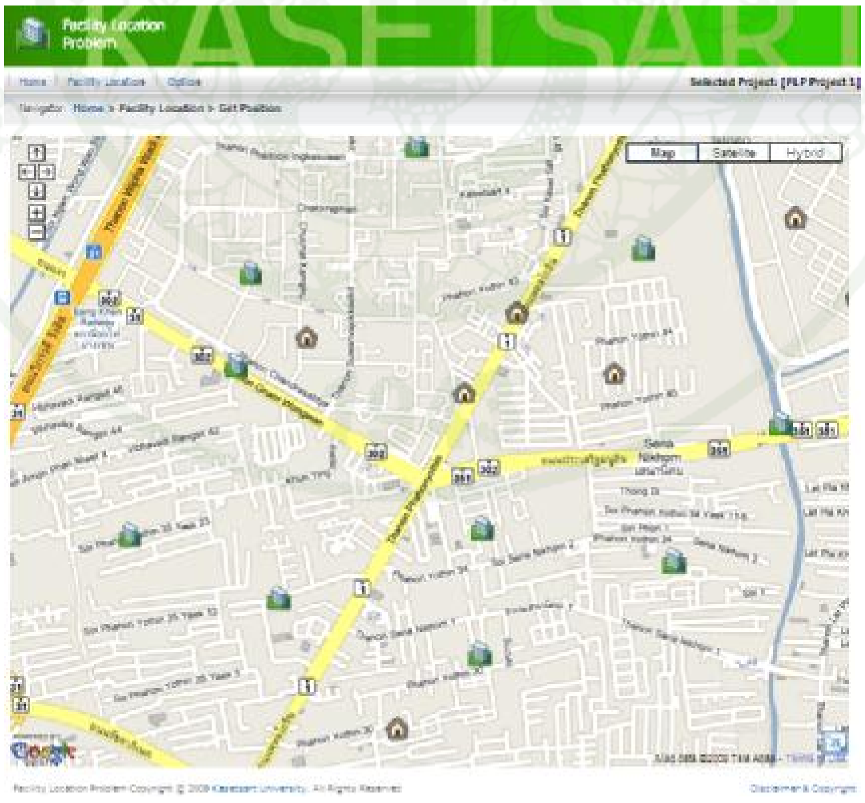
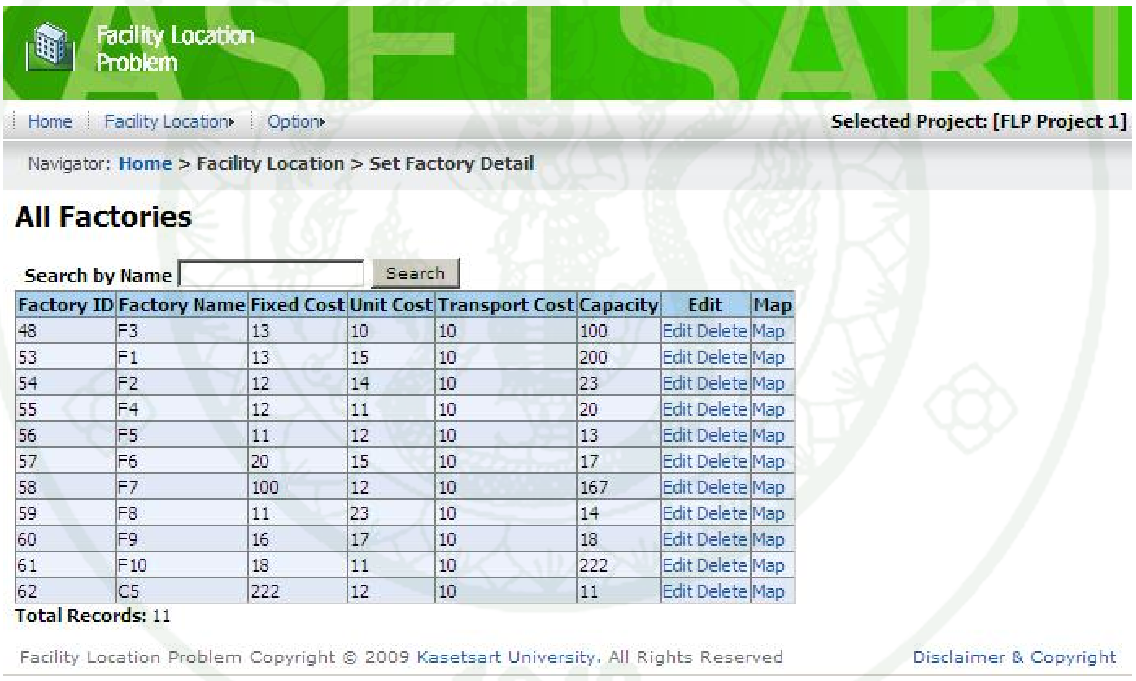

Facility Location Problem
Home
Facility Location
Option
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Get Position
Map
Satellite
Hybrid

Figure 25 Facility and customer location setting in Google Map

Then, input factory detail in the application. The unit of every input must be the same. For example, Unit of the Unit Cost, Fixed Cost and Transport Cost should be Baht.

Table 3 Facility Input Description

Facility Input	Description
Factory Name	The Factory Name
Fixed Cost	For example, Installation Cost, Factory Set-up cost, real estate cost
Unit Cost	Product cost per one unit
Transport Cost	Transportation cost per one unit per kilometer
Capacity	Product capacity that can be supplied by the facility


Facility Location Problem
Home
Facility Location
Option
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Set Factory Detail
Search by Name
Search

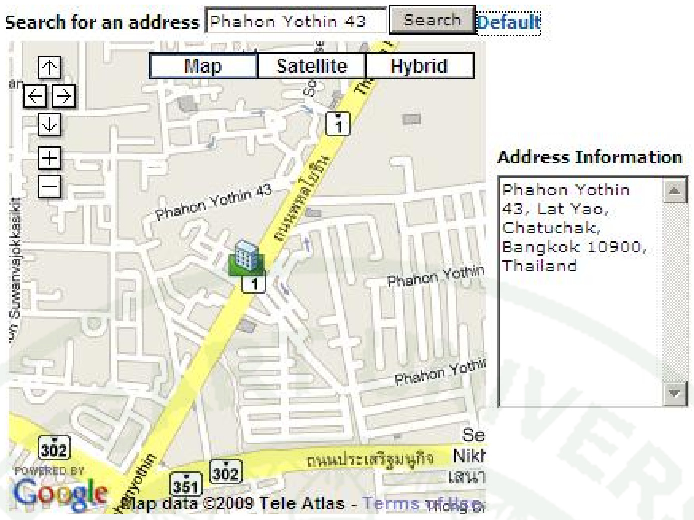
All Factories

Factory ID	Factory Name	Fixed Cost	Unit Cost	Transport Cost	Capacity	Edit	Map
48	F3	13	10	10	100	Edit Delete	Map
53	F1	13	15	10	200	Edit Delete	Map
54	F2	12	14	10	23	Edit Delete	Map
55	F4	12	11	10	20	Edit Delete	Map
56	F5	11	12	10	13	Edit Delete	Map
57	F6	20	15	10	17	Edit Delete	Map
58	F7	100	12	10	167	Edit Delete	Map
59	F8	11	23	10	14	Edit Delete	Map
60	F9	16	17	10	18	Edit Delete	Map
61	F10	18	11	10	222	Edit Delete	Map
62	C5	222	12	10	11	Edit Delete	Map

Total Records: 11

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved   Disclaimer & Copyright

Figure 26 Input Factory detail


Search for an address Phahon Yothin 43 Search Default

Address Information

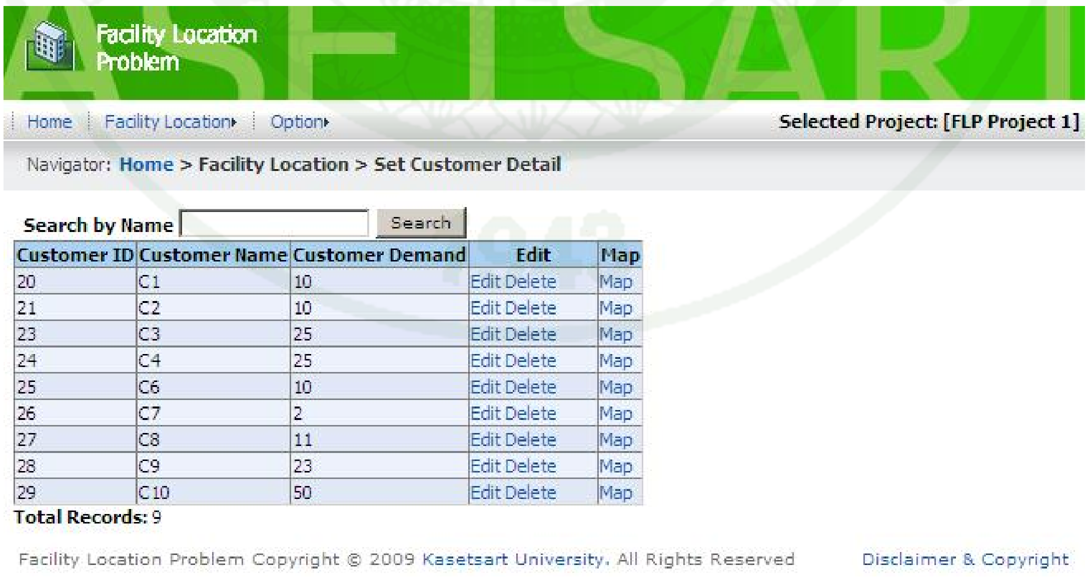
Phahon Yothin 43, Lat Yao, Chatuchak, Bangkok 10900, Thailand

Figure 27 Moving and Searching the new location

The client is able to delete the factory and customer marker. In addition, they can edit, move and search for new location.

Table 4 Customer Input Description

Customer Input	Description
Customer Name	The Customer Name
Customer Demand	Customer's product demand


Facility Location Problem

Home | Facility Location | Option | Selected Project: [FLP Project 1]

Navigator: Home > Facility Location > Set Customer Detail

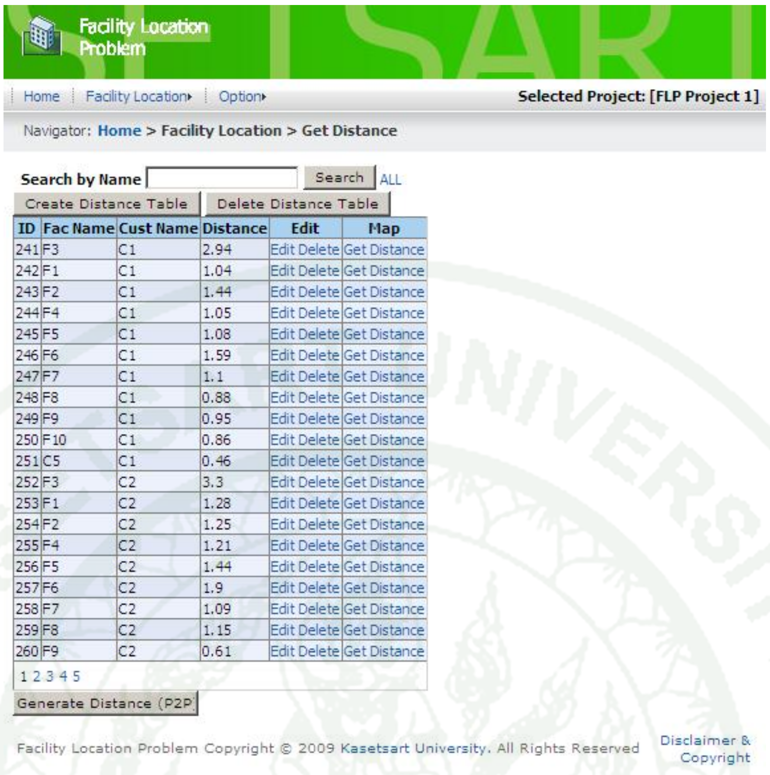
Search by Name Search

Customer ID	Customer Name	Customer Demand	Edit	Map
20	C1	10	Edit Delete	Map
21	C2	10	Edit Delete	Map
23	C3	25	Edit Delete	Map
24	C4	25	Edit Delete	Map
25	C6	10	Edit Delete	Map
26	C7	2	Edit Delete	Map
27	C8	11	Edit Delete	Map
28	C9	23	Edit Delete	Map
29	C10	50	Edit Delete	Map

Total Records: 9

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved   Disclaimer & Copyright

Figure 28 Input Customer detail



Selected Project: [FLP Project 1]

Navigator: Home > Facility Location > Get Distance

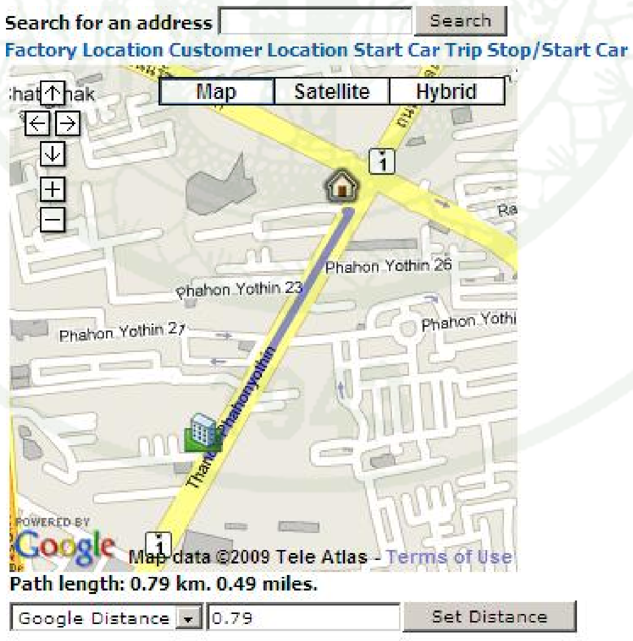
Search by Name

ID	Fac Name	Cust Name	Distance	Edit	Map
241	F3	C1	2.94	Edit Delete	Get Distance
242	F1	C1	1.04	Edit Delete	Get Distance
243	F2	C1	1.44	Edit Delete	Get Distance
244	F4	C1	1.05	Edit Delete	Get Distance
245	F5	C1	1.08	Edit Delete	Get Distance
246	F6	C1	1.59	Edit Delete	Get Distance
247	F7	C1	1.1	Edit Delete	Get Distance
248	F8	C1	0.88	Edit Delete	Get Distance
249	F9	C1	0.95	Edit Delete	Get Distance
250	F10	C1	0.86	Edit Delete	Get Distance
251	C5	C1	0.46	Edit Delete	Get Distance
252	F3	C2	3.3	Edit Delete	Get Distance
253	F1	C2	1.28	Edit Delete	Get Distance
254	F2	C2	1.25	Edit Delete	Get Distance
255	F4	C2	1.21	Edit Delete	Get Distance
256	F5	C2	1.44	Edit Delete	Get Distance
257	F6	C2	1.9	Edit Delete	Get Distance
258	F7	C2	1.09	Edit Delete	Get Distance
259	F8	C2	1.15	Edit Delete	Get Distance
260	F9	C2	0.61	Edit Delete	Get Distance

1 2 3 4 5

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved

Figure 29 Input Distance detail



Path length: 0.79 km. 0.49 miles.

