Ship Data Extraction and Search

Dinesh Kumar Subramanian
Department of Computer Science
University of Illinois at Chicago

Contents

1. Introduction

For centuries ships have been the main source of transportation for raw materials between countries. The economy of a country is determined by its export and import. Shipping has been the main source of export and import. So organization like Lloyds, Sailwx, AIS, VOA started to collect data regarding ships, ship movements, Position etc and are selling them through subscriptions.

This project is a search engine that determines specific details of the ship like its present position, the source and destination of the ship, the type of the ship (whether it is a cruise or cargo ship), the route the ship has taken and at the same time the ports the ship has visited and will visit in future . The project is implemented using PHP, Python, AJAX and MYSQL. High emphasis is given to the usability and the accessibility of the search engine making it interesting and interactive for the user. The search engine presents only those information that are of most importance rather than populating the UI with lot of options and data.

2. Objective

The main object of this project is to create a web based search engine that collects all the data in the web pertaining to a particular ship and port. The data collected is mashed up with Google maps API to facilitate interactivity and ease in understanding.

3. State of Art

Here the state-of-art section analyzes the entities that provide major shipping data and presents what each of these entities provide, how the data is provided, sources of data for these entities and also a proper analysis highlighting the pros and cons of the data these entities provide.

The following are some of the entities that provide shipping information online

3.1. Lloyds

http://www.lloydsmiu.com/lmiu/index.htm

With a history dating back nearly 300 years ago to the ship movements first published in Lloyd’s List newspaper, Lloyd's MIU is the world's acknowledged leader in maritime information and research.

Loylds MIU provides the following

- Detailed characteristics of 120,000 vessels
- Over 200 million weekly AIS position reports <<what is AIS? Not mentioned yet>>
- Over 21,000 casualty messages passed to clients in 2006
- Over 163,000 shipping companies
- Over 13,500 credit reports
- Extensive details on 2,800 of the world's commercially active ports

Pros and Cons

Lloyds provide data using AIS. This is complemented by the fact that they also obtain schedule information directly from various ports around the world. This facilitates up to date information.

The downside is that the data collected by Lloyds is available online only on subscription for a very high fee. Also the data is provided in the form of a regular table layout. This makes it difficult for the user to understand as compared to representing it on a map.

3.2. AIS Live

http://www.aislive.com/index.aspx

Established in 2004, AISLive now covers over 1,200 places worldwide and is the most cost effective method of tracking vessels in real time available today. With more than 13,000 vessels under coverage at any given time and state of the art viewing software, tracking your fleet has never been so easy.

AIS data is received from a global network of receivers on a continuous basis 24 hours a day. AISLive has the ability to stream live data for a specific area in the world right down to the level of a berth for use in a clients own application.

How it works

AISlive takes an hourly 'snapshot' of the positions of all vessels within their network. They then store these positions and match them against the Lloyd's Register of Ships vessel database to check the accuracy of the reported data. Thousands of automatic zones have been created all over the world which dynamically report when a vessel has been there and also advises how long the vessel spent in that zone.

What AISLive supplies

1. Live Data
2. Lloyd's Register Fairplay Data
3. Data by iFrame

AISlive data is comprehensive and includes:-

- Vessels - Over 150,000 vessels << but only 13,000 is mentioned in the last page>>
- Company data - Over 200,000 maritime related companies
- Ports - 9400 ports and terminals including services at each facility
- Fixtures - Wet and dry global fixtures

Pros and cons

The advantage of this service is that it provides information in a more presentable form on Google earth It collects data from AIS and also cross checks it with Lloyds data to make sure things are accurate

The real disadvantage of AISLive is that it provides data only on subscription. Providing data on Google earth is of no use (not on the Web) if it has to be displayed on a webpage <<What does that mean??>>.

3.3. World Shipping Register [e-Ships]

http://e-ships.net/

World Shipping Register is a leading Company that provides ships and shipping company data with powerful search criteria; E-ships data is comprehensive and includes

- More than 100,000 ships <<Do you know the exact number?>>
- More than 60,000 shipping companies
- More than 5,500 shipbuilders
- Cross links between the ships, shipowners and shipbuilders databases.

