



Digital maps on the web and accessibility

Master Thesis
in
Information and Communication
Technology

by
Kim Andre Andersen

Grimstad, May 2001

Abstract

The necessity of location information and knowing where you are at any time, have brought forward the Global Positioning System (GPS) technology. Graphic Information System (GIS) have since the 1980s grown to be used in all kinds of relations. Combining both GPS and GIS solutions have been used for some time now and we have seen it used for navigation at sea. Using the wireless net also mobile phones and Personal Data Assistants can take advantage of this technology. This assignment enlightens different GIS solutions and how any user can use them.

The report is a result of the investigation and will be used as information on new technology for the you-too consortium.

At first I compiled an overview of map solutions on the Internet today. The overview describes possibilities and restrictions the different solutions give. The GIS solutions have many different qualities, but what's most important for the you-too project is user friendliness, costs and if they interact with databases.

The prototype was altered during the project as some of the intended problem already was solved. We finally agreed to make a demonstration of how coordinates from a map can relate with GPS coordinates. Based on accessibility information we use GIS solutions to provide this information. This solution is for the time being only web based, but this can easily be implemented on a Personal Data Assistant. To relate this coordinates I had to connect two databases, one with the location information, and one with facility information.

Preface

This report is the closure of my education for the Norwegian sivilingeniør or Master of Science in Information and Communication Technology (ICT) at Agder University. My assignment has been in progress since January 2001 and will end in May 2001. The assignments workload is equivalent to one semester, 10 weighting.

The Diploma "you-too, digital maps and web" has been composed in co-operation with the candidate and the supervisors. The assignment takes part in the different digital maps that can be used on the web and how they can be combined with accessibility information from a database. One method to get this information is to look at the features in the existing map solutions.

I want to thank Mikael Snaprud and Øyvind Hanssen for all help and good advices and motivation during the work with the assignment.

Grimstad, 28.May 2001

Kim Andersen

Table of content:

[1 Introduction. 6](#)

[1.1 Introduction. 6](#)

Geographic Information Systems - GIS software

- [1.2 The task description. 6](#)
- [1.3 Discussion/Limitation. 6](#)
- [1.4 Background for the thesis. 7](#)
- [1.5 Research focus. 8](#)
- [1.6 Research approach. 8](#)
- [1.7 Who is you-too? 9](#)
- [1.8 Structure of the report. 9](#)
- [2 Classification of Geographic Information Systems. 10](#)
 - [2.1 Vector or raster maps? 10](#)
 - [2.2 Geographic Information Systems - GIS. 11](#)
 - [2.3 GIS formats. 12](#)
 - [2.4 What is metadata? 12](#)
 - [2.5 Cartography. 14](#)
 - [2.6 Location Based Services. 14](#)
 - [2.7 Location technology. 16](#)
- [3 Survey of Geographic Information Systems. 18](#)
 - [3.1 Introduction. 18](#)
 - [3.2 MapInfo. 18](#)
 - [3.3 Map Maker 22](#)
 - [3.4 ESRI 23](#)
 - [3.5 Manifold. 26](#)
 - [3.6 Caliper 28](#)
 - [3.7 Microsoft. 30](#)
 - [3.8 MapServer 30](#)
 - [3.9 Uglund Publikit AS. 33](#)
 - [3.10 Summary. 34](#)
 - [3.11 Discussion. 36](#)
- [4 The Application. 37](#)
 - [4.1 Choice of technologies. 37](#)
 - [4.2 The prototype 37](#)
 - [4.3 Problem. 37](#)
 - [4.4 Technologies. 40](#)
 - [4.5 Use case for present system. 44](#)

[4.6 Implementation of converting from meters to coordinates. 44](#)

[4.7 Java Server Pages and Servlet design. 45](#)

[4.8 Further work 46](#)

[4.9 Security issues. 48](#)

[4.10 User interface 49](#)

[5 Results. 51](#)

[6 Discussion. 52](#)

[7 Conclusion. 53](#)

[8 References. 54](#)

[Appendix. 56](#)

[OLE DB 56](#)

[ODBC. 56](#)

[Access database structure 57](#)

[The original task description. 59](#)

[CODE 60](#)

1 Introduction

1.1 Introduction

This chapter will give the required background and introduction needed for the following chapters. First of all I will introduce the task description I dealt with, which was altered in co-operation with my teaching supervisor. I will then go more into details about the background, motivation and approach of the thesis.

1.2 The task description

- At first I compiled an overview of map solutions on the Internet today. The overview describes possibilities and restrictions the different solutions give.
- I collected information from different databases into a prototype on web. This databases are on different servers (win 2000 server and Jakarta tomcat server). I defined and tested a structured way of doing this.
- Metadata was collected from databases from different locations. The application get current data from a database; this will make it easy to seek on coordinates with a mobile solution. Metadata from a distant database is also collected in to the prototype; seeking on coordinates and type of facility does this. I have made a template for Kristiansand that can lead to further development for other cities.
- Possibilities to connect the map solutions against mobile systems like PDA or mobile phone is outlined.

1.3 Discussion/Limitation

Some of the original task description said this:

To implement a solution I will use Publikit's solution and develop it further so that metadata can be collected from databases from different locations. The application will save current data in a database instead of on a file, this will make it easier to seek on coordinates with a mobile solution.

After some time of the project Geir Andrew from Publikit found that they had solved this in an earlier project. Together with my teaching supervisor we then aimed the prototype to consider the mobile terminals. I should make a prototype that found how to seek on facilities by means of location information from GPS and the location information in a database. The location information from this database will give information so that it is possible to get coordinated data from the access database that contain facility information

1.4 Background for the thesis

Digital map and GIS gets more and more significance in today's community. The computers today makes it possible to save the huge amount of attributes that a digital map uses in databases. These databases represent all kinds of qualities of interest – user can get driving directions, find facilities in the area, user can click on the map and get information on almost everything. From such GIS solutions you can generate digital maps with much more details and information than before the powerful computers.

People with disabilities have enormously widened their boundaries of mobility within the past years. You-too is working on this and provides an integrated system for the collection and publication of information on accessibility information. GIS solutions will give a more user friendly and better way of collecting this information through web. Using a wireless terminal the user is also able to find the facility that fits his/her needs best. And can be guided there with help from a map on the terminal.

By thinking and working geographically, professionals in every field have discovered the advantages of using maps for decision support. With easy-to-use Web browsers, GIS on the Internet provides a much more dynamic tool than a static map display. Web-enabled GIS enables the user to provide interactive queries for e.g.

- Searching for specific site locations
- Displaying and viewing multiple data sets
- Conducting queries for specialized analysis
- Performing data commerce
- Retrieving specialised data services

One of the main benefits of GIS is improved management of an organisation and resources. A GIS can link data sets together by common location data, such as addresses, which helps departments and agencies share their data. By creating a shared database, one department can benefit from the work of another—data can be collected once and used many times.

The latest years we have seen GPS combined with GIS for the purposes of location information. A user can use a wireless terminal (with GPS) to find a location and get information from the desired place.

The GIS technology is very complex and has a lot of qualities, but its still a quite new technology so we have only seen the surface of what we can use this technology for. Especially the prospects for mobile terminals using GIS look exciting.

1.5 Research focus

You-too have a database with accessibility information for people with disabilities located in Hamburg in Germany. In cooperation with a location database located in Grimstad we want to select the information using a map solution or via a mobile device. My focus will be aimed to the current GIS solutions on Internet today and describe them with focus on Interface, PDA and GPS possibilities, database connectivity and costs. These four points are important for the work later on in the you-too project. Further my focus will be aimed to find a good solution for the location database. What data it should contain, what database and so on.

1.6 Research approach

I found that the following approach was the most effective and sensible to solve the problems I was faced with in this master thesis.

1.6.1 Research

- Collected information about the different digital maps from the Internet, and structured that information for my use.
- Use this information to get an overview of the possibilities within the map solutions from the view point of accessibility
- Installation and testing of GIS systems.
- Survey the most used formats for digital maps

1.6.2 Develop a demonstrator

- Setup of experiment environment containing the tomcat server.
- Develop a prototype supporting queries to determine accessibility to facilities, in terms of distance from location and accessibility properties, figure 1-1. (Info from several DBs, Location services, Mobile access)
- Use coordinates to find information and then connect the two databases.

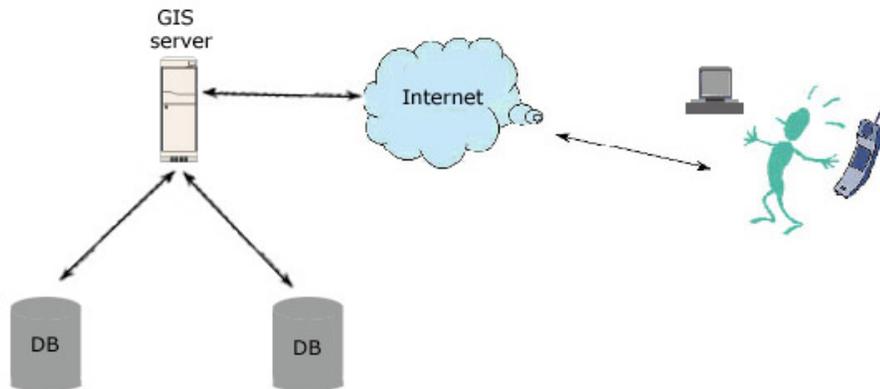


Figure 1 1 Shows where the GIS application is in the network

1.7 Who is you-too?

You-too is working on an integrated system for the collection and publication of information on accessibility of hotels, restaurants, railways stations and other public facilities. They have standardised a collection and presentation of accessibility information and they have all standardised information on several different languages that are available through one single web address.

1.8 Structure of the report

The report is divided into seven main chapters.

Chapter 2 Classification of Geographic Information System, Describes some important properties of GIS solutions. There are two main differences between the GIS solutions, raster- and vector maps.

Chapter 3 Survey of Geographic Information Systems, Describes the different GIS solutions for the Internet today. A table containing some of the characteristics about the different GIS solutions is made to give a short summary.

Chapter 4 the application, goes into detail how the application works.

Chapter 5 results, is the summary of the results

Chapter 6 discussion, is the discussion of chosen technology and chosen GIS solution.

Chapter 7 conclusion, is the conclusion of the thesis.

2 Classification of Geographic Information Systems

2.1 Vector or raster maps?

For a long time there has been a debate on this theme. Vector maps based on public information is superior, but it's a long and expensive process to produce these maps. Today only vector maps are approved as equals to paper maps by the national map organisation. But public approved Raster maps are qualified to use in navigation if it has the necessary backup systems. Which in practice means approved paper maps. We have 3 categories of maps.

1. Rastermaps
2. Vector maps (produced by private firms)
3. Vector maps (produced by public organisation)

Vector maps are divided into private and public raster maps because of the production techniques. Public vector maps is manufactured using a more advanced production technique that will make these maps more reliable than Private vector maps.

2.1.1 Rastermaps

Scanning an already existing map makes a rastermap. Here we can tell there's a big difference on the different raster maps. The solution the scanner gives are crucial to the result of the raster map. The smaller scale on the scanned map the more accuracy. Scanning already existing maps have both benefits and drawbacks. The benefit is that the user sees a known image; the drawback is that this format can't offer the same information as a vector map can. Raster map do not only have to be a copy of a real map, on top of it you can add information about hotels, transportation and other facilities. This makes it easier to use than the conventional paper maps.

An other drawback with raster maps is that they need larger storage space than vector maps, the files are larger which means the downloading time for a vector map is shorter than downloading time for a raster map showing the same information.

By scanning one and one map and have separate files you loose some of the advantage you achieve by using ENC (Electronic Navigational Charts). In some way it's like the old system where you have to open one and one map, but with some programs you can view it seamlessly. A program like this is Publikit.

The user will find that rastermap gives some restricted possibilities when it comes to searching for distances and routing directions. But the user can find information like customer addresses, store sites like restaurants, shopping center etc.

2.1.2 Private vector maps

Private firms often produce private vector maps, first by scanning an official map published by the national map organisation. The data is being handled by vectoring the map. With vectoring graphics, images are represented as mathematical formulas that create bit instructions that in turn define a displayed image. Vectoring graphics are more versatile than bit-mapped graphics because they remain identical when you scale them to different sizes. Now you can save information in different layers and hide and show information on screen by wish. Examples on these can be different facilities, curb lines and so on. By doing this the screen can provide a more perspicuous picture, for example by seeking on hotels that fits the users needs will show.