Google Distance: 0.79

Figure 30 Get Distance from Google Map

The user can select distance between Car Driving distance and Point-to-Point distance. There is Car Driving feature. It simulates how user really drives car to the destination. It calculates and display car speed that drive in each road and how long the car drive so far.

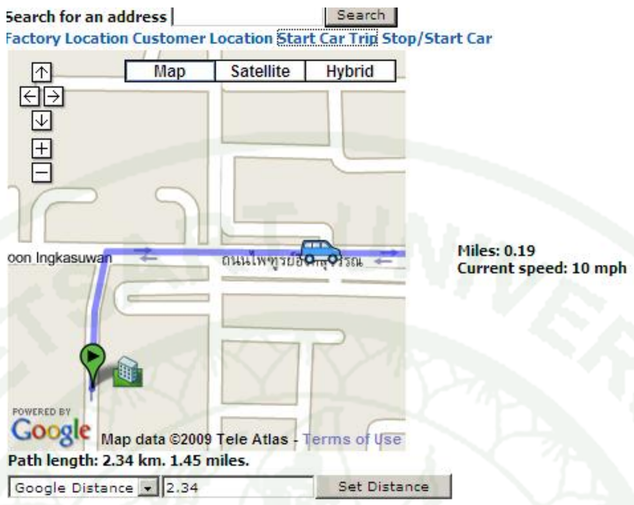
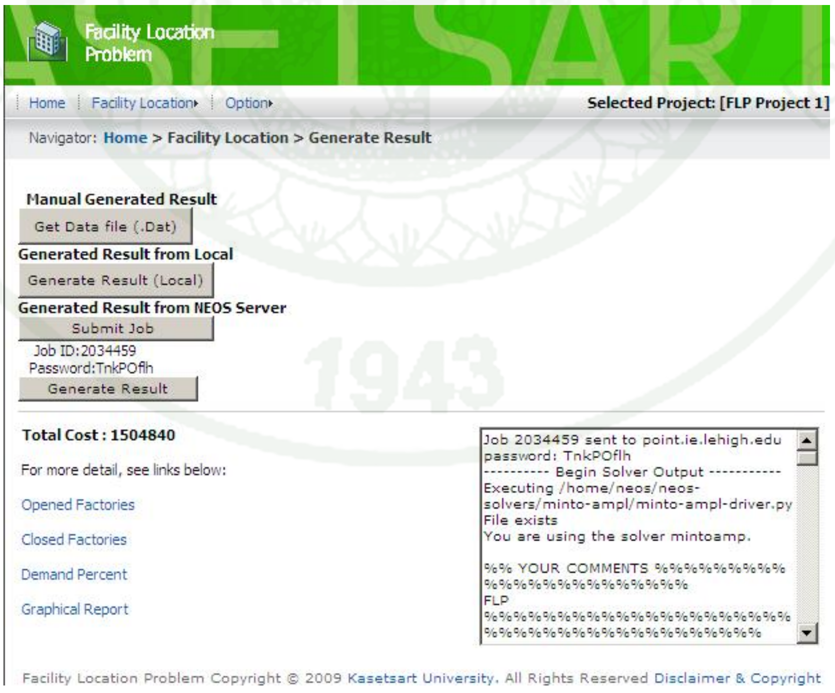

Search for an address
Search
Factory Location Customer Location Start Car Trip Stop/Start Car
Map
Satellite
Hybrid
oon Ingkasuwan
Miles: 0.19
Current speed: 10 mph
Map data ©2009 Tele Atlas - Terms of Use
Path length: 2.34 km. 1.45 miles.
Google Distance
2.34
Set Distance

Figure 31 Car Driving Distance

Finally, the user can generate optimal reports such as Opened Factories, Closed Factories, Demand Offer and Graphical Report. Moreover, user can generate raw data to solve the problem in AMPL themselves.


Facility Location Problem
Home
Facility Location
Option
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Generate Result
Manual Generated Result
Get Data file (.Dat)
Generated Result from Local
Generate Result (Local)
Generated Result from NEOS Server
Submit Job
Job ID:2034459
Password:TnkPOflh
Generate Result
Total Cost : 1504840
For more detail, see links below:
Opened Factories
Closed Factories
Demand Percent
Graphical Report
Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

```
Job 2034459 sent to point.ie.lehigh.edu
password: TnkPOflh
---------- Begin Solver Output -----------
Executing /home/neos/neos-
solvers/minto-ampl/minto-ampl-driver.py
File exists
You are using the solver mintoamp.

%% YOUR COMMENTS %%%%%%%%%%
%%%%%%%%%%%%%%%%%%
FLP
%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
```

Figure 32 Final reports

There are two ways to get the result by using Local AMPL or NEOS server. To use NEOS Server must be follows the sequence as Figure below:

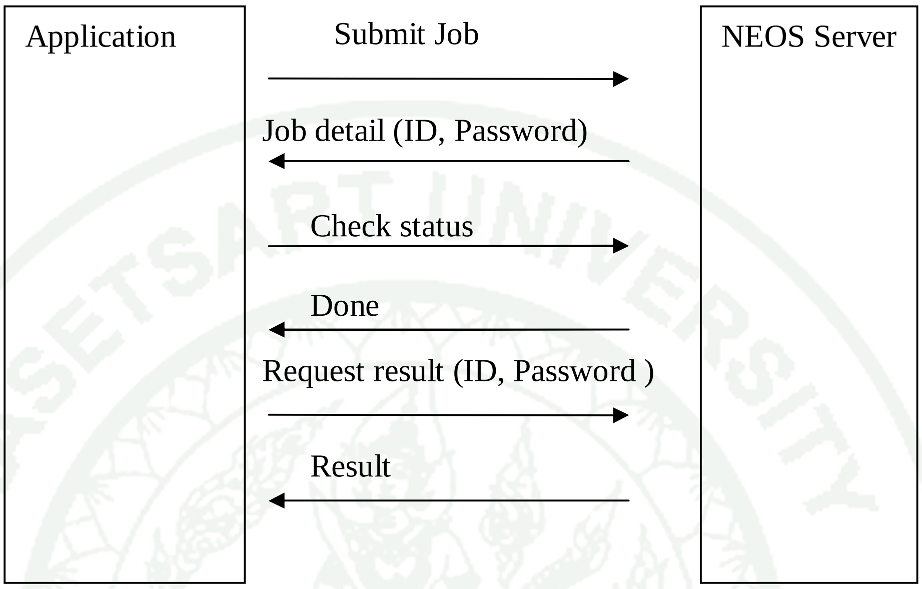


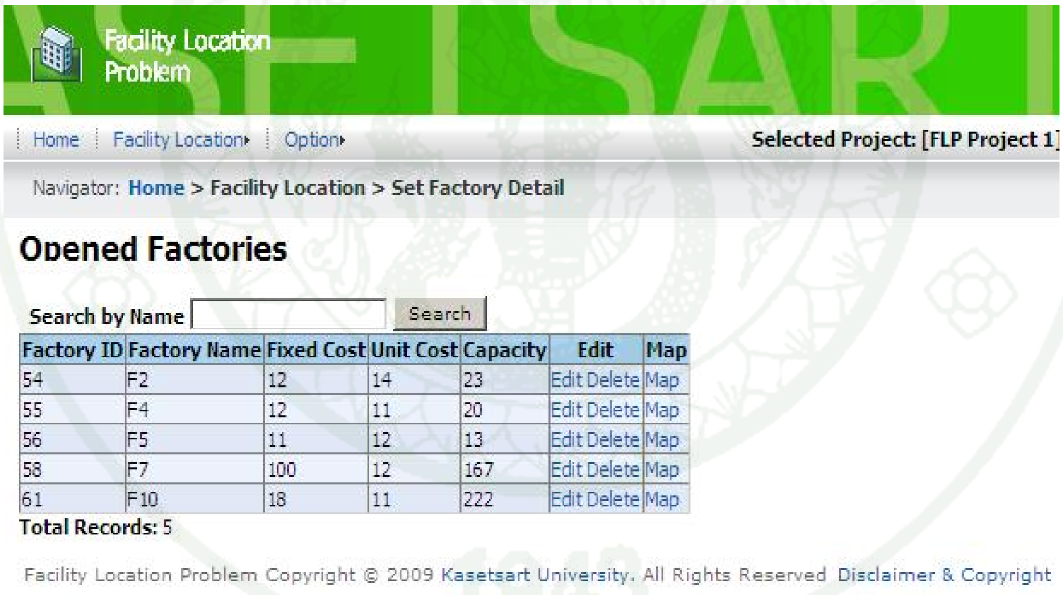
Figure 33 Submitting request to NEOS Server

RESULTS AND DISCUSSION

As the result, there are optimal total cost and four main reports. Technically, the developer can implement any dimension of report. This research implements only necessary reports.

Table 5 Result Report Description

Report Name	Description
Opened Factories	Show which facilities should be opened.
Closed Factories	Show which facilities should be opened.
Demand Percent	Show how many products that is supplied by facilities to customers.
Graphical Report	Show which facilities should be opened and closed and show relation between facilities and customers in graphical.


Facility Location Problem
Home
Facility Location
Option
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Set Factory Detail
Search by Name
Search

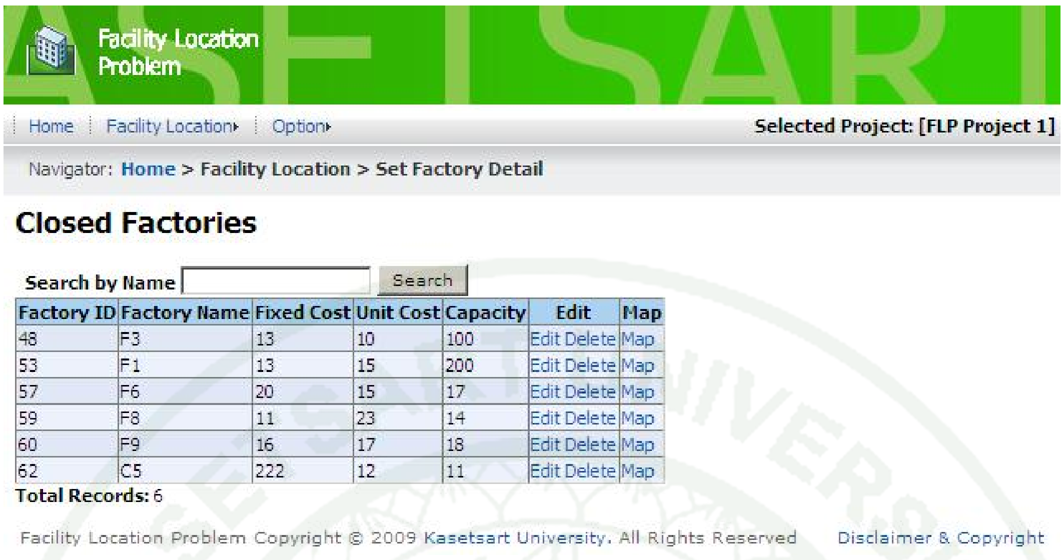
Opened Factories

Factory ID	Factory Name	Fixed Cost	Unit Cost	Capacity	Edit	Map
54	F2	12	14	23	Edit Delete	Map
55	F4	12	11	20	Edit Delete	Map
56	F5	11	12	13	Edit Delete	Map
58	F7	100	12	167	Edit Delete	Map
61	F10	18	11	222	Edit Delete	Map

Total Records: 5

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

Figure 34 Opened Factories Report



Selected Project: [FLP Project 1]