World Shipping Register has an excellent search functions which allow to find:

1. Ship or group of ships by:

   - Ship Name
   - Ship Ex-Name
   - IMO Nr.
   - Call Sign
   - Ship Type
   - Flag
   - Class
   - Deadweight
   - Displacement
   - Gross Tonnage
   - TEU Capacity
   - Length
   - Beam
   - Draft
   - Age of ship
   - Engine power (kW/HP)
   - Owner

- Ship manager
- Fleet manager

2. Company or group of companies by:

- Company Name
- Business
- Nationality
- Country of residence
- Port (City)
- Year of Foundation

World Shipping Register main preferences:

- Ship or groups of ships selection using multitude ‘From-To’ queries on deadweight, displacement, gross tonnage, teu capacity, length, beam, draft, age of ship, engine power, etc.
- On-line access to the latest details of word fleet information , print and easy data edit facility
- On-line access to shipowners, managers, operators, shipbrokers and shipbuilders details
- Access to data is available at any Internet-connected PC, no special requirements to PCs or OS

Pros and cons

e-ships provide information about both ports and ships around the world. The advantage is that it gives several options to search through the database and thereby able to get the specifics that we are interested in.

The disadvantage is that it’s a small repository and does not provide the ship movement information.

3.4. World Port Source

http://www.worldportsource.com/

World Port Source provides interactive satellite images, maps and contact information for 2,438 ports in 185 countries around the world. It helps quickly find any port using the regional map of the world.

The goal of World Port Source is to be the premier Internet website of publicly accessible seaport information.

The first objective of this site is to provide contact information and satellite images of ports and harbors throughout the world. Over a period of time, this foundation of world ports will be cross referenced with the people and companies who make their livelihood servicing the world's largest and most valuable transportation network.

Pros and Cons

This is a very good site if the user is just looking for the information about the ports around the world. The most advantageous factor is that the whole port database can be downloaded as an xml. It also presents the information on a map and allows the user to search through the ports specific to a country.

The disadvantage is that it does not provide other information such as ship details and ship movements. Further the site does not provide much user interactivity and search option to find the specifics of the data.

3.5. Maritime Global Net

http://www.mgn.com/worldports/ports.cfm

Since September of 1995, Maritime Global Net (MGN) has been an open resource to maritime professionals looking for news, industry sites, market information, and general contact details. Traffic has grown over the years and now averages over 200,000 user visits per month.

MGN offers access to products, services, news and other industry related data. MGN includes the details of over 80,000 maritime-related companies and contacts, from industry associations to world ports.

The MGN site provides Internet users with easy, unrestricted access to maritime information, products, and services.

Pros and Cons

The advantage of MGN is that they provide data that they have in their repository that includes port info and ship info for free. The database can be searchable for ships with their call sign, IMO number and searchable for ports with their name or by country.

But the downside is that their repository contains very small amount of data and the ships details with route information is not available also the data is presented in a mere table format.

3.6. Sailwx

http://www.sailwx.info/

SAilwx primarily use data reported via the World Meteorological Organization's Voluntary Observing Ship (VOS) program to provide a snapshot of current weather conditions at sea, worldwide. This data can also be used to track the progress of ships at sea. Many ships do not report their weather observations to WMO, or report only sporadically; these ships will not have records in Sailwx database. Additional information comes from the YOTREPS network of cruising yachts; YOTREPS positions are updated only once per day.

Quality control evaluations for VOS ships are available at http://www.meteo.shom.fr/vos-monitoring/

Additional weather data comes from the NOAA Forecast System Laboratory's MADIS database.

Tide predictions are provided by a heavily modified version of David Flater's program Xtide.

Maps are produced using the University of Minnesota Mapserver, with datasets derived from the NIMA VMAP-0 layers, popularly known as the Digital Chart of the World. Maps are written in PHP/Mapscript.