2.1.3 Public vector maps

Public authorised vector maps ENC (Electronic Navigational Chart). Since these maps are expensive to produce and the resources available to produce them are defective it still takes a long time before these maps are available for larger areas.

The same technical points mentioned above for Private Vector maps also apply to Public vector maps. The difference is more advanced production technique that will make these maps more reliable.

2.2 Geographic Information Systems - GIS

A GIS is much more than simply a program that draws elegant maps. A GIS not only shows a map of a geographic region, it also associates some database with that map. The map can then be used for more than presentation purposes: it can also be used as a "handle" to grab the data that is associated with the map.

For example, a simple map-drawing package might enable you to display a map of Norway showing various cities in different colours. A GIS could add more capability by associating a database of demographic variables such as population or average income with each city. You could then draw a map of Norway that showed each city coloured by a demographic variable, with more populous city's coloured blue and less populated city's coloured green and cities in between coloured some intermediate shade from blue to green. Even a very simple GIS should be able to draw the map based on information from the database associated with the map.

GIS software provides the functions and tools needed to store, analyze, and display information about places. The key components of GIS software are

- Tools for entering and manipulating geographic information such as addresses or political boundaries
- A database management system (DBMS)
- Tools that create intelligent digital maps you can analyze, query for more information, or print for presentation
- An easy-to-use graphical user interface (GUI)

2.3 GIS formats

In this chapter I have made a survey of the most used GIS formats, this survey is made after investigation from several of the GIS providers.

- DLG (Digital Line Graph) Detailed maps of the United States.
- SDTS (Spatial Data Transfer Standard) - 1:25,000-scale maps of the entire United States.
- VPF (Vector Product Format) US Military maps from DMA and NIMA providing very detailed, high resolution maps of the entire world.
- TIGER/Line 1995 and TIGER/Line 1997 - US Bureau of the Census high-resolution maps of civic boundaries, named places, streets with street names and addresses, Zip and Zip+4 coding and more.
- DEM (Digital Elevation Module) Terrain elevation data for US.
- Gtopo30 DEM Terrain elevation data for the entire world.
- ESRI .e00 and "shapefile" formats - Formats used by ESRI GIS products.
- MapInfo MID/MIF - Format used by MapInfo GIS products.

2.4 What is metadata?

Metadata is frequently described as "data about data." Metadata is additional information (besides the spatial and tabular data) that is required to make the data useful. It is information you need to know in order to use the data. Metadata represents a set of characteristics about the data that are normally not contained within the data itself. Metadata could include

- An inventory of existing data
- Definitions of the names and data items
- A keyword list of names and definitions
- An index of the inventory and the keyword list for access
- A record of the steps performed on the data including how it was collected
- Documentation of the data structures and data models used
- A recording of the steps used on the data for analysis

2.4.1 Why metadata is important

Spatial metadata is important because it not only describes what the data is, but it can reduce the size of spatial data sets.

Spatial data also supports software-based and organization wide standards. The benefit of having software-based data standards is that the program is easier to use, and users can readily move data between systems and platforms. By creating metadata, you are creating a standard in naming, defining, cataloguing, and operating standards for all departments. This in turn is a vital foundation for understanding, collaborating, and sharing resources with others. In my application I will take use of the metadata that is in the database, then the users of the GIS can get information on different facilities.

Spatial metadata is important because it supports easier spatial data access and management. Metadata provides a guide to the casual and novice user's question, "How do I know what to ask for?" Metadata can provide information on what is available in an area of interest, where the information is, how current it is, what format it is in, and what use constraints apply. For spatial data professionals, metadata provides feature- and attribute item-level metadata management. This way, updates are easily accommodated and integrated into daily use of the data. Metadata is not an end in itself; it is a tool that will greatly improve your work with spatial data and increase your overall GIS benefits.

2.5 Cartography

Mapping is an essential function of GIS. A map can present data in a fashion that other types of presentation media cannot. And best of all, you do not need to be a skilled cartographer to make maps with a GIS.

Figure 2-1 the map above is a traditional map showing political boundaries and elevations (source: ESRI)

and tables.

Maps from a GIS are created from data in the GIS database. This means that any changes in the GIS database will be automatically reflected in the next printing of a map, allowing changes to a map to be made with minimal effort and cost.

GIS gives you the layout and drawing tools that help you make great presentations with clear, compelling documents. GIS is also being employed as a multimedia technology—delivering digital audio and video information linked to maps, charts,

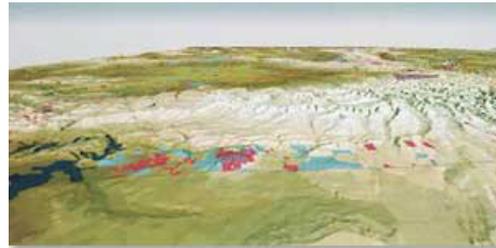


Figure 2 2 the map above is a shaded relief map with a perspective view (source: ESRI)

2.6 Location Based Services

2.6.1 Introduction

When you are going to determine the position of you're mobile phone there are two solutions, handset based and network based solutions.

The mobile Internet has a lot of properties that gives it some advantages compared to the stationary Internet access. The terminal is your own mobile phone or a Personal Data Assistant (PDA) with ip-connection to the net.

In this age of significant telecommunications competition, mobile network operators continuously seek new and innovative ways to create differentiation and increase profits. One of the best ways to do accomplish this is through the highly personalized services that soon can be delivered. One of the most powerful ways to personalize mobile services is based on location. I will discuss Location Based Services (LBS) and the basis of LBS - location technology. But first I will discuss the two positioning solutions.

2.6.2 Handset based location

GPS uses a set of satellites to find the users position. This system has been used for car navigation and in some hand held terminals for some time now. It is now time to use it more in the mobile Internet.

When GPS is used in a mobile phone, you need a receiver that is made especially for this. We can have solutions where the terminal only collects GPS data and sends it to a server for calculation, and the server sends the position data to those who need it.

A-GPS (Network Assisted GPS) is land stations that assist mobile in acquiring its position. These land stations are placed in interval from 200 – 400 kilometres, figure 2-1.

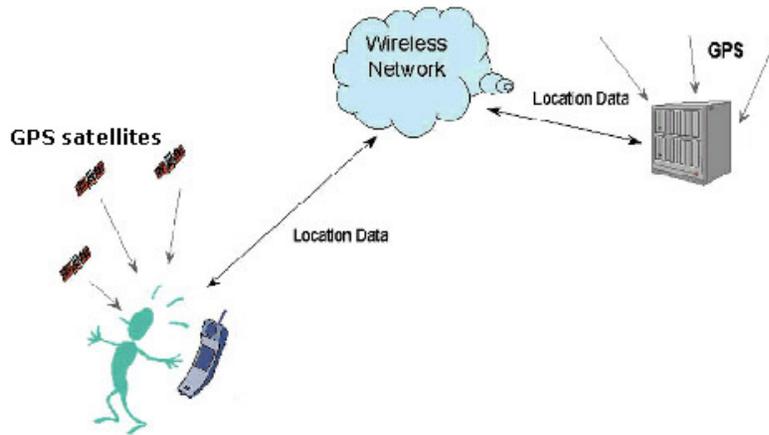


Figure 2 1 Wireless assisted GPS architecture

2.6.3 Network based location

The calculation of data can be done in the terminal or on a server. The accuracy is about 125 meter, and this position technique does not demand the terminal to have free sight to the GPS-satellites. The terminal will get a position even if it's inside a building. The CGI-TA (Cell Global Identity) uses net data about the servers' settlement to find the position of the user. This method can be combined with TA (Timing Advance)-information. TA is the time between the start of a radio frame and a data stream. This information is already built in the net, and the accuracy is acceptable. For services where questions about positions (show me the nearest restaurants) are used this is a cheap and usable solution. One of the benefits is that all terminals can use it. The accuracy varies from 10 meters inside a building to 500 meters outdoor. The cell inside a building is smaller than a cell outdoor, for that reason the accuracy indoor is better than the accuracy outdoor.

2.7 Location technology

2.7.1 Positioning

One of the most obvious technologies behind LBS is positioning, with the most widely recognized system being the GPS. There are however, other means of positioning in addition to GPS. These other technologies are network based positioning and typically rely on various means of triangulation of the signal from cell sites serving a mobile phone. In addition, the serving cell site can be used as a fix for location of the user.

2.7.2 Geographic Information Systems

Geographic data is an important aspect of any location system. Geographic Information Systems (GIS) provide the tools to provision and administer base map data such as man made structures (streets, buildings) and terrain (mountains, rivers). GIS is also used to manage point-of-interest data such as location of gas stations, restaurants, hotels, nightclubs, etc.

2.7.3 Services

There are four major categories of Location Based Services (LBS):

Location based information

Location sensitive billing

Emergency services

Tracking

2.7.3.1 Location based information

Many people are familiar with wireless Internet, but many don't realize the value and potential to make information services highly personalized. One of the best ways to personalize information services is to enable them to be location based. An example would be someone using his or her Wireless Application Protocol (WAP) based phone to search for a restaurant. The LBS application would interact with other location technology components to determine the user's location and provide a list of restaurants within a certain proximity to the mobile user.

2.7.3.2 Location based billing

The ability to have preferential billing is provided by this type of application. Through location based billing, the user can establish personal zones such as a home zone or work zone. Through arrangements with the serving wireless carrier, the user could perhaps enjoy flat-rate calling while in the home area and special rates while in other defined zones. This type of application can be especially useful when use in conjunction with other mobile applications such as prepaid wireless.

2.7.3.3 Emergency services

It is a location based emergency service application that pinpoints your location and relays it the appropriate authorities. The FCC (US Federal Communication Commission) has mandated that by the end of 2004, all wireless carriers in the United States must provide a certain degree of accuracy in pinpointing the location of mobile users who dial 9-1-1. EU has the same demand, but not before 2008.

In Norway we have <http://www.resqcom.com/> provided by Telenor Radio Systems AS.

2.7.3.4 Tracking

This is a large category that contains everything from the difficult fleet applications to enabling mobile commerce. Fleet applications typically entail tracking vehicles for purposes of the owning company knowing the whereabouts of the vehicle and/or operator. Tracking is also an enable of mobile commerce services. A mobile user could be tracking and provided information that he has predetermined he desires, such as notification of a sale on men's suits at a store close to the user's current location.

3 Survey of Geographic Information Systems

3.1 Introduction

GIS solutions for the Web are for everyone who wants to provide interactive access to maps and data on their web site. Retailers can enable customers to find the closest store, government agencies can provide the public with easier access to data, and businesses can give their employees access to geographic analysis tools. Most of the information collected for this survey is taken from the different companies who provide the GIS solutions.

3.2 MapInfo

The cornerstone of all MapInfo solutions is location information. Customer addresses, phone numbers, store sites, service and sales territories. Also travelling individual with a cell phone can use this product, with a WAP telephone this individual can get a wealth of other valuable information including customer buying patterns, demographic and lifestyle information, nearby businesses, routing directions, traffic patterns and more. All this information is stored in databases, and the databases are built up with rows and columns. Each row contains information about a particular event or facility. Each column contains a particular kind of information about the items in the table.

One product that MapInfo offers is *MapInfo Spectrum*. This product has some features that are useful and simple to use for Internet users. This program can e.g. get information from database. This database can contain information like accessibility, address, etc. This can be useful information for people with disabilities who are planning vacation or just have dinner plans on a restaurant.

The map is generated using *MapInfo MapExtreme Java*, their Internet mapping server, and is based on *MapInfo StreetPro*, a nationwide, highly detailed street database.

This product also provides route and driving directions that the user can take use of when driving from one point to an other. *MapInfo Routing J Server* generates this.

Business Data

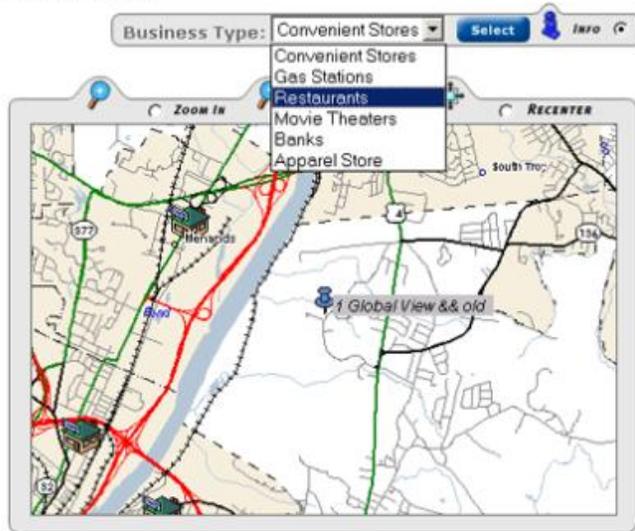


Figure 3 1 From the drop down list, select a category of business data. Use the Info Tool to click on any point to see information about the business.