Navigator: Home > Facility Location > Set Factory Detail

Closed Factories

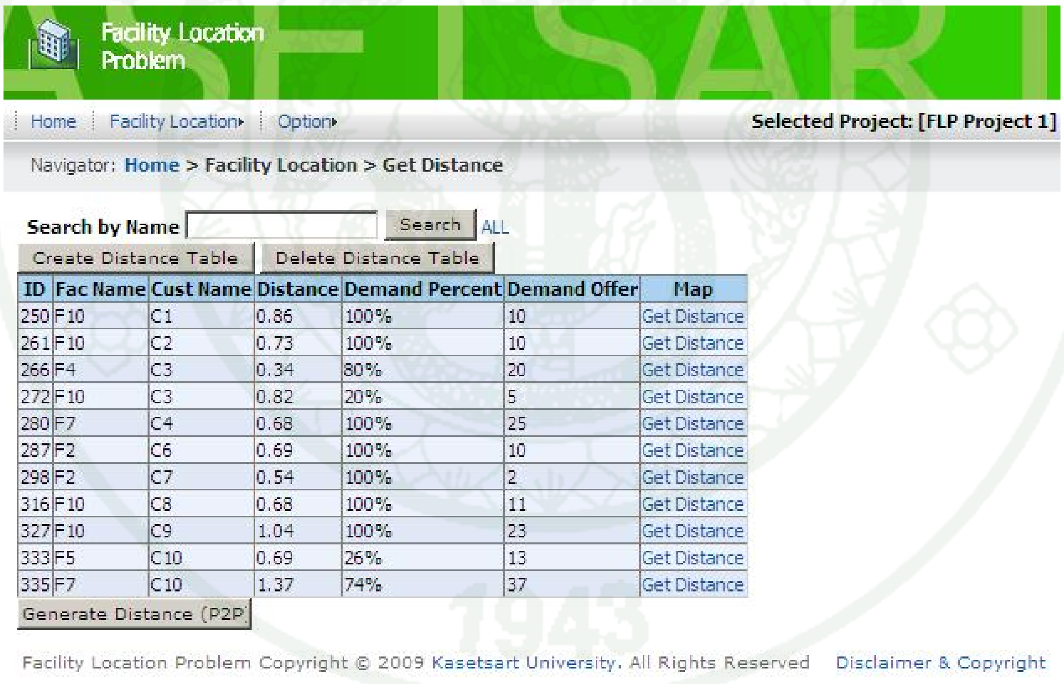
Search by Name

Factory ID	Factory Name	Fixed Cost	Unit Cost	Capacity	Edit	Map
48	F3	13	10	100	Edit Delete	Map
53	F1	13	15	200	Edit Delete	Map
57	F6	20	15	17	Edit Delete	Map
59	F8	11	23	14	Edit Delete	Map
60	F9	16	17	18	Edit Delete	Map
62	C5	222	12	11	Edit Delete	Map

Total Records: 6

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved   Disclaimer & Copyright

Figure 35 Closed Factories Report



Selected Project: [FLP Project 1]

Navigator: Home > Facility Location > Get Distance

Search by Name

ID	Fac Name	Cust Name	Distance	Demand Percent	Demand Offer	Map
250	F10	C1	0.86	100%	10	Get Distance
261	F10	C2	0.73	100%	10	Get Distance
266	F4	C3	0.34	80%	20	Get Distance
272	F10	C3	0.82	20%	5	Get Distance
280	F7	C4	0.68	100%	25	Get Distance
287	F2	C6	0.69	100%	10	Get Distance
298	F2	C7	0.54	100%	2	Get Distance
316	F10	C8	0.68	100%	11	Get Distance
327	F10	C9	1.04	100%	23	Get Distance
333	F5	C10	0.69	26%	13	Get Distance
335	F7	C10	1.37	74%	37	Get Distance

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved   Disclaimer & Copyright

Figure 36 Demand Offer Report

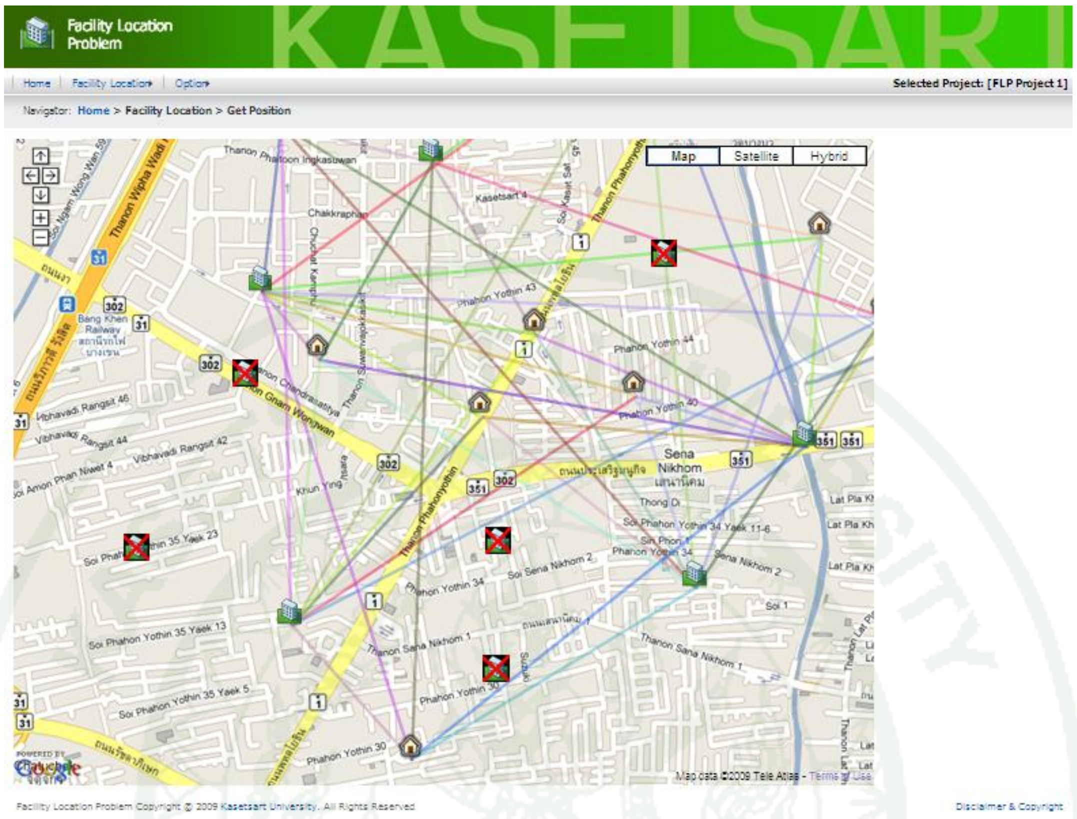

Facility Location Problem
Home | Facility Location | Option
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Get Position
Map
Satellite
Hybrid
Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved
Disclaimer & Copyright

Figure 37 Graphical Report

The application has been tested with random data to make sure that the application does work properly. The random generates 50 facilities and 50 customers. Then, the total record in distance table is 2,500 records. Additional, the researcher also tests the waiting time of NEOS Server. It does not take over than 2 minutes to get the result back from the server.

CONCLUSIONS AND RECOMMENDATIONS

There are not many visualize applications implemented for operational research problem. The research provides the proposed solution to solve a simple operational research problem called Capacitated Facility Location problem in visualize application. This proposed application provides the user friendly result that is better than reading from the text. The application has been tested with random markers to ensure that the result generates accurately.

The application can be used with real business case. For example, the Seven-Eleven or Plaza company may need to make a decision where should be new location of their business. The initial application provides the example to the user to create their own application with simple tool. Therefore, it is quite useful for business today.

With this proposed solution, there are some limitations such as limitation of geocode request, the limitation of car driving distance from Google Map, the limitation of variable number in AMPL student edition.

For limitation of geocoder, if more than 15,000 geocode requests in a 24 hour period are received from a single IP address, or geocode requests are submitted from a single IP address at too fast a rate, the Google Maps API geocoder will begin responding with a status code of 620. This article explains how to time geocode requests from Web server and provides sample code for doing so. If excessive geocoder usage continues, access to the Google Maps API geocoder from this IP address may be blocked permanently.

For limitation of car driving distance calculation, the Google Map provides road detail, one-way and u-turn location, etc. Some road details do not exist in Google Map so the calculation of car driving distance may not be accurate.
For the limitation of variable in AMPL, the AMPL student edition is able to handle only 300 variables. However, the NEOS server can be solved this problem.

Facility Location problem also needs distance information between two points so the distance table is very large if there are many facility and customer point. It consumes a lot of memory resource to calculate distance table or create AMPL data file. Therefore, the researcher suggests using this proposed solution for the small problem.

In conclusion, the proposed application in this research is able to solve the optimal solution for Capacitated Facility Location problem properly. However, there are some limitations and need more further development to improve some features to more user-friendly.

LITERATURE CITED

Balinski M, 1965. Integer programming Method**. The Management Science** 12 (3): 253-313

Dong-wan Tcha, Young-soo Myung and Ki-ho Chung, 1995. Parametric uncapacitated facility location. **European Journal of Operational Research** 86: 469-479

Efroymson, M., Ray, T., 1969. A branch-and-bound algorithm for the capacitated facilities location problem**. Naval Research Logistics Quarterly 16** (3), 331-343

Jorge Riera-Ledesma and Juan-Jose Salazar-Gonzalez, 2005. Computers & Operations Research Volume 34, Issue 9. **A branch-and-cut algorithm for the continuous error localization problem in data cleaning.** DEIOC, Universidad de La Laguna, 38271 La Laguna, Spain.

Joseph Harkness and Charles ReVelle. 2003. Facility location with increasing production costs. **European Journal of Operational Research** 145: 1-13.

Kamlesh Mathur and Daniel Solow. 1994. **Management Science, The Art of Decision Making**. PRENTICE HALL, Englewood Cliffs, New Jersey.

Soland, R. Cl, 1993. Facility location with concave costs. **Operations Research** 23 (2): 373-382

APPENDICES

Appendix A
Database Design

Appendix Table A1 Customer Table

Column Name	Data Type	Description
*Cust_id	Int [IDENTITY(1,1)]	Customer ID
Cust_name	Varchar(50)	Customer Name
Cust_longitude	Float	Longitude
Cust_latitude	Float	Latitude
Cust_demand	Int	Customer's Demand.
Project_id	Int	Foreign key of Project table

* Primary key

Appendix Table A2 Factory Table

Column Name	Data Type	Description
*Fac_id	Int [IDENTITY(1,1)]	Factory ID
Fac_name	Varchar(50)	Factory Name
Fac_longitude	Float	Longitude
Fac_latitude	Float	Latitude
Fac_fixedcost	Float	Fixed Cost, Installation Cost, etc.
Fac_unitcost	Numeric(18,0)	Cost per one unit
Fac_transportcost	Float	Transportation Cost per km
Fac_capacity	Float	Capacity
Fac_open	Smallint	Open or Close
Project_id	Int	Foreign key of Project table

* Primary key

Appendix Table A3 Distance Table

Column Name	Data Type	Description
*Dist_id	Int [IDENTITY(1,1)]	Distance ID
*Cust_id	Int	Customer ID
*Fac_id	Int	Factory ID
Distance	Float	Distance between a Facility and a Customer.
Demand_percent	Float	Record how many percent of product which is supplied by a facility to a customer.
Project_id	Int	Foreign key of Project table

* Primary key

Appendix Table A4 Project Table

Column Name	Data Type	Description
*Project_id	Int [IDENTITY(1,1)]	Project ID
Project_name	Varchar(50)	Project Name
Project_description	Varchar(200)	Project Description
Project_totalcost	Float	Optimal Total Cost
Project_rawresult	Text	Record the last output result from AMPL command line.

* Primary key

Appendix B
File and Folder Structure

Appendix Table B1 File and Folder Structure

File Name	Description
Flpweb.sln	Project File
CreateProject.aspx	Save, Create, Edit project
DeleteMarker.ashx	Call this file as background to delete selected marker
Distance_function.js	JavaScript for Pop-up dialog of distance calculation page
Epoly.js	JavaScript associated with Google Map to create poly line
ExportFile.aspx	Generate Result page
FLPConversion.cs	Global General Conversion Function for this web application
FLPFunction.cs	Global General Function for this web application
GetDistance.aspx	Get Distance page
GetPosition.aspx	Get Position Page
GoogleKey.js	Keep Google Key
HomePage.aspx	First access page
Main.Master	Header and footer template for general pages
Main2.Master	Header and footer template for error and specific pages
MainPopup.Master	Header and footer template for popup-dialog page
Map_functions.js	JavaScript for Get Position page
MessageShowControl.ascx	User control to show warning, error and information message in web page
MoveMarker.ashx	Call this file as background to move marker to new location
PointDistanceMap.aspx	Pop-up dialog to show the car driving distance from Get Distance page
PointMap.aspx	Pop-up dialog to edit the location of facility and customer from Set Facility page and Set Customer page.
Pointmap_functions.js	JavaScript of PointMap.aspx
retrieveLines.ashx	Call this file as background to show lines between each facility and each customer
retrieveMarkers.ashs	Call this file as background to get all marker locations
Search.css	Cascade style sheet
SessionCollection.cs	Keep all session variables in this class
SetCustomer.aspx	Set Customer page
SetFactory.aspx	Set Factory page
StoreMarker.ashx	Call this file as background to archive a marker to database.
Style.css	General Cascade Style sheet.
Web.config	The configuration file. There is database connection setting.
Web.Sitemap	Keep information of menu bar

Appendix Table B1 (Continued)

File Name	Description
DataAccessLayer.cs	Abstract class for DataAccessLayer
OdbcDataAccess.cs	Connect database using ODBC
OleDbDataAccess.cs	Connect database using OLEDB
OracleDataAccess.cs	Connect database using Oracle Client
SqlDataAccess.cs	Connect database using .NET provider