Hurricane data comes from NRL Monterey

Automatic Identification System (AIS) is a fairly new tool that dramatically changes the way we can track ships. An AIS transponder uses VHF frequencies, and broadcasts your own vessel's position, name, callsign, along with detailed parameters like length, beam, draft, and tonnage. It also broadcasts details of the current navigation system: speed, course, rate of turn, destination, and ETA. The transponder receives this same information from other ships, and either displays it on its own screen or emits it in an NMEA-standard data stream for use by chartplotters and other onboard nav gear. The positions and intentions of nearby vessels are available to you unambiguously and in real time.

Pros and cons

The advantage of this is that it provides almost all the information regarding all kinds of vessels ranging from research vessels , marine vessels, cruise and yachts further it also provides info about the weather tide, hurricane, temperature info.

And to further add to the point it provides the ship tracking i.e. route plotting on a map.

The main disadvantage is that it does not provide easy navigation and plotting of ship route and it also slow in processing the data. The map used to plot does not reveal a clear route of the vessel. It is not user friendly and interactive to the user.

3.7. Sea-web

http://www.sea-web.com/handler.aspx?control=seaweb_welcome

Sea-web provides you with online access to Lloyd's Register of Ships, combining comprehensive ships, owners, shipbuilders, fixtures, casualties and real-time vessel movements' data into a single application. With a powerful search facility and the ability to export data, Sea-web is the ultimate maritime reference tool.

- Details of more than 160,000 ships of 100 gross tons and above, including new buildings and casualties
- Up to 500 data fields, including tonnages, class, inspections, detentions, cargo, capacities, gear and machinery details
- More than 200,000 company records, representing 5 levels of group and operational ownership
- Extensive image library, with more than 85,000 ship photographs
- Complete shipbuilder information with comprehensive fleet listings

- Movements (real-time and historic), Fixtures and Casualty modules available
- Direct link into Equasis database and Fairplay news archive

The database contains details of millions of real-time and historic ship movements, and for each movement records the port of call, country, arrival date and sailing date. The information in the Movements Database is obtained primarily from AIS data and is supplemented by a number of other sources. Due to the nature of AIS coverage the movements do not represent all the port callings made by a vessel and not all ports are covered by the movement service.

Pros and Cons

The sea web's database pros and cons is no different from Lloyds as it provides only the information from Lloyds fair play, except it integrates a powerful search facility and ability to export data.

And finally to sum up the state of art the following are some of the other sites that provide information regarding shipping Ports, ship movements, temperature and weather at the sea etc.

http://www.pangolin.co.nz/yotreps/index.php

http://vos.noaa.gov/

http://www.ais-live.co.uk/

http://www.meteo.shom.fr/vos-monitoring/

http://www.nrlmry.navy.mil/

http://www.boatingsf.com/

http://www.shinemicro.com/

http://www.lrfairplay.com/

http://myweb.tiscali.co.uk/mikeandtina/

http://www.researchvessels.org/

4. Design and Implementation

4.1. Goal

The goal of this project is to come up with a simple yet powerful search engine that collects all the essential data on ship and port information available in the web and present it in a much understandable and faster manner.