MapInfo Routing J Server lets you add turn-by-turn driving directions to any web-based application (US only). Calculating either the shortest distance or quickest timed route between any two points, the Routing J Server returns text-based driving directions and spatial points to your users. The Routing J Server is best for applications demanding greater customisation, control and security - ideal for customers with appropriate web, database and Java expertise.

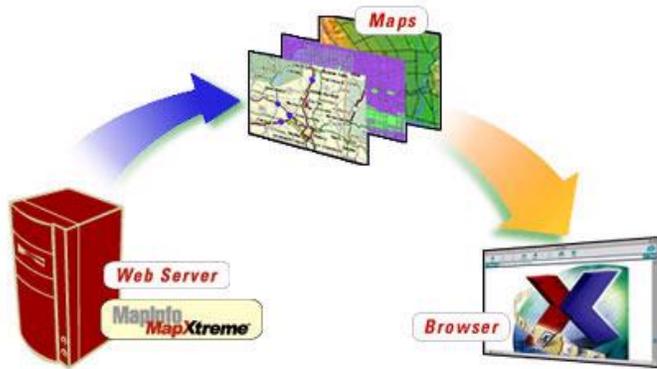


Figure 3 2 MapInfo MapXtreme Java Edition is used for Internet deployment of applications.

With a simple click, users can see their wireless coverage area, find the closest drug store or get driving directions to their next sales calls. And network engineers can visually spot disruptions in their network - automatically notifying customers of service outages.

MapInfo StreetPro Norway is as mentioned above, a nationwide, highly detailed street database. It's MapInfo's premier street product for Norway, it provides up-to date and accurate street details for the creation of quality presentation maps. The features of MapInfo's StreetPro:

- StreetPro Norway contains 13 layers of information such as roads, from motorways to streets, railway lines, urban areas, Fylke and Kommune boundaries, water and more
- Coverage for Norway, mainly Oslo
- Updated twice a year to ensure you have the most accurate and comprehensive mapping resource.
- Available with peace of mind subscription pricing so you'll always have the most up-to-date data to work with.
- Complete with MapInfo workspaces for faster and easier viewing and analysis.

Strengths: New cartographic capabilities and significant "live" database connectivity

Weaknesses: Data formats from other vendors are accessible via translation, but not direct.

3.2.1 Range of use

The cornerstone of MapInfo is location information. Using MapInfo you can locate customer addresses, phone numbers, store sites, service and sales territories.

3.2.2 Interface

MapInfo's interface gives the user the ability to search in many layers. The user can find customer addresses, store sites like restaurants, shopping center etc. Service and sales territories offer more valuable information such as customer buying patterns, the influence of competitors, local businesses, demographic and lifestyle information, routing directions, traffic patterns and much more. You can also use a WAP telephone to search on e.g. restaurants then you get the nearest restaurants and information with phone number, address and even a map that guides you through to the restaurant.

3.2.3 MapInfo and Internet connectivity

MapInfo allows you to link your maps on the Internet through its new Internet connectivity. Any object in a map can contain a URL that is automatically launched in the default web browser when a user clicks on that object. e.g. link a company web site to the point location on your map that identifies the corporate headquarters. You can also use link to other types of document files that the Windows operating system can launch (doc, xls, ppt, tab, wor, mdb, etc.).

MapInfo Professional can generate HTML Image Maps that can be included in web pages. When a user clicks on a HTML Image Map, the browser links to new web pages. MapInfo can generate an HTML Image Map from any map window that contains a layer of region objects.

3.2.4 MapInfo and database connectivity

Version 5.0 adds "live" database access. MapInfo has had the ability to connect to open database connectivity (ODBC) databases for some time, but these connections used to involve downloading a dataset from a remote database management system that could then serve as a local MapInfo data source. Although this is a powerful capability, it presented a problem in that local versions of the data could diverge from the remote/enterprise data. In version 5.0, this ODBC connection is now live. As changes occur to the remote/enterprise data, they're automatically reflected in the local MapInfo version.

3.3 Map Maker

The program — a simple geographic information system or GIS — is designed to create and manipulate maps on IBM-compatible computers that run Microsoft Windows. The results can then be saved and shared electronically, printed, or used in other GIS programs.

One product from Map Maker is *Map Maker Pro* it's a low-cost and easy to use product, it is powerful, and it's a GIS for Windows. It is designed to enable non-expert users to start producing useful maps and I think they have managed that. A variety of easy to use tools allow you to navigate around a map, measure distances and areas, draw polygons, lines and symbols, and display and edit data. *Map Maker Pro* can print maps directly on to any printer or plotter, which is fully supported by Windows. *Map Maker Pro* can export images for inclusion in documents produced on compatible Windows word processors, such as *Microsoft Word*, *Word Perfect*, and *Lotus Word Pro*. The program contains powerful data analysis and customisation tools and can read data from many other programs. *Map Maker Pro* can use scanned images, in colour or black and white. It can export and import vector graphics to and from programs such as *AutoCAD*, *ArcView*, *ArcInfo*, *MapInfo* and more. In addition, *Map Maker Pro* has a number of easy-to-use tools for making field surveys and creating maps from survey data. Non-expert users with no previous experience of surveying can map a farm, urban development, an archaeological dig, a site of special scientific interest, etc.

Eric Dudley, an independent consultant on community-based development aid based in Cambridge, England, designed map Maker. But there were also institutions involved in the development, particularly the World Conservation Union (IUCN), the Centro Agronómico Tropical de Investigación y Enseñanza (CATIE), and the Asia desk of the United Nations Centre for Human Settlements (UNCHS).

3.3.1 Range of use

Map Maker was originally designed for projects in developing countries where low-cost and ease of use are essential. Since then the program has proven itself useful in a wide range of other fields, notably archaeology, geology, agriculture, and environmental monitoring.

3.3.2 Interface

A variety of easy to use tools allow you to navigate around a map, measure distances and areas, draw polygons, lines and symbols, and display and edit data. The program contains powerful data analysis and customisation tools and can read data from many other programs.

3.3.3 Mapmaker and Internet connectivity

At this time Mapmaker cannot publish maps to the web. But later this year (2001) they will produce an Internet add-on for doing this, but it is not available yet.

3.3.4 Mapmaker and database connectivity

Mapmaker does not have any database connectivity at this time.

3.4 ESRI

Geodata AS is the distributor of ESRI products in Norway. With ArcView and suitable Norwegian street data you can seek in many ways both geographically and on qualities or a combination.

ArcView Streetmap is an American product. In Norway there is nothing equivalent. But there are solutions that can find addresses, route calculations etc. that is based on ESRI's Internet solutions:

- MapObjects Internet Map Server
- ArcIMS (latest technology)

ArcView GIS software lets you see, explore, and analyze spatial data on your desktop computer. Its basic mapping functionality and advanced geographic information system (GIS) capabilities allow you to create maps that show, integrate, and view your data.

ArcView StreetMap solves problems associated with address and point of interest locations.

- Find an address and zoom to its location.
- Find the locations of many addresses at the same time.
- Zoom to any area on the map to display state boundaries, roads, lakes, rivers, parks, airports, hospitals etc.
- Click on any facility to display information about it.
- Perform detailed analysis of a particular area and create a customized local street data set.
- Extract any street data to a local shapefile for further analysis.

Detailed analysis of a particular area can be performed by creating a customized local street data set. The data set can be used in routing applications or to study the relationships between streets and other features using the analysis capabilities of *ArcView* GIS. *ArcView StreetMap* also provides additional Avenue classes and requests. Use them to customize how *ArcView StreetMap* data are displayed and managed in your application.

ESRI has a server taking care of publishing and editing the maps on the Internet. This server is called *ArcIMS*. This server provides the framework for distributing GIS services on the Internet. *ArcIMS* adds geographic data and analysis to such diverse applications as e-commerce, enterprise resource planning (ERP), data warehousing, customer care and support, location-based services, and field data integration. *ArcIMS* is a ready to use toolset for authoring, designing, and administering Internet GIS sites.

ArcIMS operates in a distributed environment and consist of both clientside and serverside components. It works the same way as most servers, the client sends a request for information from an Internet or Intranet server. Then the server processes the request and sends the information back to the client viewer.

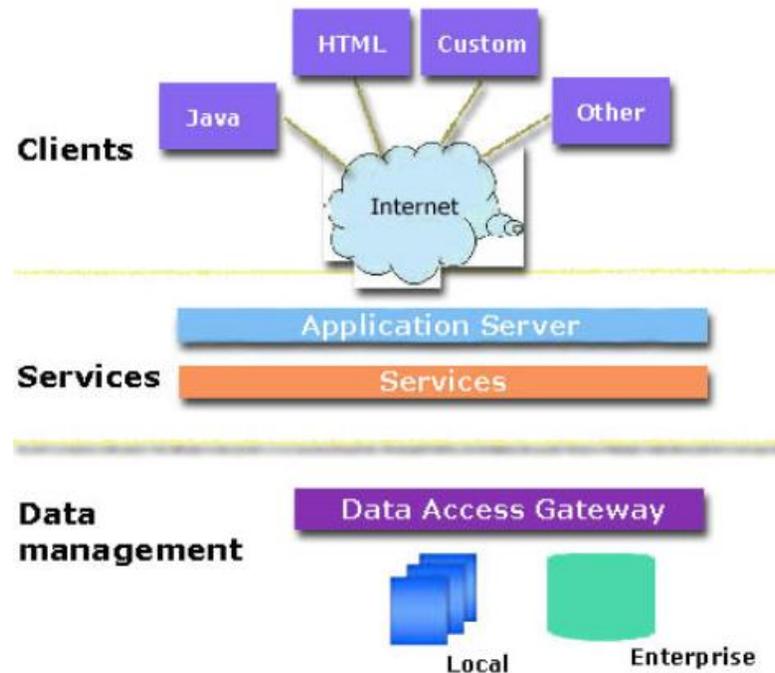


Figure 3 3 ArcIMS three-tier Architecture

3.4.1 Range of use

The range of use in all of ESRI's products is enormous. You can do almost everything you want, but you'll have to pay for it! ESRI's products can be used in various contexts. For instance, Network Analyst can be used in transportation, logical analysis, streamflow etc. using Streetmap you can zoom and collect the information you want from facilities, roads, etc. this product allows you to view simple or multiple locations in the map.

3.4.2 Interface

Through ArcView's interface you can do the same things or even more than their biggest competitor, Map Info. ArcView has an easier interface and does not require so much effort to get started. You can use the SQL queries to find an address and zoom to its location, you can also find many addresses at the same time. You can use the zoom to display almost everything you want, like state boundaries, roads, lakes, rivers, parks, airports, hospitals etc.

The user can click on any facility to display information about it. You can also extract any street data to a local shapefile for further analysis if that's wanted. New geographic data can be created with ArcView from existing tabular databases or by digitising of graphics elements. Field data can be collected with GPS and the resulting points, lines and areas can be mapped directly with ArcView.

3.4.3 ArcView and Internet connectivity

ArcView uses ArcIMS to publish images and maps to the web. Using ArcIMS users can set up and administer Internet services, it can be accessed remotely to administer a site. ArcIMS allows users to define mapping application content. Generating an online map involves adding data content and setting other map properties that create a map service. The output from the ArcIMS is a map configuration.

3.4.4 ArcView and database connectivity

Using ODBC technology you are allowed to query an outside database directly from ArcView. Data linked to ArcView projects via ODBC can be automatically updated to reflect changes in the outside database.

3.5 Manifold

Manifold allows you to use the map to "grab" data. For example, you can click on those restaurants you want, and then ask Manifold about the restaurants' accessibility information's. This is an example of more sophisticated GIS analysis where one uses the visual image of data (the coloured drawings of the facilities in the map) as a visual handle to grab the data in which we are interested. To achieve this level of sophistication, Manifold includes a database engine that can manipulate databases as part of the map.

The Manifold GIS approach is easy. See the data in the map and click on what interests you. You can also incorporate relationships that are important to you, such as selecting only those facilities that lie relatively close to some location of interest. Such relationships are immediately understandable when looking at a map and clicking with the mouse but very hard to write down as a correct SQL statement.