Appendix C
Database Script

The database schema must be created using the database script below:

```
USE master
GO
CREATE DATABASE flpdb
ON
( NAME = Sales_dat,
   FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Data\flpdbdat.mdf',
   SIZE = 10,
   MAXSIZE = 50,
   FILEGROWTH = 5 )
LOG ON
( NAME = 'flpdb_log',
   FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Data\flpdblog.ldf',
   SIZE = 5MB,
   MAXSIZE = 25MB,
   FILEGROWTH = 5MB )
GO

-- Create TABLE
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[distance]') and OBJECTPROPERTY(id, N'IsUserTable')
= 1)
drop table [dbo].[distance]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[factory]') and OBJECTPROPERTY(id, N'IsUserTable')
= 1)
drop table [dbo].[factory]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[customer]') and OBJECTPROPERTY(id, N'IsUserTable')
= 1)
drop table [dbo].[customer]
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[project]') and OBJECTPROPERTY(id, N'IsUserTable')
= 1)
drop table [dbo].[project]
GO


SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[distance](
      [dist_id] [int] IDENTITY(1,1) NOT NULL,
      [cust_id] [int] NOT NULL,
      [fac_id] [int] NOT NULL,
      [distance] [float] NULL,
      [demand_percent] [float] NULL,
	[project_id] [int] NOT NULL,
 CONSTRAINT [PK_distance] PRIMARY KEY CLUSTERED
(
	[dist_id] ASC,
	[cust_id] ASC,
	[fac_id] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON)
ON [PRIMARY]
) ON [PRIMARY]


-- Customer
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[customer](
	[cust_id] [int] IDENTITY(1,1) NOT NULL,
	[cust_name] [varchar](50) COLLATE Thai_CI_AS NULL,
	[cust_longitude] [float] NULL,
	[cust_latitude] [float] NULL,
	[cust_demand] [int] NULL,
	[project_id] [int] NOT NULL,
 CONSTRAINT [PK_customer] PRIMARY KEY CLUSTERED
(
	[cust_id] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON)
ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF

-- Factory
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[factory](
	[fac_id] [int] IDENTITY(1,1) NOT NULL,
	[fac_name] [varchar](50) COLLATE Thai_CI_AS NULL,
	[fac_longitude] [float] NULL,
	[fac_latitude] [float] NULL,
	[fac_fixedcost] [float] NULL,
	[fac_unitcost] [numeric](18, 0) NULL,
	[fac_transportcost] [float] NULL,
	[fac_capacity] [float] NULL,
	[fac_open] [smallint] NULL CONSTRAINT [DF_factory_fac_open]
DEFAULT ((1)),
	[project_id] [int] NOT NULL,
 CONSTRAINT [PK_factory] PRIMARY KEY CLUSTERED
(
	[fac_id] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON)
ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
SET ANSI_PADDING OFF

-- Project
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[project](
      [project_id] [int] IDENTITY(1,1) NOT NULL,
      [project_name] [varchar](50) COLLATE Thai_CI_AS NULL,
      [project_description] [varchar](200) COLLATE Thai_CI_AS NULL,
      [project_totalcost] [float] NULL,
      [project_rawresult] [text] COLLATE Thai_CI_AS NULL,
      [project_neos_id] [int] NULL,
      [project_neos_password] [varchar] (50) COLLATE Thai_CI_AS NULL,
 CONSTRAINT [PK_project] PRIMARY KEY CLUSTERED
(
      [project_id] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON)
ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF

set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
go

-- Create Stored procedure
CREATE PROCEDURE [dbo].[up_calculatedistance]
      (
            @project_id int
      )
AS
BEGIN

      UPDATE DISTANCE SET distance =
round(ACOS(SIN(RADIANS(fac_latitude))*SIN(RADIANS(cust_latitude))+COS
(RADIANS(fac_latitude))*COS(RADIANS(cust_latitude))*COS(RADIANS(cust_
longitude)-RADIANS(fac_longitude)))*6371, 2)
      FROM  Customer, Factory, Distance
      WHERE Customer.Cust_id= Distance.Cust_id
      AND Factory.Fac_id = Distance.Fac_id
      AND Distance.Project_id = @project_id

      RETURN
END
```

Appendix D
AMPL Model Script and Example Data

AMPL Model Script

```
set I;
set J;

param f {i in I};
param t {i in I,j in J};
param e {i in I};
param d {j in J};

param s {i in I};

var y {i in I} integer >=0, <=1;
var x {i in I,j in J};

minimize Total_Cost: (sum {i in I} f[i] * y[i]) + (sum{i in I, j in J} (t[i,j] + e[i]) *
d[j] * x[i,j]);

subject to Con1 {j in J}: sum {i in I} x[i,j] >= 1;
subject to Con2 {i in I,j in J}: x[i,j] <= y[i];
subject to Con3 {i in I,j in J}: x[i,j] >= 0;
subject to Con4 {i in I}: sum {j in J} (d[j] * x[i,j]) <= (s[i] * y[i]);
```

AMPL Data Example

```
data;
set I:=I48 I53 I54 I55 I56 I57 I58 I59 I60 I61 I62 ;
set J:=J20 J21 J23 J24 J25 J26 J27 J28 J29 ;
param: f e s:=
I48 13 10 100
I53 13 15 200
I54 12 14 23
I55 12 11 20
I56 11 12 13
I57 20 15 17
I58 100 12 167
I59 11 23 14
I60 16 17 18
I61 18 11 222
I62 222 12 11
;
param t:J20 J21 J23 J24 J25 J26 J27 J28 J29 :=
I48 29.4 33 27.4 34.4 44 44.5 34.1 44.2 19.1
I53 10.4 12.8 3.3 17.1 25.8 28.2 8.7 21.3 14.2
I54 14.4 12.5 21.7 7.7 6.9 5.4 25.4 15.9 20.2
I55 10.5 12.1 3.4 16.8 24.7 27.4 5.5 19.1 16.7
I56 10.8 14.4 8.8 16.9 26.6 28 16.4 25.3 6.9
I57 15.9 19 10.4 22.6 32 33.8 15.6 28.6 13.7
I58 11 10.9 18.2 6.8 12.8 12.1 23.7 18.9 13.7
I59 8.8 11.5 13.2 11.2 20.4 20.8 20.6 22.8 4.6
I60 9.5 6.1 15.6 4.4 7 9.8 17.6 8.4 19.7
```

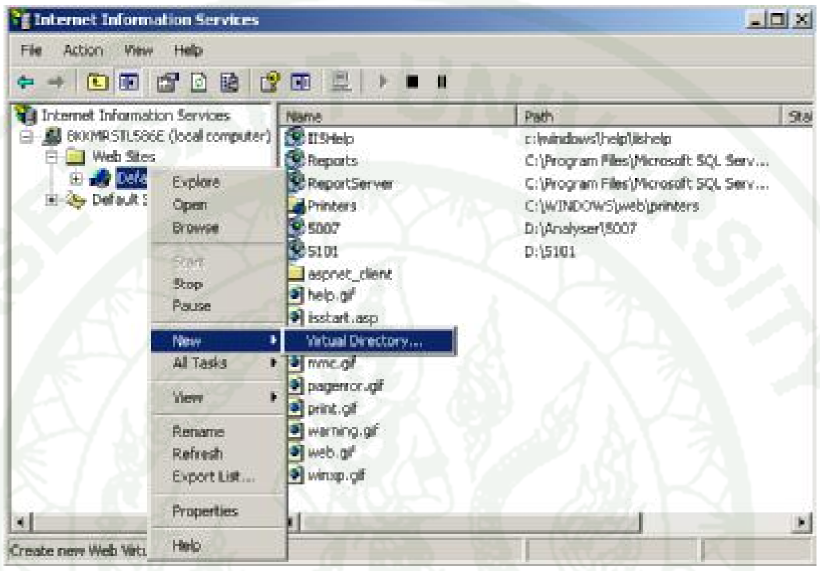
```
I61 8.6 7.3 8.2 11.8 17.4 20.6 6.8 10.4 19.6
I62 4.6 7.4 10.2 7.9 17.6 18.6 17 18.8 7.8;
param: d :=
J20 10
J21 10
J23 25
J24 25
J25 10
J26 2
J27 11
J28 23
J29 50
;
```

Appendix E
Installation Guide and User Guide

Follow up the installation process below to set up the FLP web application.

Web Server Set-up

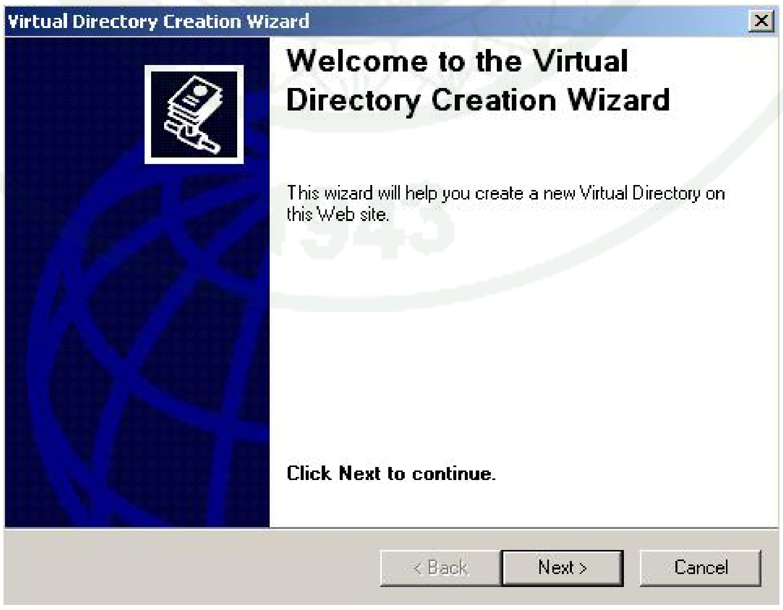
1) Set up the web application.



Name	Path
IISHelp	c:\windows\help\iishelp
Reports	C:\Program Files\Microsoft SQL Serv...
ReportServer	C:\Program Files\Microsoft SQL Serv...
Printers	C:\WINDOWS\web\printers
5007	D:\Analyser\5007
5101	D:\5101
aspnet_client	
help.gif	
iisstart.asp	
mmc.gif	
pagerror.gif	
print.gif	
warning.gif	
web.gif	
winxp.gif	

Appendix Figure E1 IIS Server Setting (1)

2) Click Next button.



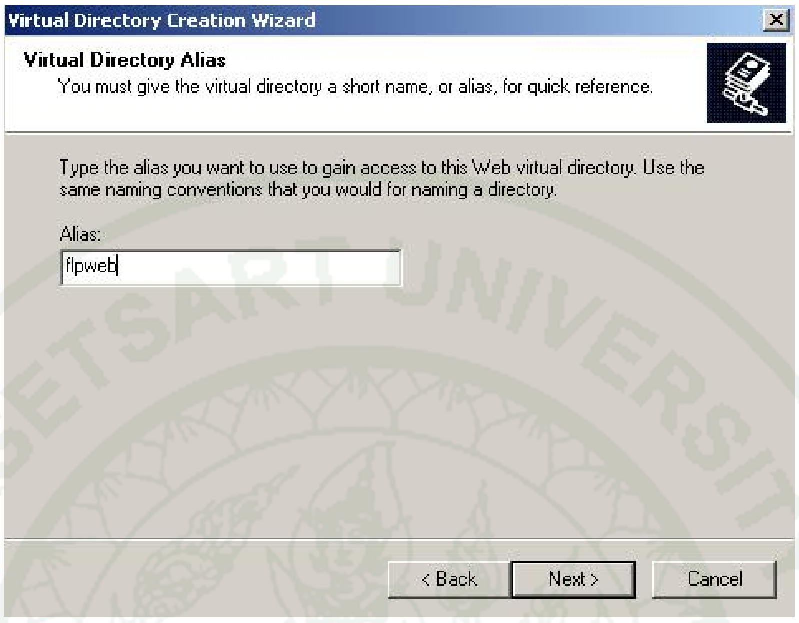
Welcome to the Virtual Directory Creation Wizard

This wizard will help you create a new Virtual Directory on this Web site.

Click Next to continue.

Appendix Figure E2 IIS Server Setting (2)

3) Name the virtual directory. For example, flpweb.


Virtual Directory Creation Wizard

Virtual Directory Alias

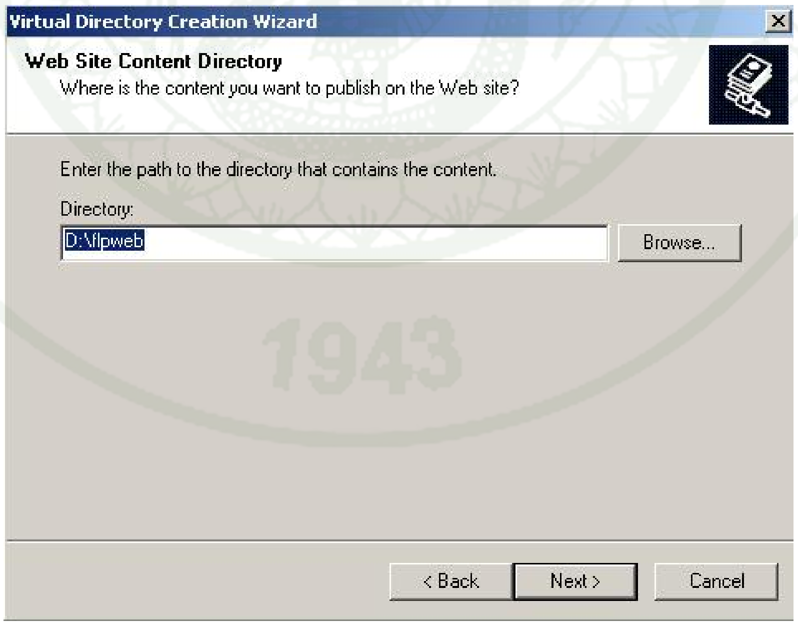
You must give the virtual directory a short name, or alias, for quick reference.