4.2. Overall Design

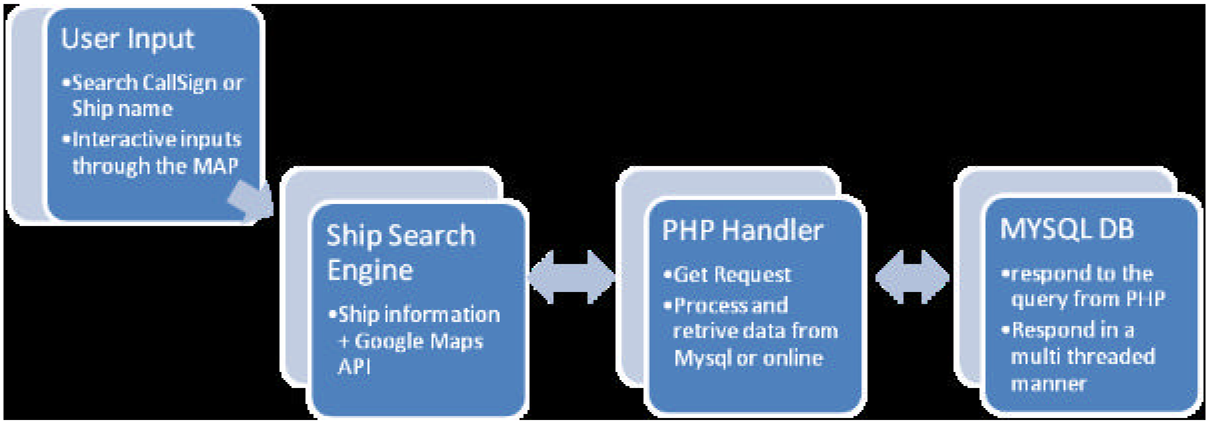

User Input
•Search CallSign or Ship name
•Interactive inputs through the MAP
Ship Search Engine
•Ship information + Google Maps API
PHP Handler
•Get Request
•Process and retrive data from Mysql or online
MYSQL DB
•respond to the query from PHP
•Respond in a multi threaded manner