Manifold provides the best flexibility and freedom to go back and forth at will within the mapping and database worlds for the convenience of the user. Manifold, for example, has a very rich repertoire of selection capability that works the same whether it is invoked from within a map view, within a text table view of databases, or from within any of the analytic solver systems Manifold provides.

Manifold also includes a variety of statistics, logic, mathematics, and networking capabilities for use on maps, on databases, or all together at the same time. Manifold does all this while preserving at all times the freedom to go back and forth between what used to be different worlds. For example, not only can you go back and forth between databases and mapping, you can also invoke the world of network analysis to understand a better way to lay out facilities data within the city's or invoke statistics solvers to get instant statistics on the data the user selects.

To help you get data into Manifold, the system includes a wide variety of data import and data discovery capabilities. There are numerous importers and converters for all sorts of situations. Some of these are highly specific to GIS use and some are specific to networking. Some converters are aimed at converting pre-existing data into Manifold formats. Other tools, such as the GPS Console, may be used to create new maps by fetching positional data from GPS receivers.

The capabilities I have described above in the context of GIS software are important ideas in areas other than GIS. After all, the general idea of linking a database to a map is still a good idea even when the "map" might be a CAD blueprint of a factory floor, of a printed circuit board layout, or when it is a diagram of a computer network. In all of these cases, one can easily think of situations where it would be very useful to be able to associate a drawing with a database and then fetch data out of the database by clicking on the drawing instead of by struggling with SQL or with text tables.

3.5.1 Range of use

Manifold has a very rich repertoire of selection capability that works the same whether it is invoked from within a map view, within a text table view of databases, or from within any of the analytic solver systems Manifold provides. Manifold Database Commander (MDC) is a universal command and control console for working with almost any database. MDC support HTML, this means that you can get data directly from Internet and in to MDC. It also takes care of statistics, analysis, multilevel sorts and updates to data in a simple way.

3.5.2 Interface

Through Manifolds Interface you can browse almost any data file or database, also SQL Server. You can transform data at will with a point and click, without any programming, also including type changes or format changes. Perform analysis without having to write complex SQL queries or database programs.

Run statistics on fields and automatically report results.

You can harvest data from Web sites for use in your own databases, doing this you can easy take down competitors price list and compare with your own prices.

3.5.3 Manifold and Internet connectivity

Manifold System Release 5.00 includes a simple web server as part of the product. You also need the Internet Information Server that is built into all Windows server products. You can add, delete, move, alter and link anything you want to this map. Once created an image, map or dataset you can publish it to the web.

3.5.4 Manifold and database connectivity

Manifold support both ODBC and OLE DB connectivity.

3.6 Caliper

Caliper is the developer of **Maptitude**.

Maptitude for the Web applications are scripts that run on your web server. The scripts call Maptitude for the Web functions that produce web page elements such as graphics files (PNG or JPEG) and data tables. The output of the script typically is either HTML or XML, and does not require plug-ins or applets to be viewed in a browser. While Maptitude for the Web is responsible for all of the geographic computations, the script controls the actual output format sent to the browser. By editing the script, you can target your application to different clients such as Palm Web Clipping Applications, WAP/WML, CHTML, WebTV, VoiceXML, or OpenGIS.

Web Site Administrators can easily add Maptitude for the Web to one or more servers. No additional application server software is needed beyond Maptitude for the Web. All applications include a built-in load balancing system for scaling the performance over multiple server computers.

Administrator can also edit, display and map his own data. You can use open database connectivity to do this.

The Mapplication that calliper offer is updated periodically to take advantage of new server and browser capabilities such as ASP+, Visual Basic.NET, DHTML, XHTML, and SVG.

There is less third-party data available with Maptitude than with ArcView and MapInfo, but Maptitude can import much of the data intended for these products. You'll also find fewer consultants available for this product than for the others, if you need to customize it. Otherwise Maptitude might be the choice for you-too project.

3.6.1 Range of use

On callipers homepage they say "Maptitude Geographic Information System for Windows is the intelligent mapping solution for business, government, and education." But with a map over for instance Kristiansand and facility information in a database an administrator can publish the Maptitude on web with this information.

3.6.2 Interface

Caliper has put a lot of effort to make it easy for the administrator to make an easy user interface. But from the examples on the web the user has to know a lot from the city or place that he want to search within.

3.6.3 Maptitude and Internet connectivity

All of Maptitude's standard capabilities are available for use on the Internet with *Maptitude for the Web*, available separately. It includes the development tools necessary to design, test, and publish interactive map applications on the Internet. The product works on Windows 95, 98, or NT and 2000 with a Web server. Maptitude for the Web produces HTML output and does not require installation of additional application server software or browser plug-ins.

3.6.4 Maptitude and database connectivity

Maptitude 4.1 reads Excel spreadsheets, dBASE files, and text files directly. It is ODBC compliant, so it can read other ODBC compliant data sources.

3.7 Microsoft

Microsoft is the developer of MapPoint 2000. MapPoint 2000 has the same features as many other GIS products I've studied. When Microsoft combines MapPoint 2000 with other Microsoft Office products you can see a big difference in ease of use in the products. You can easily drag and drop information from for instance Access, Outlook or Excel to MapPoint 2000 and a wizard will guide you through the mapping of the data. Or you can take data from the map and drag it into some other office solution.

3.7.1 Range of use

MapPoint 2001 is a capable mapping program aimed squarely at the business user. A business can take advanced analysis of their costumers and find where they stay, what they buy etc. By doing this they can for instance advertise in the local papers that their customers get. They can also use MapPoint 2001 to give them an advanced route description with MapPoints route planner. MapPoint 2001 allows you to import mapping data from databases and export to html. It can also be used to get information about restaurants, hotels and other attractions. And you can update and add facilities to any map.

3.7.2 Interface

If you're used to use Office products, MapPoint 2001 will give you a head start because of its familiar functionality and design.

3.7.3 MapPoint 2001 and Internet connectivity

MapPoint enable users to save maps as .html pages or copy them into FrontPage or other Web site creation programs, but it is not available as a dynamic online application at present.

3.7.4 MapPoint 2001 database connectivity

MapPoint 2001 support both ODBC and OLE DB connectivity.

3.8 MapServer

MapServer is an OpenSource development environment for building spatially enabled Internet applications. The software builds upon other popular OpenSource or freeware systems like Shapelib, FreeType, Proj.4, libTIFF, Perl and others. MapServer can run on Linux/Apache platforms as well as MS DOS platforms. MapServer can compile on most UNIX servers and will run under Windows NT/2000/98/95.

The install and configure procedure [1] is very easy, took me about 10 minutes to install and get it out on the web. I installed it on a Windows 2000 machine, using the MS information server (IIS).

Trough the testing stage of this application I did not have the time to go deeper into the program and learn how to get my own queries, layers, databases and maps into the application. I used a demo version that was available and by using this I was able to find what MapServer could do for the users.

The MapServer system supports MapScript that allows scripting languages such as Perl, Python, Tk/Tcl, Guile and even Java to access the MapServer C API. MapScript provides a rich environment for developing applications that integrate disparate data. If the data has a spatial component and you can get to it via your favourite scripting environment then you can map it. For example, using Perl's DBI module it is possible to integrate data from just about any database vendor that support ODBC compliant data sources with traditional GIS data in a single map graphics or web page.

3.8.1 Range of use

I found that MapServer is not a full-featured GIS system, because it does not provide the core functionality described in chapter 2.2. MapServer don't aspire to be one either. A full-featured GIS system that also is an OpenSource is GRASS (Geographic Resources Analysis Support System). But this is a desktop solution, not Internet. MapServer does, however, provide enough core functionality to support a wide variety of web applications. Beyond browsing GIS data, MapServer allows you create "geographic image maps".

Here's a sampling of what MapServer provides:

- vector formats supported: ESRI shapefiles, simple embedded features, ESRI ArcSDE
- raster formats supported (8-bit only): TIFF/GeoTIFF, GIF, PNG, ERDAS, JPEG and EPPL7
- quadtree spatial indexing for shapefiles
- fully customisable, template driven output
- feature selection by item/value, point, area or another feature
- TrueType font support
- support for tiled raster and vector data (display only)
- automatic legend and scalebar building
- scale dependent feature drawing and application execution
- thematic map building using logical or regular expression based classes
- feature labeling including label collision mediation
- on-the-fly configuration via URLs
- on-the-fly projection

3.8.2 Interface

MapServers interface is up to the administrator to make user friendly, you can make your own user interface and publish on the web.

Through MapServers interface you are able to get all the info the administrator allows, for instance roads and facilities.

The only problem is that the administrator has to use some time to get up the details of the system, because there is no user interface to apply new maps, databases, etc.

3.8.3 MapServer and Internet connectivity

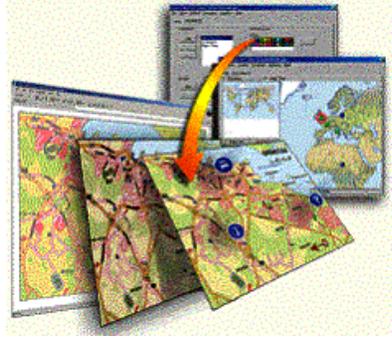
MapServer uses MS IIS, Apache server or Netscape servers to connect to the Internet. All are available free of charge. (I presume the IIS will only run on a proprietary Operating System provided by the same supplier, which makes it limitedly free of charge)

3.8.4 MapServer and database connectivity

Using ODBC technology you are allowed to query an outside database directly from MapServer.

3.9 Ugland Publikit AS

Figure Error! No text of specified style in document.-1 **user interface from Publikit 1.4**



Ugland Publikit AS is the developer of Publikit 1.4. Publikit is a complete web page editor and Internet-publishing system. It integrates a template technology for the easy handling of information structures. This template makes it possible for anybody to publish and maintain information on the Internet. It

also integrates a template for GIS solutions, so anybody can scan a map and publish it on the Internet. This is a unique property for web publishing tools.

Ugland Publikit AS is about to integrate ArcIMS into Publikit, which enables the clients to get vector maps from the ArcIMS. But this functionality will take some time and will probably be on their next release (Publikit 1.5).

3.9.1 Range of use

Publikit is not a GIS tool, but a web-publishing tool. It offers some support for GIS solutions, but only rastermap. There are image-editing functions, so you can edit the map and add symbols for hotel, restaurants, etc.

3.9.2 Interface

Publikit's interface is well arranged and thereby easy to use. The templates make it easy for the administrator to publish and edit the map on any .html sites.

3.9.3 Publikit and Internet connectivity

Publikit allows you to publish rastermap on the Internet through any .html site.

3.9.4 MapInfo and database connectivity

At this time there is no "live" database access. But you can connect to any database using for instance ASP script or JSP. Doing this you can get all information on restaurants, hotels, etc.

3.10 Summary

Comparison as of May 2001								
Product	Publikit 1.4	MapInfo PRO 6.0	ArcView 3.1	MapMaker	Manifold 5.0	Maptitude 4.0	MapPoint 2000	MapServer
Company	Ugland Publikit AS	MapInfo	ESRI	MapMaker	Manifold	Caliper	Microsoft	Open source
Operative System	All	Win95-2000 NT	UNIX Win95-2000 NT	Fully	Win95-2000 NT	Win3.1 - 2000	Win95-2000 NT	UNIX Win95-2000 NT
Ease of Use	High	High	Moderate	High	High	High	High	Poor
Free Viewer	Yes	Yes	Yes	Yes	Use ArcExplorer	No	Yes	Yes
Features:								
Continuous thematic shading	No	Yes	NO	Yes	Yes	No	Yes	Yes

Geographic Information Systems - GIS software

DBMS Support	No	All ODBC compliant	All ODBC compliant	All ODBC compliant	All ODBC compliant	All ODBC compliant	All ODBC compliant	All ODBC compliant
OLE automation	No	Yes	No	No	Yes	Yes	Yes	No
Extensions	No	Yes	Yes	Yes	Yes	No	No	No
Geocoding	No	Yes	Yes	Yes	By Web Server	Yes	Yes	Yes
Crystal Reports	No	Yes	Yes	No	No	No	Yes	No
Web Access	Yes	Yes	Yes	No	Yes	Yes	No	Yes
PDA interface	No	Yes	Yes	No	No	No	No	No
GPS mapping	No	Yes	Yes	Yes	Yes	Yes	No	almost
Programming Language	Java	Proprietary (MapBasic), Visual Basic, PowerBuilder, Delphi, C++,	Proprietary (Avenue)	Proprietary	Visual Basic, Visual C++, Visual Basic Scripting, JavaScript, Python and PERL	Proprietary (GIS DK)	Visual C++, MFC, Visual Basic	Perl, Python, Tk/Tcl, Guile and Java
Price	-	\$1700	\$1,195.00	\$265	\$145	\$995	\$109	Free
Price to publish on Internet	-	\$5500	-	-	No extra	\$2,995	-	Free

Table 3 1 comparison of GIS solutions as of May 2001

Description of the table:

ODBC compliant data sources, including Access, Btrieve, DB2, Informix, Ingres, InterBase, NetWare SQL, Oracle, Paradox, PROGRESS, SQLBase, SQL Server, and Sybase.