Type the alias you want to use to gain access to this Web virtual directory. Use the same naming conventions that you would for naming a directory.

Alias: flpweb

Appendix Figure E3 IIS Server Setting (3)

4) Browse to the source code location.


Virtual Directory Creation Wizard

Web Site Content Directory

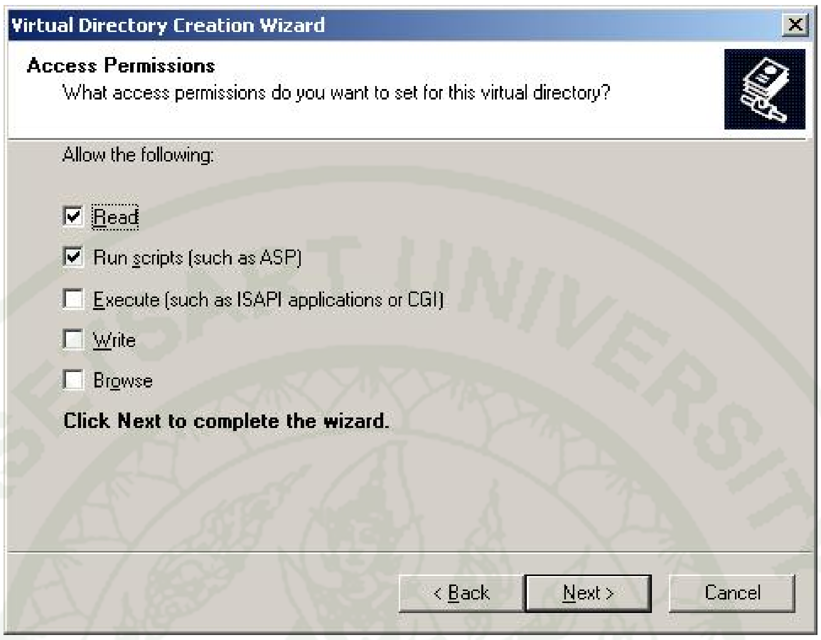
Where is the content you want to publish on the Web site?

Enter the path to the directory that contains the content.

Directory: D:\flpweb

Appendix Figure E4 IIS Server Setting (4)

5) Click Next button



Virtual Directory Creation Wizard

Access Permissions

What access permissions do you want to set for this virtual directory?

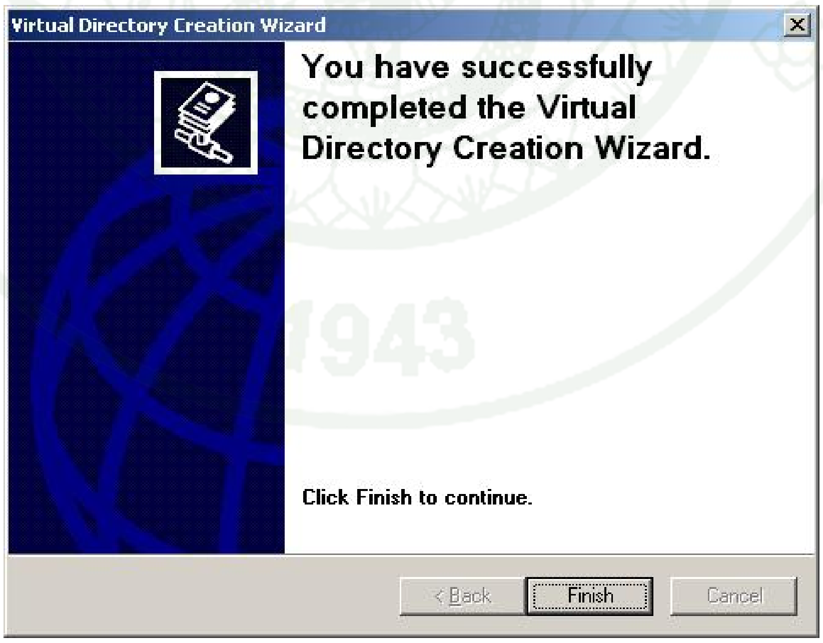
Allow the following:

- [✓] Read
- [✓] Run scripts (such as ASP)
- [ ] Execute (such as ISAPI applications or CGI)
- [ ] Write
- [ ] Browse

Click Next to complete the wizard.

Appendix Figure E5 IIS Server Setting (5)

6) Click "Finish" button.



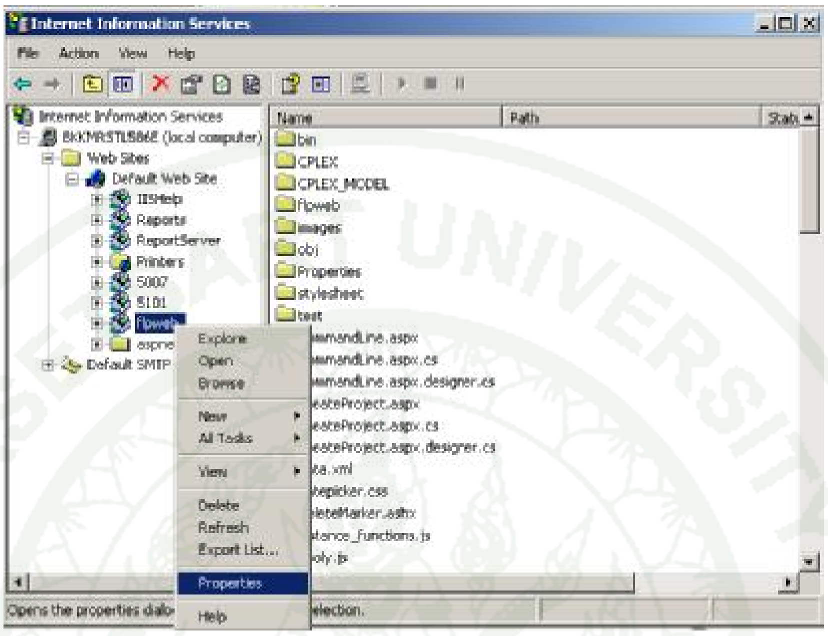
Virtual Directory Creation Wizard

You have successfully completed the Virtual Directory Creation Wizard.

Click Finish to continue.

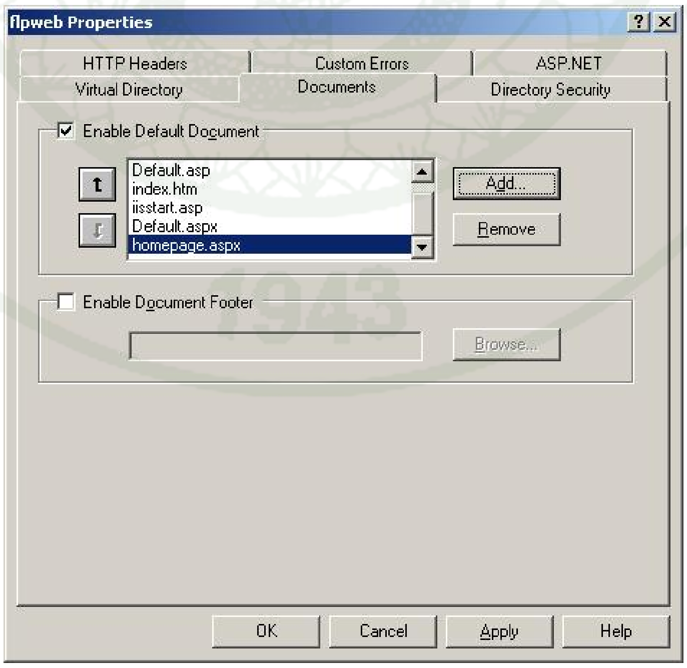
Appendix Figure E6 IIS Server Setting (6)

7) Set “homepage.aspx as the default page. First, right-click on “flpweb” virtual directory name.



Appendix Figure E7 IIS Server Setting (7)

8) Click “Add” button. Then, add “homepage.aspx” as default page.



☑ Enable Default Document

Default.asp
index.htm
iisstart.asp
Default.aspx
homepage.aspx

☐ Enable Document Footer

Appendix Figure E8 IIS Server Setting (8)

Database Connection

The database connection is provided in the configuration file (web.config)

Modify the web.config file.

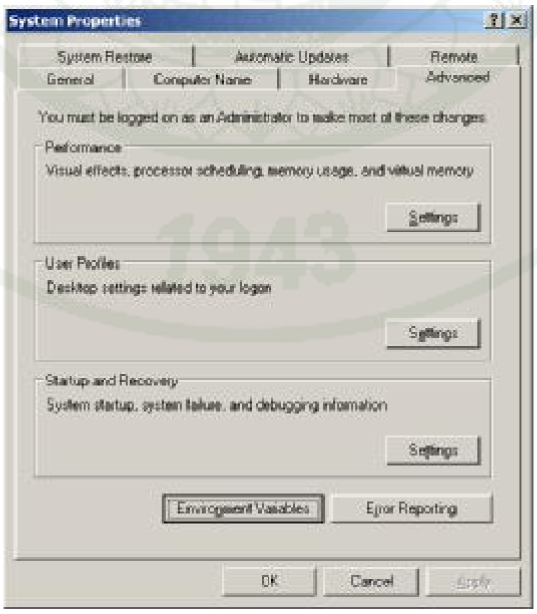
```
<appSettings>
  <add key="DataProviderType" value="SQLServer"/>
  <add key="ConnectionString" value="Data Source=localhost;
            Initial Catalog=flpdb;User Id=sa;Password=xxxxx;"/>
</appSettings>
```

Appendix Figure E9 Database setting in web.config

- **Data Source:** IP Address or machine name's database server.
- **Initial Catalog:** The database name or instance name of database.
- **User Id:** User name.
- **Password:** Password.

AMPL Setting

1) Copy AMPL student version 10.0
2) Set AMPL program directory as Environment Variable. Right-Click on "My Computer". Select "Properties". Then, the "System Properties" dialog will display.



System Properties

System Restore | Automatic Updates | Remote
General | Computer Name | Hardware | Advanced

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings

User Profiles
Desktop settings related to your logon
Settings

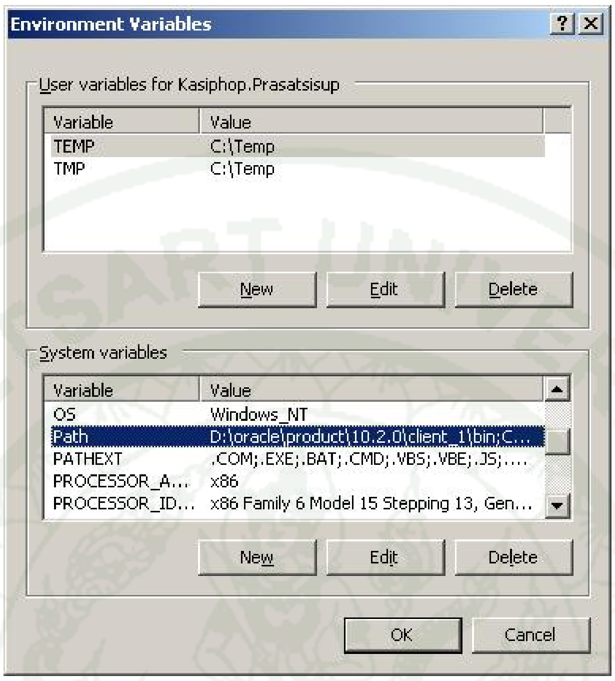
Startup and Recovery
System startup, system failure, and debugging information
Settings

Environment Variables | Error Reporting

OK | Cancel | Apply

Appendix Figure E10 System Properties

Go to Advanced tab. Then, press “Environment Variables” button. The “Environment Variables” dialog will display.

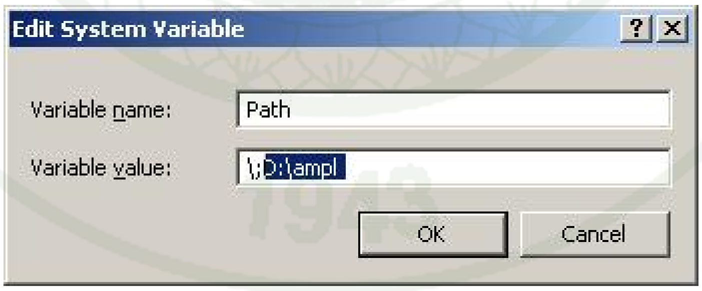


Variable	Value
TEMP	C:\Temp
TMP	C:\Temp

Variable	Value
OS	Windows_NT
Path	D:\oracle\product\10.2.0\client_1\bin;C...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86
PROCESSOR_ID...	x86 Family 6 Model 15 Stepping 13, Gen...

Appendix Figure E11 Environment Variables

At “System Variables” section, select “Path” variable. Then, click on “Edit” button to add the directory path of AMPL. For example, “D:\ampl”. After that, click “OK” button.