Figure 1: Overall Design and Flow of the search engine

4.3. Approach Taken

Overall the project contains 3 phases

1. Data Extraction and Manipulation
2. Data Maintenance and updation
3. Data Presentation using Google Maps

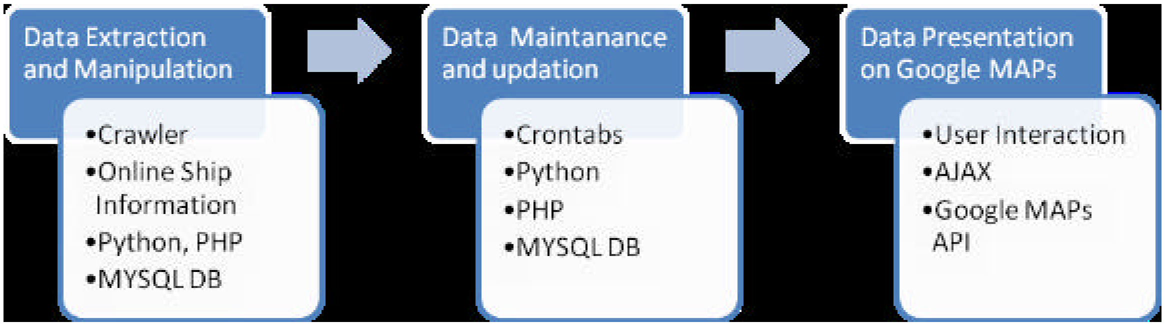


Figure 2: The 3 important phases of the search engine

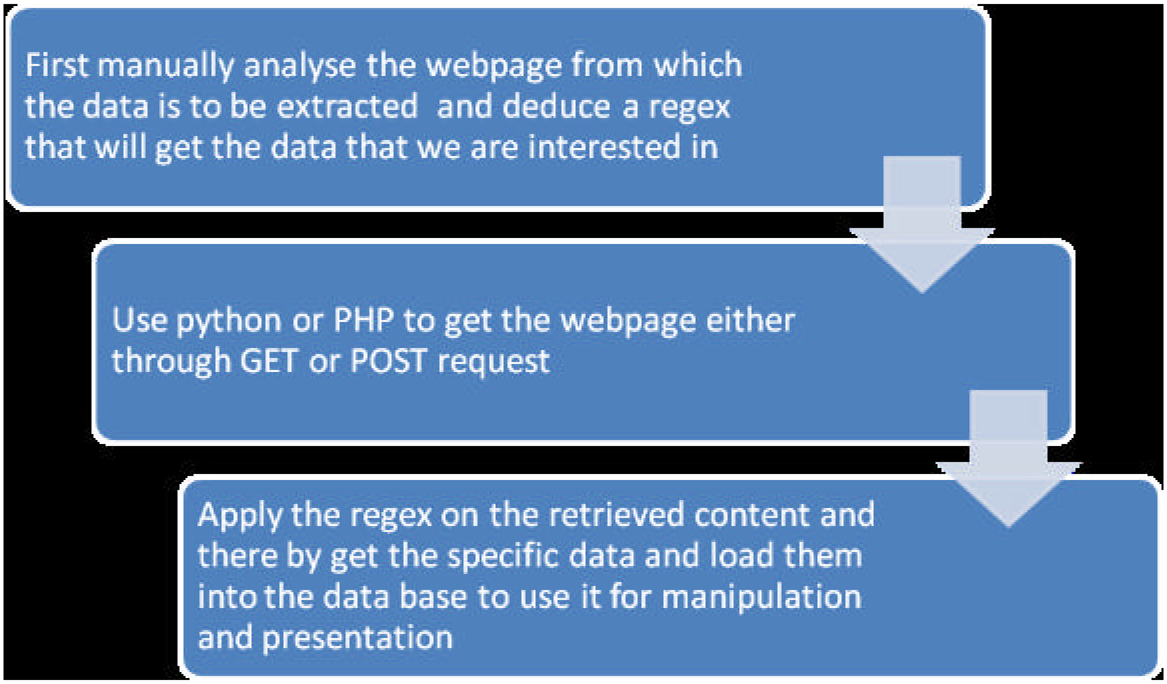
4.4. Data Extraction and Manipulation

In order to make this the most comprehensive ship database in the world, a specific crawler is built to extract the following details

1. Port Information
2. Ship Information
3. Ship Route
4. Ships in the Port

4.4.1. How the Information is collected

The following is a general protocol used to crawl specific data from the page. Since different pages have different structures, one can use the generic method presented below to retrieve specific from the page.



First manually analyse the webpage from which the data is to be extracted and deduce a regex that will get the data that we are interested in

Use python or PHP to get the webpage either through GET or POST request

Apply the regex on the retrieved content and there by get the specific data and load them into the data base to use it for manipulation and presentation

Figure 3: General Protocol to retrieve specific data from any webpage

4.4.2. Port information

The port information comprises of the following

1. Port name
2. Latitude and longitude of the port
3. Port website and
4. the schedule information the port website provides

In order to get the port information the following web pages were crawled

http://www.worldportsource.com/

http://e-ships.net/

http://www.mgn.com/worldports/ports.cfm

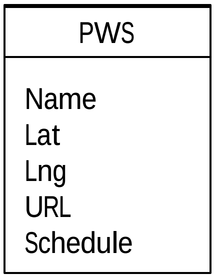
Further the most daunting task is that of collecting the schedule information present in each port's website . The way the schedule is collected and how the collected schedule information is used to predict the ports the ships visited in the past and will visit in future in addition to the route of the ship obtained by crawling is explained while extracting the ships in port information.

Schema Design in the data base

Ports with schedule(PWS)

```
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| lat      | float        | YES  |     | NULL    |       |
| lng      | float        | YES  |     | NULL    |       |
| name     | varchar(100) | YES  |     | NULL    |       |
| url      | varchar(500) | YES  |     | NULL    |       |
| schedule | varchar(500) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

Class Diagram



4.4.3. Ship information

The Ship information comprises of the following

1. Ship Name
2. Call sign
3. Flag
4. Service

In order to get the ship information the following web pages were crawled

http://cgmix.uscg.mil/PSIX/VesselSearch.aspx

http://www.shom.fr/cgi-bin/meteo/list_vos_country.cgi?country=US

http://e-ships.net/

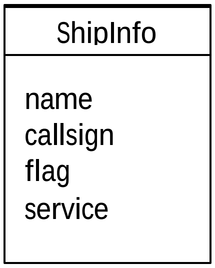
The difficulty in crawling ship information is that the information is not explicitly provided. So data is posted in a sequential order and analyzed to get information pertaining to the ship.

Schema Design in the data base

shipinfo