Ease of Use: Based on interface, the ease of publishing on the Internet and administration of the map.

Free viewer

Continuous

Thematic Shading: Gives the opportunity to give shades on the map. So that different part of the city can have different shades based on for instance population. Makes it easier to see the landscape.

DBMS support:

OLE automation: The GIS have a set of APIs to create and display a document. The automation part provides the possibilities for a container or server application to drive that document.

Extensions: You can get extensions for this product. GIS extensions make it easy to add new features. Extensions are plug-ins that you can load and unload as you need them.

Geocoding: Products that add spatial coordinates to data (latitude/longitude) and keep address information clean and accurate.

Crystal Reports: Makes it easy for the users to make sophisticated reports. Can get useful information about customers and so on.

Web Access: Possibilities to publish the map solution on Internet, not only images but also with searching possibilities.

PDA interface: The GIS provider has the map solution also in a PDA.

GPS mapping: The GIS provider uses GPS to show location of the user.

Programming

language: Which programming language the map solutions use.

3.11 Discussion

As we can read from the table 3-1 most of the GIS applications I have investigated have the important qualities [1,5] for a GIS solution, but they are expensive. We found one solution that might fit the you-too consortium; the MapServer witch is open source. MapServer has the needed qualities, but is more difficult to install and use than the other GIS solutions, they will need a person with experience or the ability to get up the details from this application.

GIS solutions from ESRI and MapInfo already support PDA interface and have some demo applications where they use GPS with some mobile terminal. But because of the new technology the prices are high.

4 The Application

4.1 Choice of technologies

The purpose of the prototype was to show how we could connect two databases together and to show how to use location information to find topical facilities. Before I started the implementation I got a map over Kristiansand in a GIS application from Publikit. From this map I got the coordinates to all facilities on the map. I had to use these coordinates to find the distance between the user and the different facilities. To do this I chose to store the data in a Postgresql database. The Postgresql database supports the datatype *point* (x, y), which I used to store the location information in. Using this datatype I could use some stored functions in Postgresql that made it easier to make the select sentences. Since I had access to a Postgresql database, the choice was obvious. I had knowledge of the Access database from earlier work[3], but it had some missing data so I had to do some minor adjustment to it[123], to get the test result I wanted.

Because the Postgresql database only supported JDBC connection I was forced to get into JSP to get the database connection. I could also used a driver called *type 1 driver*, it comprises drivers that are able to access JDBC databases via an ODBC – JDBC bridge, but it has several deficiencies which I will discuss later. To get into JSP took some time since I never had used this before. When using JSP I needed a server that serves the .jsp pages and run the servlet that my site contains. For this I chose Jakarta Tomcat. Because of the time limit I used ASP and VBScript, where it was possible. The choice of ASP and VBScript is based on previous experience.

The best solution for this application would be to use JSP and servlet from scratch to the end. This is because of easier maintenance for the administrator. I have discussed why I didn't do this in the previous section.

4.2 The prototype

To test and view the prototype navigate to this location:

<http://128.39.145.12/kaande99/location.htm>

4.3 Problem

Publikit cannot search on the facilities using location information. This means Publikit don't support GPS. I will make a demonstration of how PDA with GPS hand devices will work when you want e.g. the restaurants within 200m (figure 4-1).

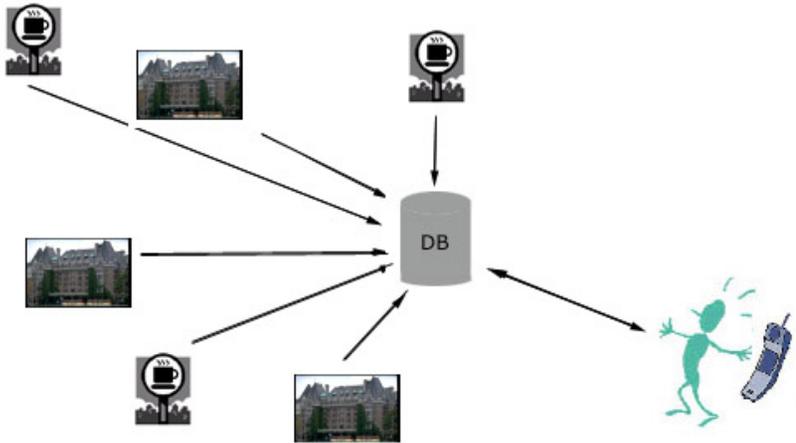


Figure 4 1 the user can find facilities by searching through the database with his mobile terminal

At the same time I will try to connect two databases together, one database with location information and one with facilities information. I will connect them together using a unique ID in both databases. I will use one Postgresql database and one Access database. When the user sends his location, type of facility and distance information to the application the application will return the information of current interest. To make this happened I would use one SQL sentence that selects the correct information from the database.

1. Take care of the incoming coordinates, and distance
2. Convert the distance to be e.g. 200m à 1 coordinate, make a function to do this.
3. Use the select sentence to get the facilities within the distance.
4. Another select sentence towards the Access database to get the facilities that the user wants.
5. Return the facility information to the application.

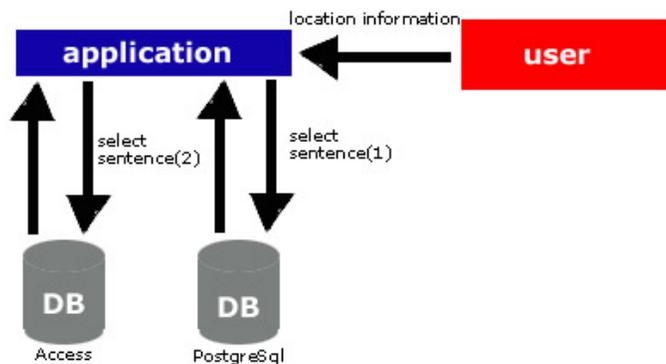


Figure 4 4 2 Shows the data flow when user sends location

The *user* uses the application to search for facilities near by. The application handles the two select sentence and write the result to the user (Figure 4-2).

Select sentence (1): Sends the coordinate- and radius information to the Postgresql database. In return it gets the ID of all the facilities within the range of the search.

Using a predefined function in Postgresql:

- “*Select id from facilities where point_distance(location, '"+altitude+"', '"+longitude+"') < '"+radius'”*

point_distance: Returns the items that is within range of the radius from the point(altitude, longitude)

Select sentence (2): Sends the facilities *ID* and *type* of facility information to the access database. In return it gets the facilities within the range of the search and only the facilities of the right type.

4.3.1 Database structure

The Postgresql database (facilities) will provide the website (location.jsp) with information the user can use to navigate into other websites for more detailed information (info.asp). To get valuable information I use the location field in the facilities database which keeps track of the facility position. This way the user with his PDA using GPS can get information on facilities nearby. The Access database with all detailed information about each facility is predefined and it will be used only to show how to select information from it, based on information from the Postgresql database.

Postgresql structure:

```
create table facilities (  
ID int,           // Unique identification  
location point,  // coordinates  
name char(40),   // facility name  
map_id int,      // for later use what map the for facility  
type int         // type of facility  
);
```

Access structure:

The access structure was predefined, but I had to make some changes to make it work in my prototype. To view the tables in the database go to the [Access database structure](#) in the appendix.

4.4 Technologies

4.4.1 Java Server Page

Java Server Page (JSP) is a technology for controlling the content or appearance of Web pages through the use of servlet, small programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it. Sun Microsystems, the developer of Java, also refers to the JSP technology as the servlet application program interface (API). JSP is comparable to Microsoft's Active Server Page (ASP) technology. Whereas a Java Server Page calls a Java program that is executed by the Web server, an Active Server Page contains a script that is interpreted by a script interpreter (such as VBScript or JScript) before the page is sent to the user.

4.4.2 ASP

An ASP is an HTML page that includes one or more script (small-embedded programs) that are processed on a Microsoft Web server before the page is sent to the user. An ASP is somewhat similar to a Server-side include or a common gateway interface (CGI) application in that all involve programs that run on the server, usually tailoring a page for the user. Typically, the script in the Web page at the server uses input received as the result of the user's request for the page to access data from a database and then builds or customizes the page on the fly before sending it to the requestor.

ASP is a feature of the Microsoft Internet Information Server (IIS), but, since the server-side script is just building a regular HTML page, it can be delivered to almost any browser. You can create an ASP file by including a script written in VBScript or JScript in an HTML file or by using ActiveX Data Objects (ADOs) program statements in the HTML file. You name the HTML file with the ".asp" file suffix. Microsoft recommends the use of the server-side ASP rather than a client-side script, where there is actually a choice, because the server-side script will result in an easily displayable HTML page. Client-side scripts (for example, with JavaScript) may not work as intended on older browsers.

4.4.3 VBScript

VBScript is an interpreted script language from Microsoft that is a subset of its Visual Basic programming language. VBScript can be compared to other script languages designed for the Web, including:

- Netscape's JavaScript
- Sun Microsystem's Tool Command Language
- The UNIX-derived Practical Extraction and Reporting Language
- IBM's Restructured Extended Executor

In general, script languages are easier and faster to code in than the more structured, compiled languages such as C and C++ and are ideal for smaller programs of limited capability or that can reuse and tie together existing compiled programs.

VBScript is Microsoft's answer to Netscape's JavaScript. Both are designed to work with an interpreter that comes with a Web browser - that is, at the user or client end of the Web client/server session. VBScript is designed for use with Microsoft's Internet Explorer browser together with other programming that can be run at the client, including ActiveX control, automation servers, and Java applet. Although Microsoft does support Netscape's JavaScript (it converts it into its own JScript), Netscape does not support VBScript. For this reason, VBScript is best used for intranet Web sites that use the Internet Explorer browser only.

4.4.4 Jakarta Tomcat

Jakarta is the "overall" project for many subprojects. For example, Tomcat is the Servlet + JSP Engine, which is a subproject of the Jakarta Project. Jakarta is essentially just a website and Tomcat is a product on that website.

Tomcat is the servlet container that is part of the JavaServer Web Development Kit (JSWDK) available from Sun. The source code for Tomcat is being provided to the Jakarta project for further development through the Open Source model.

When configured correctly APACHE will continue serving .html pages, while TOMCAT actually serves the .jsp pages and run the servlet that your site is containing.

Installation of Jakarta Tomcat

I followed the installation guide_[1] to install the Tomcat Server. I created a folder (hovedoppg) to save my files in. The BUILD.XML file (tomcat/conf/build.xml) had to be changed.

I added these lines in the build.xml file.

```
<mkdir dir="${tomcat.build}/webapps/hovedoppg"/>
<copydir src="src/hovedoppg" dest="${tomcat.build}/webapps/hovedoppg"/>
    <javac srcdir="src/hovedoppg/WEB-INF/classes"
destdir="${tomcat.build}/webapps/hovedoppg/WEB-INF/classes"
        classpath="${tomcat.build}/classes"/>
```

```
<jar jarfile="${tomcat.home}/webapps/hovedoppg.war"  
    basedir="${tomcat.home}/webapps/hovedoppg"  
    includes="**" />  
<deltree dir="${tomcat.home}/webapps/hovedoppg"/>
```

Finally we found that the JDBC driver had to be downloaded and installed in the *tomcat/lib/* directory on the server.

4.4.5 Drivers

The application uses a JDBC driver, but in the start I wanted to use a ODBC – JDBC bridge so that I could do the programming in ASP. But I soon found the disadvantages with this driver.

- The ODBC drivers are not thread-safe. This means the access to the database is serialised: a middle tier server cannot send request to the database from more than one client at a time.
- The bridge requires installation of code on the client.