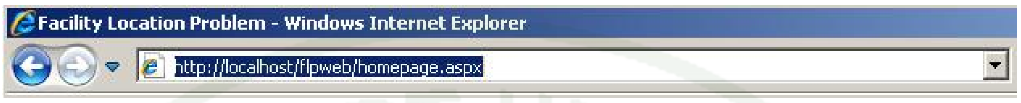
Variable name: Path

Variable value: \;D:\ampl

Appendix Figure E12 Edit System Variable

User Guide

1) Access the web application. Type "http://<IP Address>/<Virtual Directory>/" at address bar.

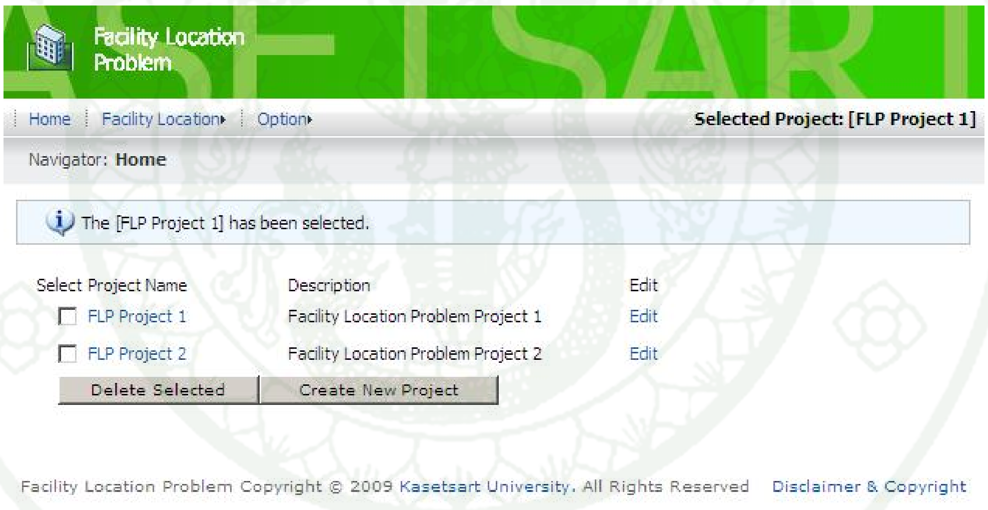

Facility Location Problem - Windows Internet Explorer

http://localhost/flpweb/homepage.aspx

Appendix Figure E13 Address bar

<IP Address> : The web server's IP Address.
<Virtual Directory> : The virtual name that set in IIS.

2) Create your new project by clicking on "Create New Project" button.



Selected Project: [FLP Project 1]

Navigator: Home

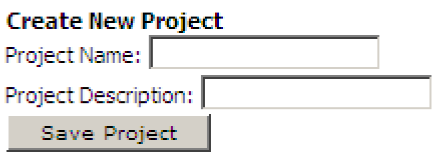
The [FLP Project 1] has been selected.

Select Project Name	Description	Edit
☐ FLP Project 1	Facility Location Problem Project 1	Edit
☐ FLP Project 2	Facility Location Problem Project 2	Edit

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

Appendix Figure E14 Project Selection

3) Fill the Project Name, Project Description and click on "Save Project". After that, select the project that you want again at project selection page.

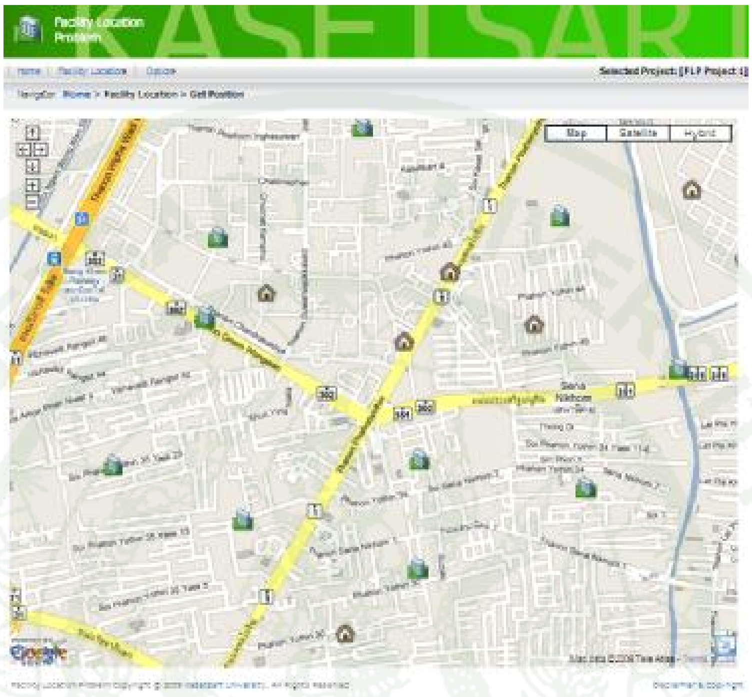

Create New Project

Project Name:

Project Description:

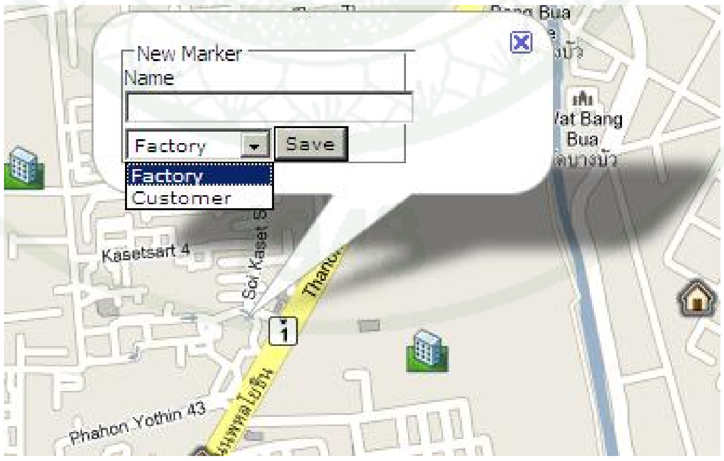
Appendix Figure E15 Create a new project

4) Go to menu bar. Select “Facility Location” > “Get Position”.
5) The Get Position page will be shown. Then, input the facility and customer‘s location.


Facility Location Problem
Home | Facility Location | Option
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Get Position
Map
Satellite
Hybrid

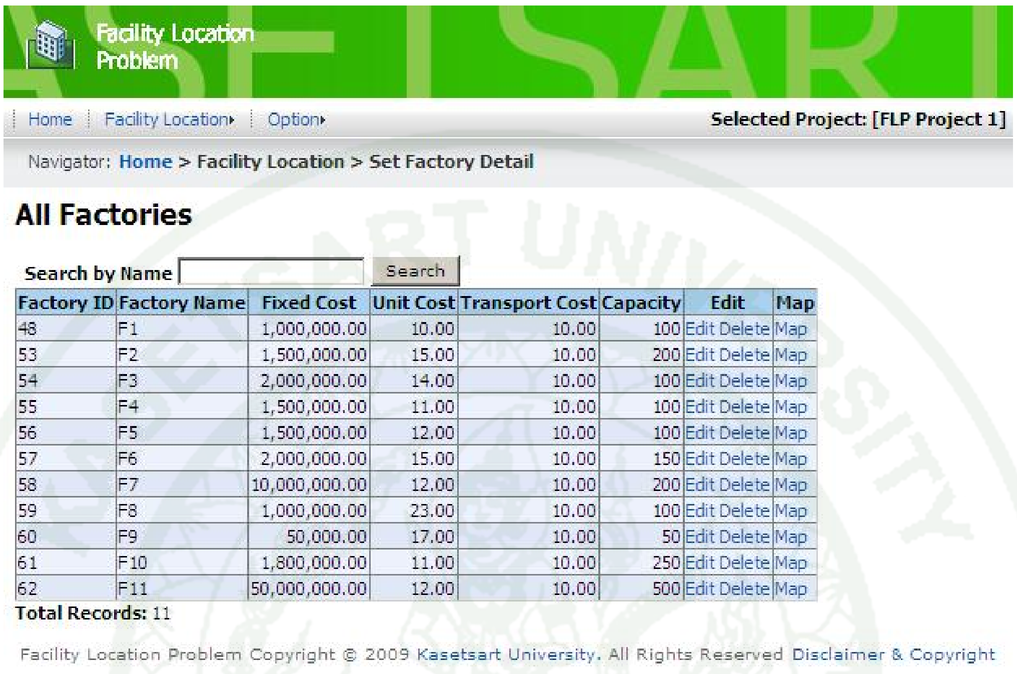
Appendix Figure E16 The customer and facility location

Click on the map and fill name and select type. Then, click save.



Appendix Figure E17 Facility and customer position setting

6) At Menu bar, select "Facility Location"> "Set Factory". Fill the Fixed Cost, Unit Cost, Transport Cost and Capacity in each facility.



Selected Project: [FLP Project 1]

Navigator: Home > Facility Location > Set Factory Detail

All Factories

Search by Name

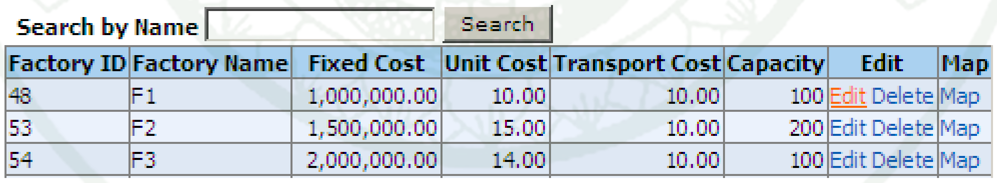
Factory ID	Factory Name	Fixed Cost	Unit Cost	Transport Cost	Capacity	Edit	Map
48	F1	1,000,000.00	10.00	10.00	100	Edit Delete	Map
53	F2	1,500,000.00	15.00	10.00	200	Edit Delete	Map
54	F3	2,000,000.00	14.00	10.00	100	Edit Delete	Map
55	F4	1,500,000.00	11.00	10.00	100	Edit Delete	Map
56	F5	1,500,000.00	12.00	10.00	100	Edit Delete	Map
57	F6	2,000,000.00	15.00	10.00	150	Edit Delete	Map
58	F7	10,000,000.00	12.00	10.00	200	Edit Delete	Map
59	F8	1,000,000.00	23.00	10.00	100	Edit Delete	Map
60	F9	50,000.00	17.00	10.00	50	Edit Delete	Map
61	F10	1,800,000.00	11.00	10.00	250	Edit Delete	Map
62	F11	50,000,000.00	12.00	10.00	500	Edit Delete	Map

Total Records: 11

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

Appendix Figure E18 Set Factories page

Fill detail by clicking on Edit hyperlink.

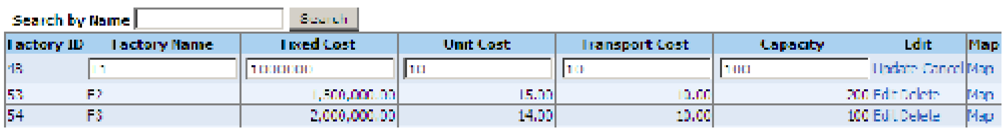


Search by Name

Factory ID	Factory Name	Fixed Cost	Unit Cost	Transport Cost	Capacity	Edit	Map
48	F1	1,000,000.00	10.00	10.00	100	Edit Delete	Map
53	F2	1,500,000.00	15.00	10.00	200	Edit Delete	Map
54	F3	2,000,000.00	14.00	10.00	100	Edit Delete	Map

Appendix Figure E19 Edit Factory information

After finishing, update by clicking on Update hyperlink.

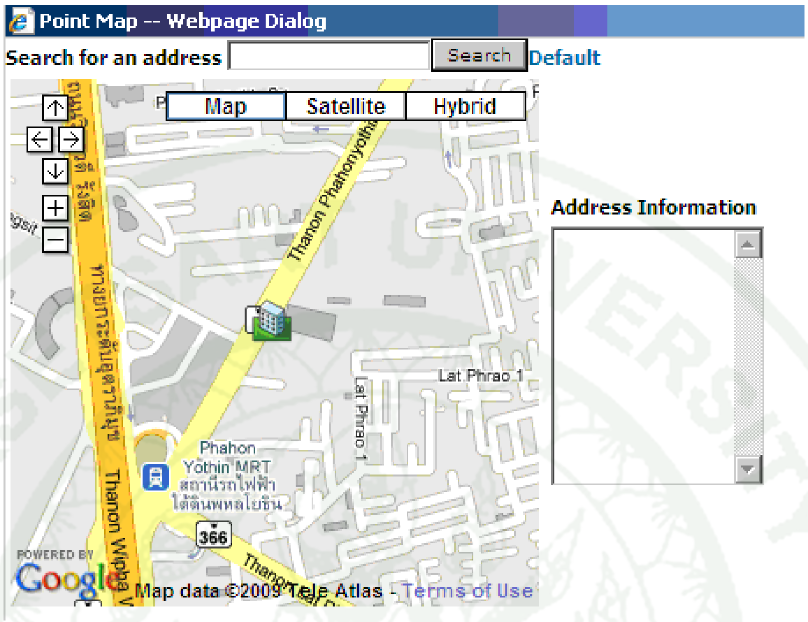


Search by Name

Factory ID	Factory Name	Fixed Cost	Unit Cost	Transport Cost	Capacity	Edit	Map
48	F1	1000000	10	10	100	Update Cancel	Map
53	F2	1,500,000.00	15.00	10.00	200	Edit Delete	Map
54	F3	2,000,000.00	14.00	10.00	100	Edit Delete	Map

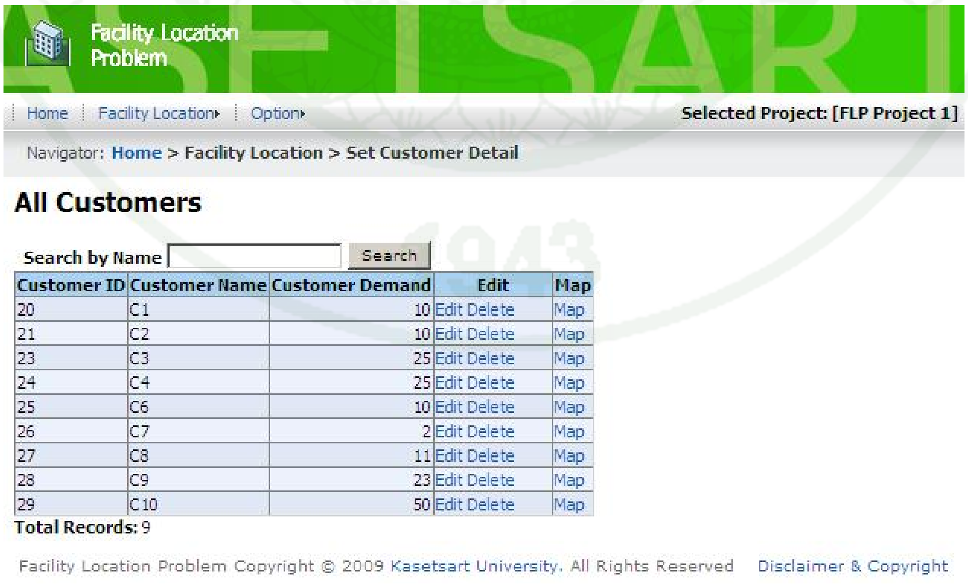
Appendix Figure E20 Edit and update Factory information

The facility location can be change by clicking on Map hyperlink. New dialog will pop-up.