```
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| name     | varchar(100) | YES  |     | NULL    |       |
| callsign | varchar(100) | YES  |     | NULL    |       |
| flag     | varchar(100) | YES  |     | NULL    |       |
| service  | varchar(100) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

Class Diagram



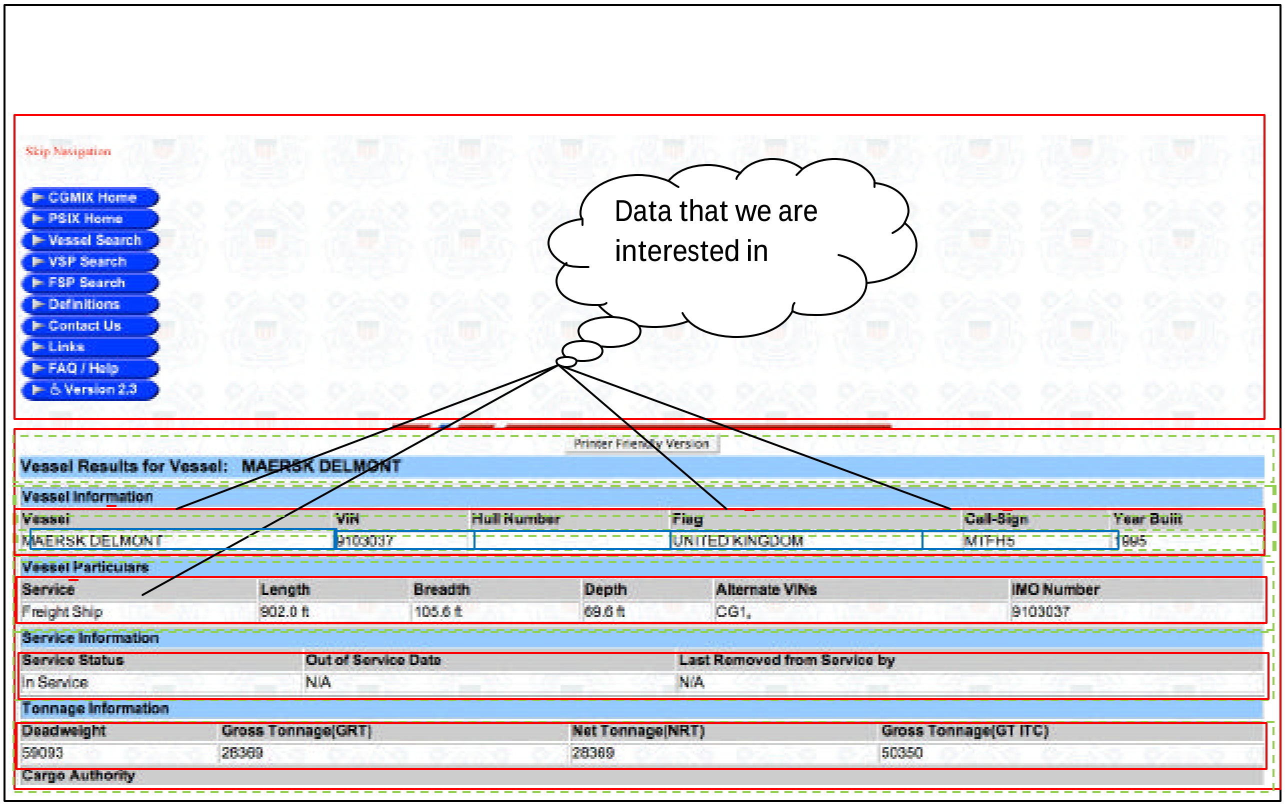
Example

Using the above general protocol for data extraction, Ship info can be collected from http://cgmix.uscg.mil/PSIX/VesselSearch.aspx in the following manner

First manually analyze the webpage from which the data to be extracted and deduce a regex that will get the data that we are interested in