4.4.6 System requirements

The You-too application uses the following software on the server side:

The databases:

- Postgresql database
- Access database

The script:

- JSP (Java Server Pages)
- ASP (Active Server Pages)
- Java

The server:

- Debian/GNU Linux (OS)
- Linux core 2.2.19
- Apache web server
- Jakarta Tomcat (Java Servlets and JSP)
- Postgresql database system

The client side:

The client needs some web browser, for instance,

- Internet Explorer 3.01 - à
- Netscape navigator 3.01 - à

4.5 Use case for present system

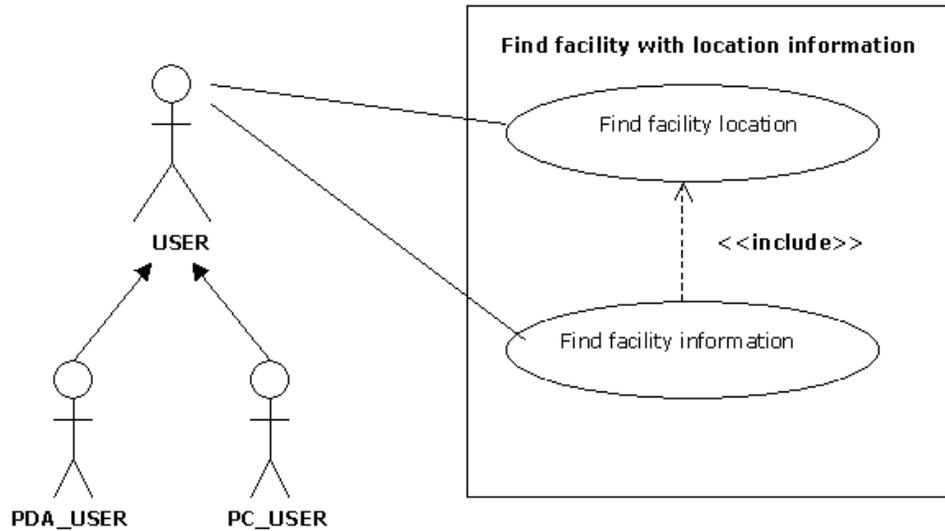


Figure 4 3 use case for present system.

Find facility location:

The user, whether it's the PDA_USER or PC_USER, has to send the first request to "Find facility location". The data transferred is radius from the users location, facility and GPS data. Then the response is the facilities within the users search data.

Find facility information:

The user sends a new request to "Find facility information". The data transferred is the selected facility ID where the response is information on the chosen facility.

4.6 Implementation of converting from meters to coordinates

The coordinates taken from the rastermap (.gif) are stored in a database, for the user to be able to search on distance the application has to convert the users input, which is meter, to coordinates.

To make the converting from a rastermap (.gif) the only solution is to measure the distance between two points on the picture (cm) and find how many coordinates goes in 1 cm. I did this only to show how this conversion could work in my prototype.

The use of rastermap does not give the opportunity to do this in any other way. A vectorised map keeps all information, for instance cm pr. Pixel. By using a vector map the solution would be accurate and easy to do. The Java class take care of this method.

4.7 Java Server Pages and Servlet design

During the design of this system and in examining similar web applications I found that the Model View-Controller (MVC) could be used_[2]. The Model-View-Controller (MVC) is a commonly used and powerful architecture for GUIs. There are now many flavours of MVC implementations. What they have in common is the separation of the model or business logic from the view logic in an application. In many cases the controller will manage the interaction between the model and the view. This is the model I used.

The model in my implementation was implemented using JavaBeans, which provided the database connection and the application methods. In using this flavour of MVC to build my application I found three basic questions:

1. Where to put the controller?

2. How to return the result to the client?
3. What goes into the JSP?

4.7.1 Where to put the controller?

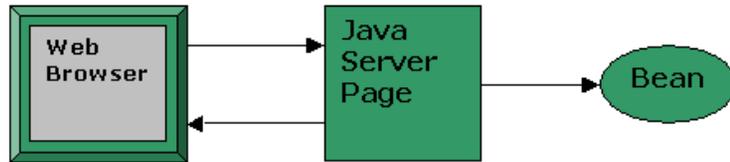


Figure 4 4 JSP as Controller and View

The controller could be a servlet or a JSP. Using a servlet to both the controller and view has some disadvantages. If the servlet is used to generate dynamic content, then any changes made to the format of the output also require that the servlet be recompiled. This makes the application more difficult to maintain. It's also quite difficult to create sophisticated web pages using HTML output from Servlets. The application uses JSP as the controller (Figure 4-2), which is best for small projects or prototype where the view is fairly simple.

4.7.2 How to return the result to the client?

The second design issue is the form in which results (the dynamic content created by interacting with the model) are returned to the client. This application uses beans, which interact with the JSP and sends back results from the methods. The queries are handled in the JSP because it's a small prototype and only a few SQL-queries. So the JSP return the results from the database and writes them out to the user with the `out.print()` command.

4.7.3 What goes into the JSP?

In my application I have Java code within the JSP, which interact with the beans, that handles the database connection. I will have all SQL-queries in the JSP; this means the JSP will contain both HTML and Java code. This could be untidy, but since it's a small prototype it will not cause problems.

4.8 Further work

This application provides an initial insight in how data from GIS can be used in other solutions than ordinary maps. To obtain more accurate and comprehensive data, further work is however required.

Furthermore:

- Given that the users for instance found a restaurant that they wants to visit, then they could be provided with that restaurants menu and prices. The restaurant administrates this information and we just keep a link to that information.
- In some portals on the web all users can save their profile, this is maybe more necessary in this application. All handicapped can describe their handicap in a profile. This has some security issues described in chapter 4.9 Security issues.
- A more advanced further solution is to provide the user with a map on the PDA or PC. This map puts all facilities within the search on the map and shows directions to all facilities from the location to the user.
- Map solutions over buildings. The user can take advantage of the network-based location [2.6.3] and the mobile terminal can get its GPS location. The user can navigate in the building using the mobile terminal and get information about the different facilities in the building.

Applications like this “never” finish. There is always some new functions or other extension to applications like this.

4.8.1 Use case for extended system

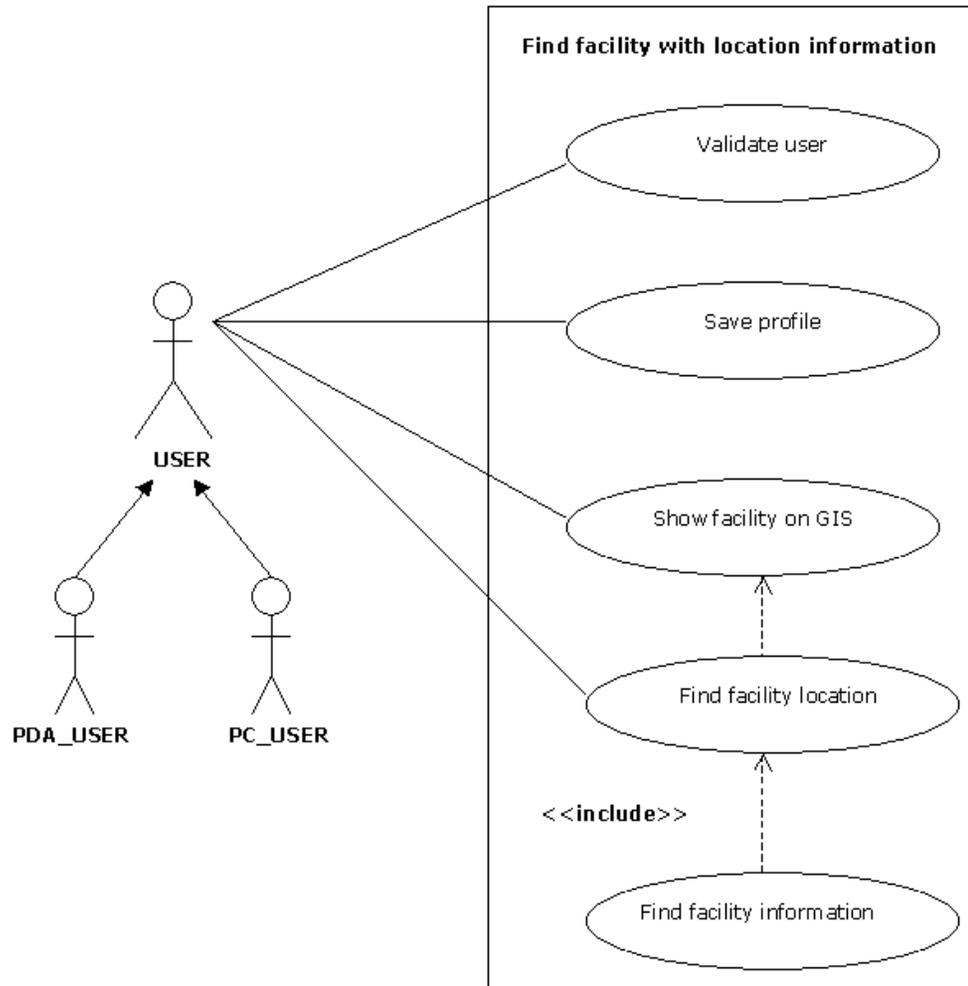


Figure 4 5 use case for extended system.

Validate user:

Checks if the user already is registered. The user sends the cookie or if it is password-protected, username and password is send.

Save profile:

If not validated the users can register their profile.

Find facility location:

The user, whether it's the PDA_USER or PC_USER, has to send the first request to "Find facility location". The data transferred is radius from the users location, facility and GPS data. Then the response is the facilities within the users search data.

Find facility information:

The user sends a new request to "Find facility information". The data transferred is the selected facility ID where the response is information on the chosen facility.

Show facility on GIS:

When the user has sent the request, the response is provided to the user in a GIS. The details for each facility in the GIS are provided by "Find facility location".

4.9 Security issues

When a user registers personal data this information should not be available to anyone else, to do this we can use for instance sessions and store the data in a database or we can make a cookie. Both this techniques have advantages and disadvantages.

Sessions advantages:

- Convenient and easy to use
- Cache data across requests. Rather than redoing the same expensive calculation or making the same expensive query from a database for every request, calculate the data once and store the data in the Session object. Often a tremendous performance boost.
- No need to build custom infrastructure. You can concentrate on building your application instead.
- Often good enough for applications that don't require special functionality or massive storage capacity.

Sessions disadvantages:

- Session state is fragile.
- Session state is not persistent.
- Sessions last a long time, if a user visits only one or two sites the session persist in the memory for a long time. On a busy site that can add up to a lot of memory.

Cookie advantages:

- Easy to use.
- The user only gets what's relevant when registered.

Cookie disadvantages:

- The browser must support cookies and cookies must be enabled on the browser.
- Cookies have size limitations: a browser need store no more than 300 total cookies, with 20 cookies per server, and no more than 4KB per cookie.
- It's insecure as the session state is stored on the users machine and travels back and forth across the network with every request.

Other technique for ensuring privacy communications is Cryptography.

This method is mostly used so that hackers can't get information while data is send over the Internet. Cryptography is typical used when e.g. credit card number is required from a website and send over the Internet. For more extensive source for Cryptography available on the net, go to the RSA security Pages₀.

4.10 User interface

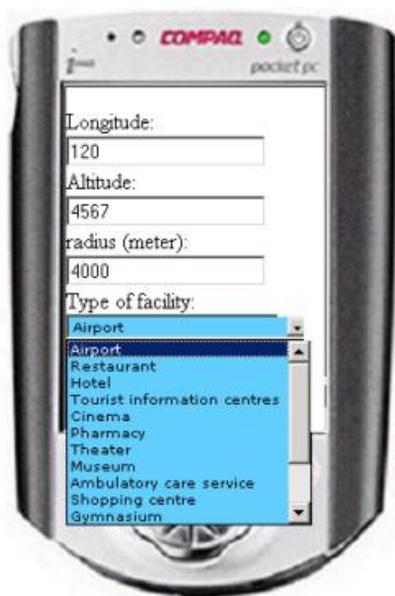


Figure 4 6 the opening interface. User can decide distance to the facility he/she chooses.

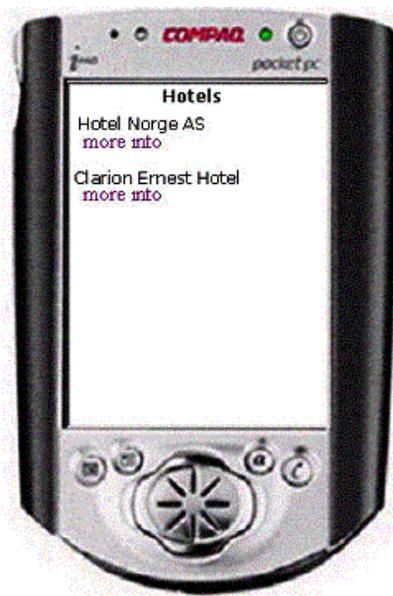


Figure 4 7 User gets the facilities of current interest. And can navigate further for more information



Figure 4 8 the user chose to view more information from Clarion Ernest Hotel. User then gets all available information. He/she can now choose to navigate for accessibility information.