Appendix Figure E21 Point Map page

7) At Menu bar, select "Facility Location"> "Set Customer". Fill "Customer Demand" detail. This page is able to modify, delete and update location as same as facility.


Facility Location Problem
Home
Facility Location
Option
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Set Customer Detail

All Customers

Search by Name [ ] Search

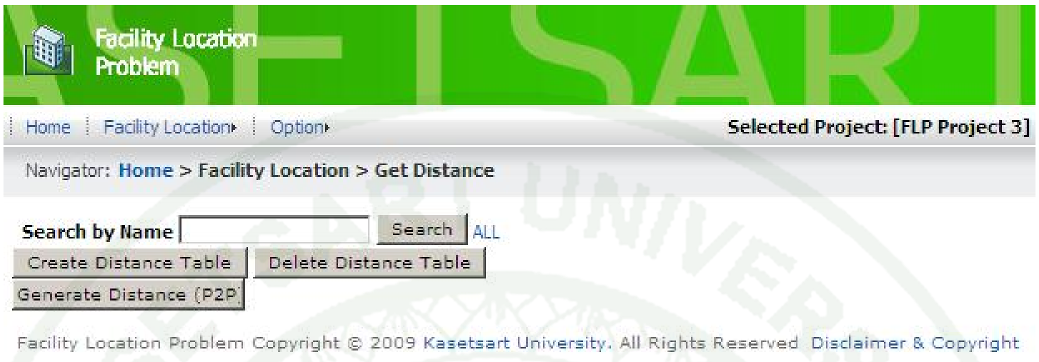
Customer ID	Customer Name	Customer Demand	Edit	Map
20	C1	10	Edit Delete	Map
21	C2	10	Edit Delete	Map
23	C3	25	Edit Delete	Map
24	C4	25	Edit Delete	Map
25	C6	10	Edit Delete	Map
26	C7	2	Edit Delete	Map
27	C8	11	Edit Delete	Map
28	C9	23	Edit Delete	Map
29	C10	50	Edit Delete	Map

Total Records: 9

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

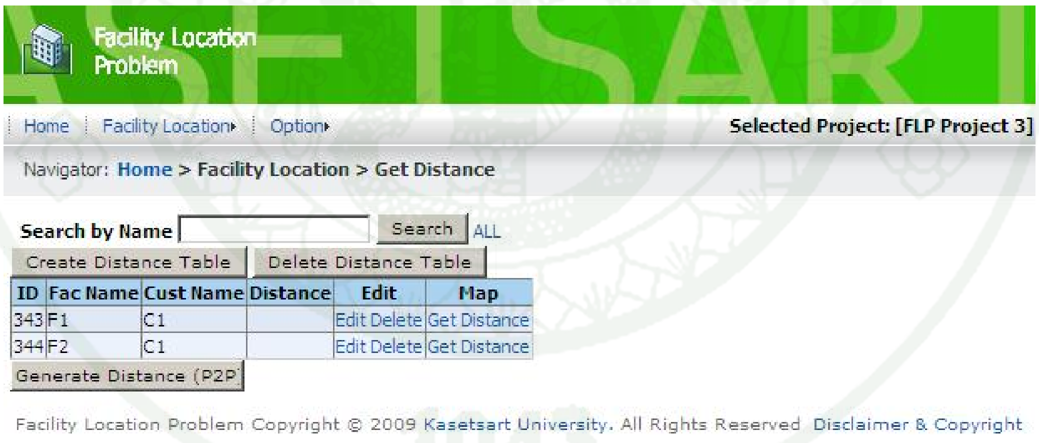
Appendix Figure E22 Set Customers page

8) At Menu bar, select “Facility Location”> “Get Distance”. Create the cross product of facilities and customers by clicking on “Create Distance table” button. Then, fill in distance detail.


Facility Location Problem
Home | Facility Location | Option
Selected Project: [FLP Project 3]
Navigator: Home > Facility Location > Get Distance
Search by Name
Search
ALL
Create Distance Table
Delete Distance Table
Generate Distance (P2P)
Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

Appendix Figure E23 Set Distance page

After clicking on the “Create Distance Table”, then relation between facilities and customers will be created.
Note: The number of facilities and customers are related what we show in other pictures in this appendix

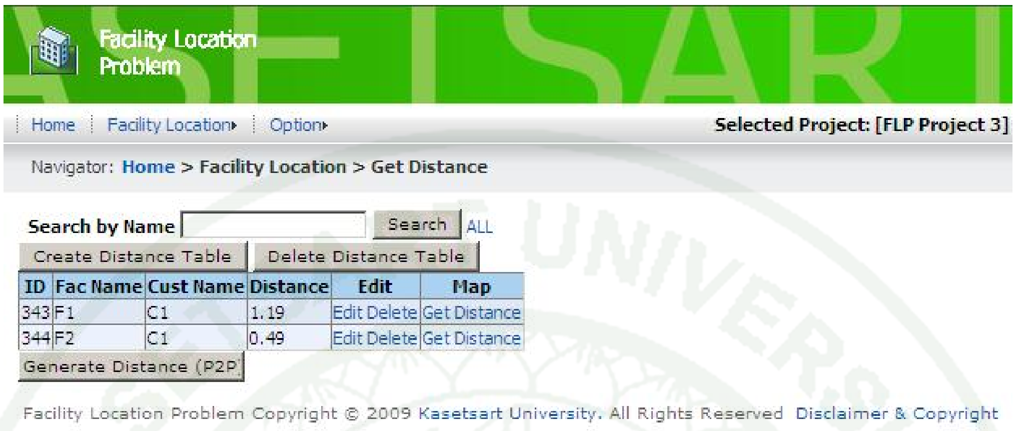

Facility Location Problem
Home | Facility Location | Option
Selected Project: [FLP Project 3]
Navigator: Home > Facility Location > Get Distance
Search by Name
Search
ALL
Create Distance Table
Delete Distance Table
Generate Distance (P2P)
Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

ID	Fac Name	Cust Name	Distance	Edit	Map
343	F1	C1		Edit Delete	Get Distance
344	F2	C1		Edit Delete	Get Distance

Appendix Figure E24 Create distance table

.

User can select distance from two types. First, car driving distance and point-to-point distance.



Selected Project: [FLP Project 3]

Navigator: Home > Facility Location > Get Distance

Search by Name

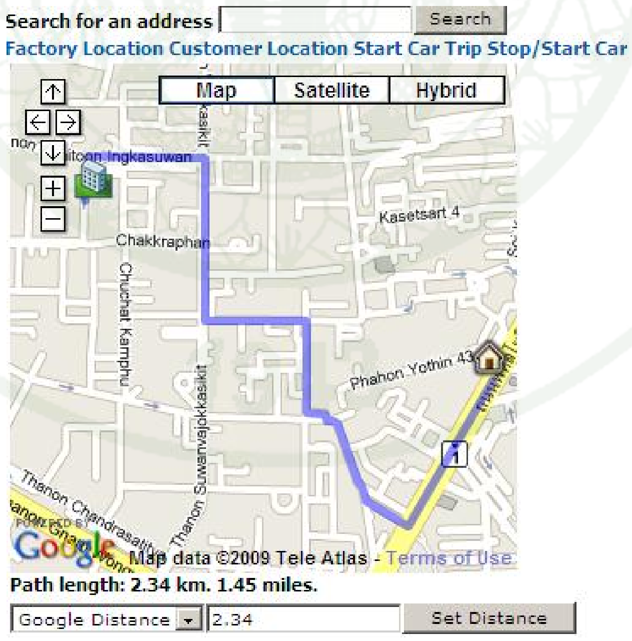
ID	Fac Name	Cust Name	Distance	Edit	Map
343	F1	C1	1.19	Edit Delete	Get Distance
344	F2	C1	0.49	Edit Delete	Get Distance

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

Appendix Figure E25 Create distance information

For 'point-to-point' distance, click on "Generate Distance (P2P)" button. The 'point-to-point' value will fill in all records in Distance column.

For 'car-driving' distance, click on "Get Distance". Then, new dialog will pop-up. Wait for the distance calculation. Then, submit "Set Distance"

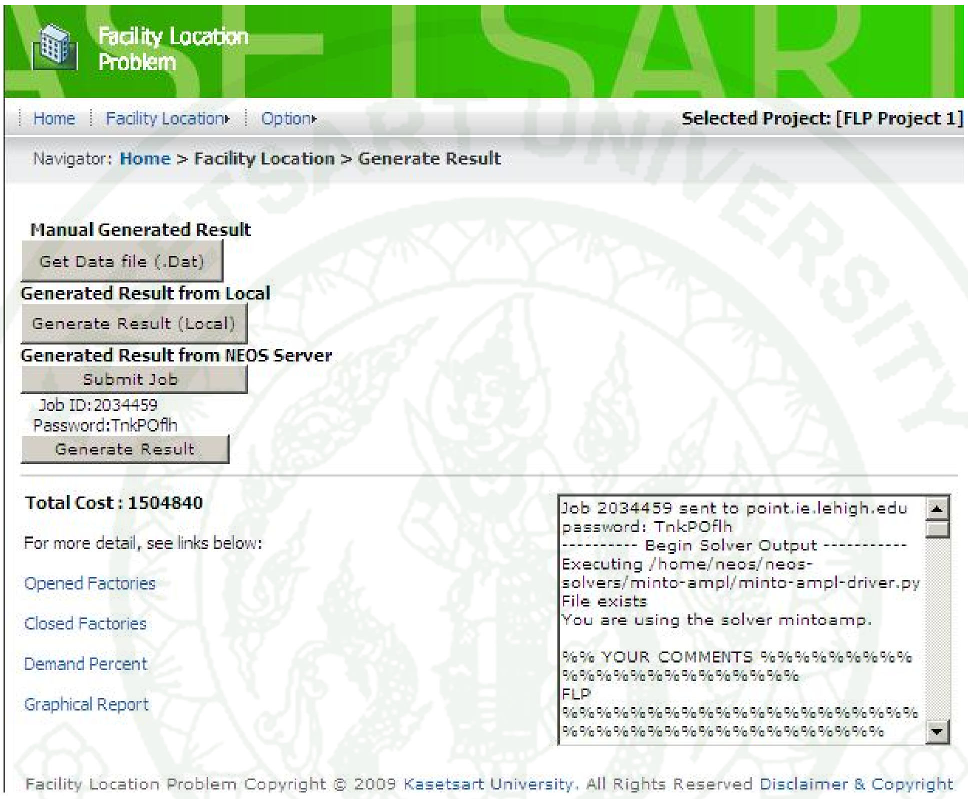


Path length: 2.34 km. 1.45 miles.

2.34

Appendix Figure E26 Distance between factory and customer

9) After complete all inputs facility, customer and distance detail, click “Generate Result” button to get final result. If users need to solve the problem using NEOS Server, users must submit the job to NEOS Server first by clicking on ‘Submit Job’ button. Then, waiting for NEOS Server completes job. After that, click ‘Generate Result’ button.



Job ID:2034459
Password:TnkPOflh

Total Cost : 1504840

For more detail, see links below:

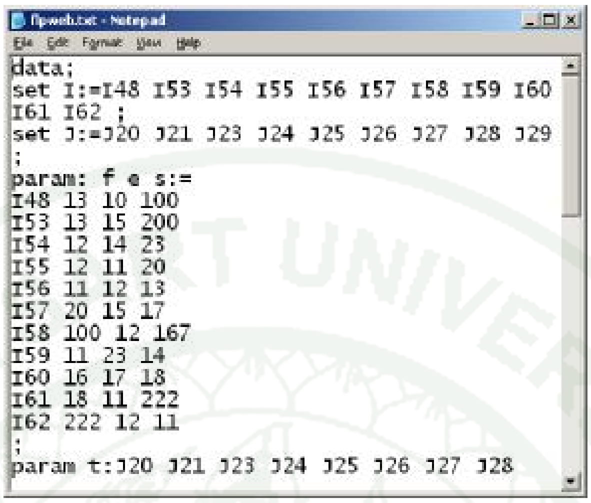
```
Job 2034459 sent to point.ie.lehigh.edu
password: TnkPOflh
---------- Begin Solver Output -----------
Executing /home/neos/neos-
solvers/minto-ampl/minto-ampl-driver.py
File exists
You are using the solver mintoamp.

%% YOUR COMMENTS %%%%%%%%%%
%%%%%%%%%%%%%%%%
FLP
%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
```

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved Disclaimer & Copyright

Appendix Figure E27 Generate Result page

Moreover, if you need the raw data to solve the problem in AMPL by yourself, click on “Get Data file (.Dat)”. it will popup the “flpweb.txt” as below.



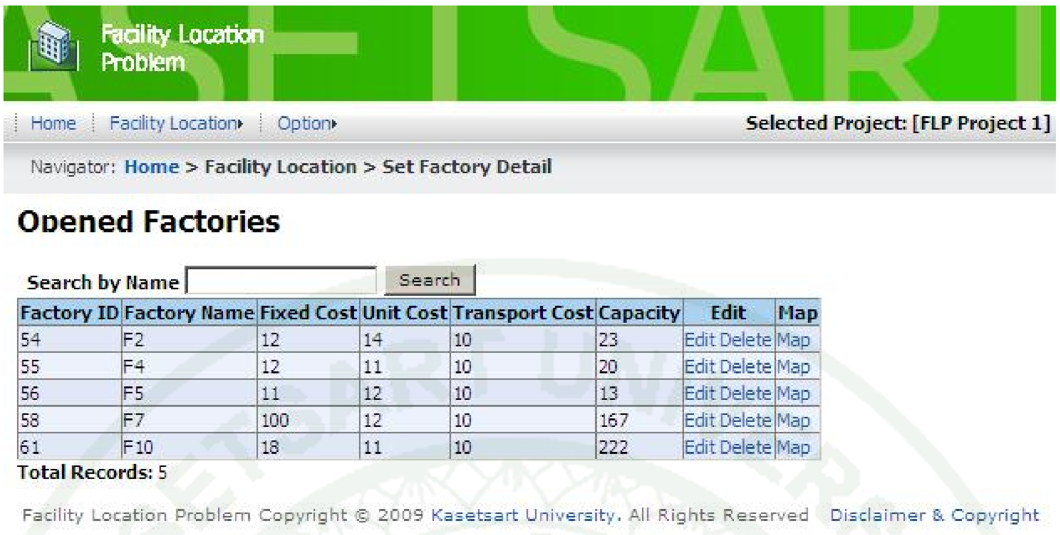
```
data;
set I:=I48 I53 I54 I55 I56 I57 I58 I59 I60
I61 I62 ;
set J:=J20 J21 J23 J24 J25 J26 J27 J28 J29
;
param: f e s:=
I48 13 10 100
I53 13 15 200
I54 12 14 23
I55 12 11 20
I56 11 12 13
I57 20 15 17
I58 100 12 167
I59 11 23 14
I60 16 17 18
I61 18 11 222
I62 222 12 11
;
param t:J20 J21 J23 J24 J25 J26 J27 J28
```

Appendix Figure E28 Get data file

10) After generate result, there are four reports.

Appendix Table E1 Report Description

Report Name	Description
Opened Factories	Show which facilities should be opened.
Closed Factories	Show which facilities should be closed.
Demand Percent	Show how many products that is supplied by facilities to customers.
Graphical Report	Show which facilities should be opened and closed and show relation between facilities and customers in graphical.



Selected Project: [FLP Project 1]

Navigator: Home > Facility Location > Set Factory Detail

Opened Factories

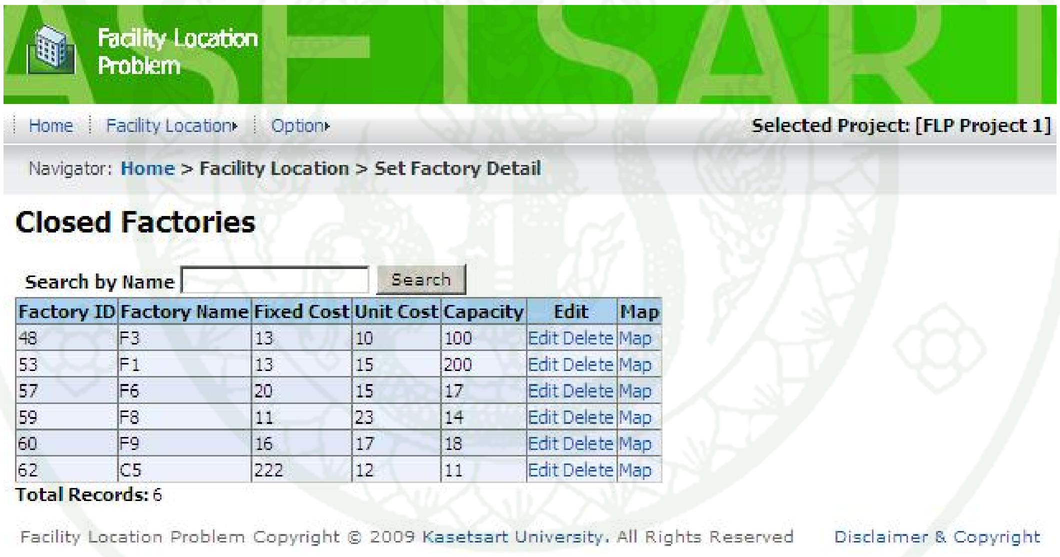
Search by Name

Factory ID	Factory Name	Fixed Cost	Unit Cost	Transport Cost	Capacity	Edit	Map
54	F2	12	14	10	23	Edit Delete	Map
55	F4	12	11	10	20	Edit Delete	Map
56	F5	11	12	10	13	Edit Delete	Map
58	F7	100	12	10	167	Edit Delete	Map
61	F10	18	11	10	222	Edit Delete	Map

Total Records: 5

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved   Disclaimer & Copyright

Appendix Figure E29 Opened Factories report



Selected Project: [FLP Project 1]

Navigator: Home > Facility Location > Set Factory Detail

Closed Factories

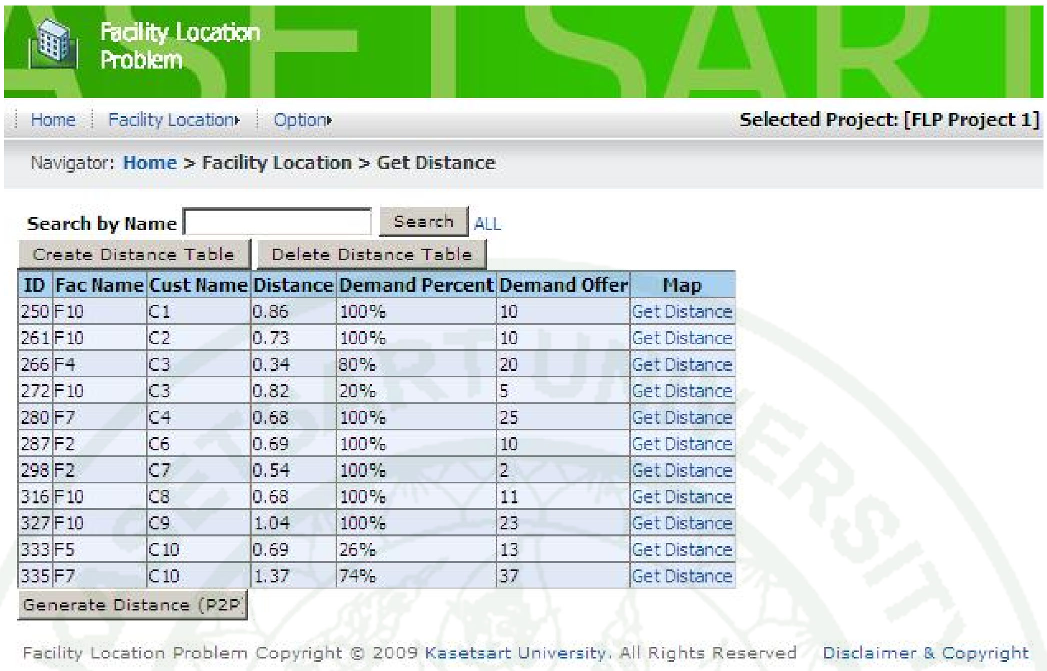
Search by Name

Factory ID	Factory Name	Fixed Cost	Unit Cost	Capacity	Edit	Map
48	F3	13	10	100	Edit Delete	Map
53	F1	13	15	200	Edit Delete	Map
57	F6	20	15	17	Edit Delete	Map
59	F8	11	23	14	Edit Delete	Map
60	F9	16	17	18	Edit Delete	Map
62	C5	222	12	11	Edit Delete	Map

Total Records: 6

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved   Disclaimer & Copyright

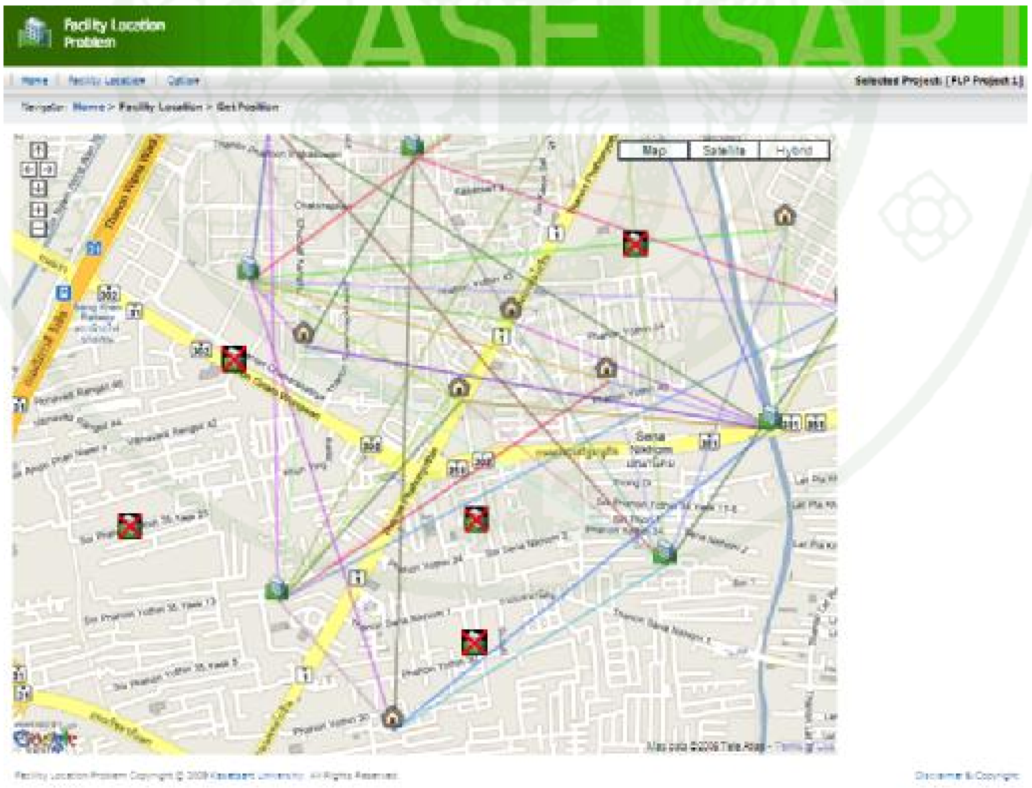
Appendix Figure E30 Closed Factories report


Facility Location Problem
Home | Facility Location▸ | Option▸
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Get Distance
Search by Name
Search
ALL
Create Distance Table
Delete Distance Table
Generate Distance (P2P)

ID	Fac Name	Cust Name	Distance	Demand Percent	Demand Offer	Map
250	F10	C1	0.86	100%	10	Get Distance
261	F10	C2	0.73	100%	10	Get Distance
266	F4	C3	0.34	80%	20	Get Distance
272	F10	C3	0.82	20%	5	Get Distance
280	F7	C4	0.68	100%	25	Get Distance
287	F2	C6	0.69	100%	10	Get Distance
298	F2	C7	0.54	100%	2	Get Distance
316	F10	C8	0.68	100%	11	Get Distance
327	F10	C9	1.04	100%	23	Get Distance
333	F5	C10	0.69	26%	13	Get Distance
335	F7	C10	1.37	74%	37	Get Distance

Facility Location Problem Copyright © 2009 Kasetsart University. All Rights Reserved   Disclaimer & Copyright

Appendix Figure E31 Demand percent report


Facility Location Problem
Home | Facility Location | Option
Selected Project: [FLP Project 1]
Navigator: Home > Facility Location > Get Position
Map
Satellite
Hybrid

Appendix Figure E32 Graphical report

CURRICULUM VITAE

NAME	: Mr. Kasiphop Prasatsisuparp		
BIRTH DATE	: October 1, 1981		
BIRTH PLACE	: Bangkok, Thailand		
EDUCATION	: **YEAR**	**INSTITUTE**	**DEGREE/DIPLOMA**
	2001	KMITL Univ.	B.Eng (Computer)
POSITION/TITLE	: Development Group Leader		
WORK PLACE	: Reuters Software (Thailand) Limited		
SCHOLARSHIP	: International Graduate Program in Industrial Engineering		