Figure 4 9 User can decide where he/she wants to view accessibility information from.



Figure 4 10 shows all accessibility information available for this facility

5 Results

Raster- and vector maps can look the same on screen, but that's all. The quality on vector maps and the prospects are much better. When it comes to functioning, vector maps have much more functions; you can save different things in different layers. For instance roads can be stored in one layer and facilities can be stored in another layer. There are no restrictions on how many layers you can make. This gives the user the chance to search on road directions, distance between to points and so forth. The restrictions with vector based maps are few but important. Vector maps do not exist for all areas; only the biggest cities are available. This is because they are expensive to create.

Raster maps used for showing facilities and roads are used in a wide range, but as soon as you want to expand the application and add some functionality like road directions and distance, vector map is needed.

Most of the GIS applications I investigated had all the important qualities [1.5], but they were too expensive. We found one solution that might fit the you-too consortium; the MapServer which is open source. Since this program is more difficult to install and use than other GIS solutions, they will need a person with experience or the ability to get up the details from this application.

GIS solutions from ESRI and MapInfo already support PDA interface and have applications that users already can use. But because of the new technology the prices are too high so these products are not current interest.

6 Discussion

To store the location coordinates I had to look in to several different technologies to find the most usable solution. The Postgresql database supports the *point datatype* and has several functions to support this datatype. Since I had access to this database and some persons with good competence in that area I chose Postgresql to store the location information. This database only supports JDBC connection, the administrator could have installed an ODBC driver on the Postgresql server but this was not done. Since the connection to the database uses JDBC connection I chose to make a Java class that connected to the database, and since the prototype should be demonstrated on the web I had to use JSP to connect to the Java class. Because of the time limitation and previous experience in ASP and VBScript I used these technologies where it was possible.

The best solution for this application would be to use JSP and servlet from scratch to the end. This is because of easier maintenance for the administrator.

The GIS solution that I think fit the you-too consortium is not the best GIS solution on the web. This is based on that it is not a full-featured GIS system. But it has the qualities needed for the you-too consortium. The cost of other systems is why I wouldn't recommend those systems for you-too, even if some of the systems were easier to maintain and use.

7 Conclusion

In this master Thesis I have presented GIS solutions on the web. I have presented a GIS that I think fits the you-too project in chapter 5 Results. The main reason for this choice is that it is open source, and thereby free of charge.

The foundation for an integration of GIS on PDA or any mobile terminal is at present. Constant improvement on efficiency and capacity in hardware have led to applications like GIS can be used on almost any PDA and also some mobile phones. For this reason this technology is of very great present interest.

Two of the presented GIS solutions support mobile terminals; a user can for instance use a PDA to find the nearest facility of interest. This solutions also provide more information than my employer needs for the

you-too project and the problem with these solutions is the costs, the you-too consortium do not have the money for solutions like this. For this reason I have made a prototype that takes advantage of the stored location data.

A GIS like the one I have proposed have the ability to store information such as location in a database. The prototype illustrates how this information can be used to take advantage of this data, and provide the user with topical data. The prototype simulates how location data from a GPS can be used to select the facilities within a radius decided by the user. In chapter 4.8 Further work, I have made some suggestions for how to obtain more accurate and comprehensive data. These suggestions should be applied to the application before use.

8 References

- [1] Computerworld NR.80 November 2000, Information on Location based services.
- [2] Java Report December 1999, volume 4, Number 12.
- [3] Report from earlier you-too project by Kim Andersen & Nils Li Pedersen, September 2000
- [4] G.Booch, J.Rumbaugh, I.Jacobson, The Unified Modeling Language User Guide pp 219 – 241, Addison-Wesley, Reading Massachusetts, 1999.
- [5] Installation guide of Jakarta Tomcat server [online], last visited april 2001, web address:
<http://jakarta.apache.org/>
- [6] USNO GPS Timing Operations [online], last visited April 2001, web address:
<http://tycho.usno.navy.mil/gps.html>
- [7] Products from ESRI, GIS & mapping software [online], last visited May 2001, web address:
<http://www.esri.com/>
- [8] Products from Manifold, GIS & mapping [online], last visited April 2001, web address:
<http://www.manifold.net/>
- [9] Products from Mapinfo [online], last visited May 2001, web address:
<http://www.mapinfo.com/>
- [10] Products from Mapmaker [online], last visited May 2001, web address:
<http://www.mapmaker.com/>
- [11] Products from Mappoint [online], last visited May 2001, web address:
<http://www.microsoft.com/office/mappoint/default.htm>
- [12] Information on MapServer [online], last visited May 2001, web address:
<http://gis.about.com/science/gis/gi/dynamic/offsite.htm?site=http%3A%2F%2Fmapserver.gis.umn.edu%2Findex.html>
- [13] Information on Caliper products [online], last visited May2001, web address:
<http://www.caliper.com>
- [14] Information on postgresql server [online], last visited May 2001, web address:
<http://www.postgresql.org>
- [15] Source for Cryptography available on the net [online], last visited May2001, web address:
<http://www.rsa.com/>

[16] A global network of geographic information users and providers. [online], last visited May 2001, web address:

<http://www.geographynetwork.com/>

[17] GIS dictionary on the web [online], last visited May2001, web address:

<http://www.cadforum.com/>

[18] General information on GIS [online], last visited April 2001, web address:

<http://www.gis.com/>

Appendix

OLE DB

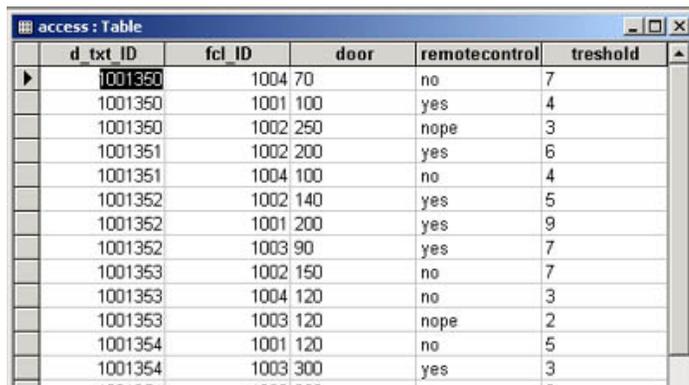
OLE DB is the foundation for Microsoft's current data access architecture. OLE DB provides greater flexibility over ODBC in that it allows access to non-relational data in addition to data stored in relational databases such as DB2. It presents an object-oriented interface for generic data access. OLE DB is based on the Microsoft Component Object Model (COM) architecture for application components. To implement an application using OLE DB, you typically use ADO methods and properties to call the OLE DB interface. ADO provides a high-level interface that can easily be used from within C++, Visual Basic, VBScript and so on.

ODBC

ODBC (Open Database Connectivity) is a commonly accepted Application Programming Interface (API) for database access from Windows. It is based on the specifications from X/Open and ISO/IEC for database access and uses SQL as its database access language. Although Microsoft and others have developed higher-level interfaces over ODBC, it is fundamentally a low-level, procedural, state-managed interface.

ODBC middleware is available for all commonly used relational databases, and hundreds of third-party products are designed to work with ODBC. To use an ODBC driver, you first define a data source (a set of parameters to connect to a specific DB2 database using a specific network connection). Your application can then connect to a DB2 server using the ODBC Manager, which calls a database-specific ODBC driver in the Windows environment.

Access database structure



d_txt_ID	fcl_ID	door	remotecontrol	threshold
1001350	1004	70	no	7
1001350	1001	100	yes	4
1001350	1002	250	nope	3
1001351	1002	200	yes	6
1001351	1004	100	no	4
1001352	1002	140	yes	5
1001352	1001	200	yes	9
1001352	1003	90	yes	7
1001353	1002	150	no	7
1001353	1004	120	no	3
1001353	1003	120	nope	2
1001354	1001	120	no	5
1001354	1003	300	yes	3

Figure A 0 1 the access table. Contains different accessibility information in each facility.

Figure A-Error! No text of specified style in document.-1 the detailinfo table. Contains

detailinfo : Table	
facility ID	dt_txt ID
1001350	1001
1001350	1002
1001350	1004
1001351	1002
1001351	1004
1001352	1001
1001352	1002
1001352	1003
1001353	1002
1001353	1003

detailinfo_txt : Table	
detailinfo ID	txt
1001	General Info
1002	Reception
1003	Dining room
1004	Room
0	

Figure A 0 3 detailinfo_txt table. Contains the different rooms in each facility with text.

FCL_ID	FCL_NewDate	FCL_LastDate	FCL_Publishe	FCL_NewPubl	FCL_Name	FCL_Type ID	FCL_Publishe	FCL_Status	FCL Parent
1001360	2000 08:18:49	2000 14:23:37	1001063	-1	Clarion Ernst H	26	0	0	
1001362	2000 13:25:08	2000 14:32:02	1001063	-1	Kristiansand lut	13	0	0	
1001350	2000 23:20:47	2000 13:43:02	1001063	-1	Hotel Norge	26	0	0	
1001367	2000 13:54:27	2000 13:32:38	1001063	-1	Serlandshallen	1000062	0	0	
1001354	2000 11:35:55	2000 13:23:14	1001063	-1	Mc Donalds	34	0	0	
1001359	2000 12:43:44	2000 14:02:23	1001063	-1	Serlandets Kun	52	0	0	
1001351	2000 21:52:06	2000 14:48:47	1001063	-1	Tourist Informat	81	0	0	
1001361	2000 08:20:52	2000 13:41:51	1001063	-1	Glipp	1000055	0	0	
1001357	2000 15:45:42	2000 14:10:14	1001063	-1	Vaskeriet, stud	46	0	0	
1001368	2000 09:32:07	2000 14:32:40	1001063	-1	Quality Hotel	26	0	0	
1001366	2000 10:39:21	2000 13:56:00	1001063	-1	Sjehuset	33	0	0	
1001363	2000 13:27:18	2000 14:30:54	1001063	-1	Kristiansand Kc	1000089	0	0	
1001356	2000 09:21:26	2000 13:38:12	1001063	-1	Aader Tester	49	0	0	

Figure A 0 4 the facility table. Contains the different facilities with address, phone number etc

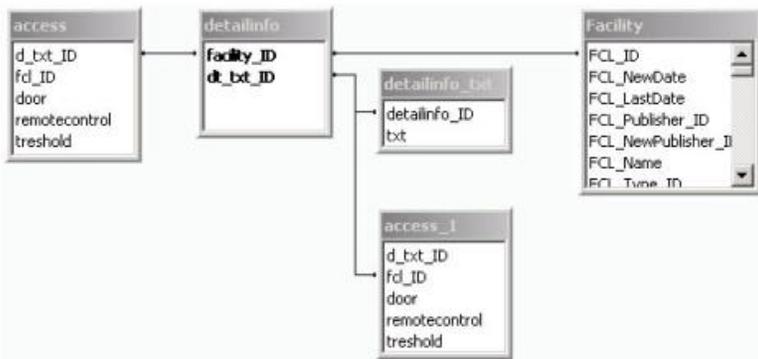


Figure A 0 5 Shows the relationship between the tables I have used in my prototype

The original task description

Funksjonshemmede kan i stor grad øke sin mobilitet gjennom tilforlætelig informasjon om tilgjengelighet til steder de ønsker å reise. Et EU prosjekt har utviklet et rammeverk for innsamling og organisering av slik informasjon. Jeg skal se nærmere på slik informasjon i en kartløsning. Informasjonen kan hentes ut via et digitalt kart. Oppgaven går ut på å lage en oversikt over løsninger som kopler kart og informasjon om tilgjengelighet, implementere en løsning for dette, og foreslå innhold i et grensesnitt.

Innledningsvis vil en oversikt over kartløsninger som finnes på Internett i dag sammenstilles. Oversikten vil beskrive muligheter og begrensninger de forskjellige kartløsninger gir, spesielt i henhold til informasjon om tilgjengelighet.

Fra et digitalt kart i en web applikasjon er det ønskelig å hente inn informasjon fra flere databaser. Det kan være databaser som ligger på forskjellige servere. Oppgaven går her ut på å definere og prøve ut en strukturert måte å gjøre dette på.

For å implementere en løsning, vil Publikt sin web applikasjon videreutvikles slik at metadata kan hentes ut fra databaser fra forskjellige lokasjoner. Applikasjonen skal lagre aktuelle data i database istedenfor på fil, dette for lettere å kunne søke på koordinater ved en mobil løsning. Metadata fra en fjerntliggende database skal også hentes inn i kartløsningen, dette gjøres ved søk på koordinater og/eller navn.

Det skal lages en mal for Kristiansand som kan utvikles videre for andre byer.

Videre vil muligheter for å koble kartløsningen mot mobile systemer som GPRS, mobiltelefon, og bruk av GPS skisseres.

Om tiden strekker til, vil også et forslag til innspill til en ISO standard for et grensesnitt utvikles.

CODE

----- location.jsp -----

```
<%//@ page import="no.hia.ikt.dogma.DBConnection.*" %>
```

```
<HTML>
```

```
<BODY>
```

```
<FORM ACTION="location3.jsp? METHOD="POST">
```

Longitude:

```
<INPUT TYPE="TEXT" NAME="longitude" value="120"><BR>
```

Altitude:

```
<INPUT TYPE="TEXT" NAME="altitude" value="4567"><BR>
```

radius (meter):

```
<INPUT TYPE="TEXT" NAME="radius" value="4000"><BR>
```

Type of facility:

```
<SELECT name=typesok style="font-family:VERDANA, ARIAL, HELVETICA, GENEVA; font-size:10px; BACKGROUND-COLOR: #66ccff">
```

```
    <OPTION value="13">Airport</option>
```

```
    <OPTION value="33">Restaurant</option>
```

```
    <OPTION value="26">Hotel</option>
```

Geographic Information Systems - GIS software

<OPTION value="81">Tourist information centres</option>

<OPTION value="48">Cinema</option>

<OPTION value="171">Pharmacy</option>

<OPTION value="49">Theater</option>

<OPTION value="52">Museum</option>

<OPTION value="1000089">Ambulatory care service</option>

<OPTION value="1000080">Shopping centre</option>

<OPTION value="1000062">Gymnasium</option>

<OPTION value="12">Train station</option>

<OPTION value="34">Fast food</option>

<OPTION value="46">Event centre</option>

<OPTION value="1000055">Bistro</option>

</SELECT>

<INPUT type="submit" value="button">

<HR>

```
<%/longitude=120&altitude=250&radius=1200"
```

```
    ResultSetVectors vec = null;
```

```
    DBConnection con = new DBConnection();
```

```
    String test;
```

```
    //test = hmm();
```

```
    //out.println(test);
```

```
    if (con.connect("location", "kaande99", "1515")) {
```

```
        vec = con.executeQuery("select * from facilities3 where point_distance(location,'250,120') < 200");
```

```
        if (vec.getRowCount() > 0)
```

```
        {
```

```
            out.print(DefaultShowTable.defaultModel(vec));
```

```
        }
```

```
    } else
```

```
    {
```

```
        out.println("Could not connect to the database, please try again later!");
    }
    */
%>

</BODY>
</HTML>
```

----- Location3.jsp -----

```
<% @ page import="no.hia.ikt.dogma.DBConnection.*" %>
```

```
<HTML>
```

```
<BODY>
```

```
<%
```

```
    int numberOfColumns = 0;
```

```
    String longitude = request.getParameter("longitude");
```

```
    String altitude = request.getParameter("altitude");
```

```
    String radius = request.getParameter("radius"); //må regnes om fra meter til pixler
```

```
    String intype = request.getParameter("typesok");
```

```
    ResultSetVectors vec = null;
```

```
    DBConnection con = new DBConnection();
```

```
    int px_radius = con.get_radius(radius);
```

```
    out.print("long:"+longitude);
```

```
    out.print("alt:"+altitude);
```

```
    out.print("radius:"+px_radius);
```

```
    //test = hmm();
```

```
    //out.println(test);
```

```
    if (con.connect("location", "kaande99", "1515")) {
```

```
        vec = con.executeQuery("select id, name from facilities where point_distance(location,'" + altitude + "','" + longitude + "') <= " + px_radius + "and type="+intype);
```

```
        numberOfColumns = vec.getRowCount();
```

Geographic Information Systems - GIS software

```
    if (numberOfColumns > 0)
    {
        for (int i = 0; i < numberOfColumns; i++)
        {
            out.print("<br>" + vec.getRow(i) + "<br>");
            out.print("<a href='http://128.39.202.254/hovedoppg/info.asp?id="+vec.getRow(i)+">more info</a>");
        }
        //out.print(DefaultShowTable.defaultModel(vec));
    }
    //out.print("<a href='location.jsp'>more info</a>");

} else
{
    out.println("Could not connect to the database, please try again later!");
}

%>

</BODY>
</HTML>
```

----- info.asp -----

<%

Function SearchAndReplace(s, Str1, Str2)

Dim pos

s = s & "" 'Fix Null

pos = InStr(s, Str1)

While pos > 0

 s = Mid(s, 1, pos - 1) & Str2 & Mid(s, pos + Len(Str1))

 pos = InStr(pos, s, Str1)

Wend

SearchAndReplace = s

End Function

'Dim MyString, LeftString

```
'MyString = "VBScript"
```

```
'LeftString = Left(MyString, 8) ' LeftString contains "VBS".
```

```
'-----
```

```
inid = Request.QueryString("id")
```

```
Dim MyString
```

```
MyString = inid
```

```
inid = Left(MyString, 8)
```

```
inid = SearchAndReplace(inid, "[", "")
```

```
'LeftString = SearchAndReplace(inid, "]", "")
```

```
'tester det over!
```

```
'inid = Request.QueryString("id")
```

```
'inid = 1001352
```

```
Course_ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & server.mappath("database/nybidb.mdb")
```

```
set conn = Server.CreateObject("ADODB.Connection")
```

```
conn.open Course_ConnectionString
```

```
strsql = "Select * From Facility where FCL_ID ="&inid
```

```
'Response.Write strsql
```

```
set rs = conn.Execute (strsql)
```

```
%>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE></TITLE>
```

```
</HEAD>
```

```
<BODY topmargin="0" leftmargin="0">
```

```
<table align=center>
```

Geographic Information Systems - GIS software

```
<%
do while not rs.eof
%>
<tr><a href="moreinfo.asp?id=<%=inid%>"><%=rs("FCL_Name")%></a></tr>
<tr><td><b>a</b></td>&nbsp;<td><%=rs("FCL_Street")%>&nbsp;<%=rs("FCL_Number")%></td></tr>
<tr><td></td><td><%=rs("FCL_Zipcode")%>&nbsp;<%=rs("FCL_City")%></td></tr>
<tr><td><b>a:</b></td>&nbsp;<td><%=rs("FCL_Phone")%></td></tr>
<%if rs("FCL_Fax") <> " " then%>
<tr><td><b>a:</b></td>&nbsp;<td><%=rs("FCL_Fax")%></td></tr>
<%end if%>
<%if rs("FCL_Email") <> " " then%>
<tr><td><b>a:</b></td>&nbsp;<td><a href=mailto:<%=rs("FCL_Email")%>><%=rs("FCL_Email")%></a></td></tr>
<%end if%>
<%if rs("FCL_Website") <> " " then%>
<tr><td><b>a:</b></td>&nbsp;<td><a href="http://<%=rs("FCL_Website")%>" target=_blank><%=rs("FCL_Website")%></a></td></tr>
<%end if%>

<%
rs.movenext
loop
%>
```

</table>

</BODY>

</HTML>

----- moreinfo.asp -----

```
<%
```

```
inid = Request.QueryString("id")
```

```
'responsee.write inid
```

```
Session("Course_ConnectionString") = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & server.mappath("database/nybidb.mdb")
```

```
set conn = Server.CreateObject("ADODB.Connection")
```

Geographic Information Systems - GIS software

```
conn.open Session("Course_ConnectionString")
```

```
strsql = "Select dt.detailinfo_ID, dt.txt From detailinfo_txt dt, detailinfo di where di.dt_txt_ID = dt.detailinfo_ID"
```

```
strsql = strsql & " and di.facility_ID ="&inid
```

```
'Response.Write strsql
```

```
set rs = conn.Execute (strsql)
```

```
%>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE></TITLE>
```

```
</HEAD>
```

```
<BODY topmargin="0" leftmargin="0">
```

```
<table align=center>
```

```
<%
```

```
do while not rs.eof
```

```
%>
```

```
<a href="detailinfo.asp?id=<%=rs("detailinfo_ID")%>&fcl=<%=inid%>&txt=<%=rs("txt")%>"><%=rs("txt")%></a><br>
```

```
<%
```

```
rs.movenext
```

```
loop
```

```
%>
```

```
</table>
```

```
</BODY>
```

```
</HTML>
```

```
----- detailinfo.asp -----
```

```
<%
```

```
infcl_id = Request.QueryString("fcl")
```

```
inid = Request.QueryString("id")
```

```
'response.write inid
```

```
Session("Course_ConnectionString") = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & server.mappath("database/nybidb.mdb")
```

```
set conn = Server.CreateObject("ADODB.Connection")
```

```
conn.open Session("Course_ConnectionString")
```

```
strsql = "Select * From access where d_txt_ID = "&infc1_id
```

```
'strsql = strsql & " and di.dt_txt_ID = a.d_txt_ID"
```

```
strsql = strsql & " and fcl_ID = "&inid
```

```
'Response.Write strsql
```

```
set rs = conn.Execute (strsql)
```

```
%>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE></TITLE>
```

```
</HEAD>
```

```
<BODY topmargin="0" leftmargin="0">
```

```
<h3 align=center><%response.write (Request.QueryString("txt"))%></h3>
```

```
<table align=center width=190>
```

```
<%
```

```
do while not rs.eof
```

```
%>
```

```
<tr><td>Door opening:&nbsp;&nbsp;&nbsp;</td><td><%=rs("door")%></td></tr>
```

```
<tr><td>TV remote control:&nbsp;&nbsp;&nbsp;</td><td><%=rs("remotecontrol")%></td></tr>
```

```
<tr><td>Treshold:&nbsp;&nbsp;&nbsp;</td><td><%=rs("treshold")%></td></tr>
```

```
<%
```

```
rs.movenext
```

```
loop
```

```
%>
```

```
</table>
```

```
</BODY>
```

</HTML>

----- DBConnection.java -----

```
package no.hia.ikt.dogma.DBConnection;
```

```
import java.sql.*;
```

```
import java.net.*;
```

```
import java.lang.*;
```

```
import java.util.*;
```

```
import org.postgresql.Driver;
```

```
public class DBConnection {
```

```
    private    Connection con;
```

```
    private String database = null;
```

```
    private String login = null;
```

```
    private String password = null;
```

```
    public DBConnection() {
```

```
    }
```

```
    public boolean connect(String db, String name, String pwd) {
```

```
        database = "jdbc:postgresql://gullfisk.agder-ikt.hia.no/" + db;
```

```
        login = name;
```

```
        password = pwd;
```

```
        boolean connected = true;
```

```
        try {
```

```
            Class.forName("org.postgresql.Driver");
```

```
        } catch (ClassNotFoundException e) {
```

Geographic Information Systems - GIS software

```
        System.out.println("Error in Class.forName()");
    }
    connected = false;
}
try {
    con = DriverManager.getConnection(database, login, password);
} catch (SQLException e) {
    System.out.println("Error in DriverManager.getConnection()");
    connected = false;
}
return connected;
}
```

```
public void disconnect() {
    try {
        con.close();
    } catch (SQLException e) {
        System.out.println("Error closing connection");
    }
}
```

```
public ResultSetVectors executeQuery(String query) {
    ResultSetVectors result = null;
    try {
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        result = new ResultSetVectors(rs);
    } catch (SQLException e) {
        System.out.println("Error in con/stmt");
    }
    return result;
}
```

```
public boolean executeUpdate(String query) {
    boolean result = false;
    try {
        Statement stmt = con.createStatement();
```

Geographic Information Systems - GIS software

```
        stmt.executeUpdate(query);
        result = true;
    } catch (SQLException e) {
        System.out.println("Error in DBConnection.execteUpdate()");
    }
    return result;
}

public double get_radius(String rad) {
    double radius = (double) rad;
    double px_radius = (radius/5.6); //5,6 er en faktor regnet ut fra Rasterkart(GIF fil)
    return px_radius;
}

public static void main (String[] args)
{
    DBConnection con = new DBConnection();
    ResultSetVectors vec = null;
}

}
```