Screen shot of page structure for the following page

http://cgmix.uscg.mil/PSIX/VesselResults.aspx?VesselID=1



Data that we are interested in

Vessel Results for Vessel: MAERSK DELMONT

Vessel Information

Vessel	VIN	Hull Number	Flag	Call-Sign	Year Built
MAERSK DELMONT	9103037		UNITED KINGDOM	MTFH5	1995

Vessel Particulars

Service	Length	Breadth	Depth	Alternate VINs	IMO Number
Freight Ship	902.0 ft	105.6 ft	69.6 ft	CG1,	9103037

Service Information

Service Status	Out of Service Date	Last Removed from Service by
In Service	N/A	N/A

Tonnage Information

Deadweight	Gross Tonnage(GRT)	Net Tonnage(NRT)	Gross Tonnage(GT ITC)
59093	28369	28369	50350

Cargo Authority

Figure 4: screen shot of http://cgmix.uscg.mil/PSIX/VesselResults.aspx?VesselID=1 with layouts shown in differnt color, representing rectangles red: tables, green: tr, blue: td and red circles: Data to be extracted

DOM structure for the page displayed above

```
<html>
  <body background="/psix/Images/CGShieldBackGround.gif" ms_positioning="GridLayout">
   <form id="Form1" action="VesselResults.aspx?VesselID=1" method="post" name="Form1">
     <table id="HeaderPanel" width="100%" cellspacing="0" cellpadding="0" border="0">
        <tr>
```

```
<td>
  <table width="755"></table>
  <table id="VesselResultsPanel" width="100%">
    <tr></tr>
    <tr>
      <td>
        <table width="100%" b
        order="1">
          <tr></tr>
          <tr style="font-
weight: bold; font-family: Arial;"></tr>
          <tr>
            <td>
                <span id="LabelVessel" title="MAERSKDELMONT">MAERSK
        DELMONT</span>

            </td>
```

The Page is extracted using Urllib.urlopen and by get method sending a request for vesselID

Data we are interested in

Analyzing the layout

According to the layout of the DOM structure displayed above, the data we are interested in is enclosed in a span and the span is present inside td. The td in turn is present inside a table which is a subset of another table.

But the cache here is that each and every span that we are interested in has got a special Id which is unique and thereby we can use that to get the data through regex

The page has been analyzed and we need to deduce a regex to retrieve the data that's of interest to us

By analyzing the page we were able to come up with a unique identifier that could be used to retrieve the fields. The following regex can be used on the above DOM to retrieve the ship name as MAERSK DELMONT

```
<span id="LabelVessel">(.*?)</span>
```

And incase if there is any <> we can remove them using the following regex

```
'<[^!>](?:[^>]|\n)*>', ''
```

Python Code

So the python code to get the above data extraction part will be

```
import urllib
for i in range(1,100):
  vid=str(i)
```

```
   data=urllib.urlopen("http://cgmix.uscg.mil/PSIX/VesselResults.aspx?VesselID="+vid)
doc=data.read()

  p=re.compile(r'<table width="100%" border="1">(.*?)</table>',re.I | re.M | re.S)
   subdoc=p.findall(doc)
   if subdoc==[]:
       continue
   p=re.compile(r'<span id="LabelVessel">(.*?)</span>',re.I | re.M | re.S)
   vessel=p.search(subdoc[0]).group()
   vessel=re.sub('<[^!>](?:[^>]|\n)*>', '', vessel)
   #print vessel
   #print callsign

 self.con = MySQLdb.connect(host = "hostname",port= 3306,user = "username",passwd = "password",db
= "databasename")
self.cursor = self.con.cursor ()
              try:
```

The regex is applied to get the specifics from the webpage. In this case Vessel Name is retrieved

```
                  self.cursor.execute("""insert into shipinfo(name,callsign,flag,service)
values(%s,%s,%s,%s)""",(vessel,callsign,country,shiptype))
              except:
                  print 'Error writing to db'
end_time = time()
print "The program spents", end_time -
start_time, "seconds"
```

The extracted data is stored in Mysql database to use it for presentation here it is stored in table ship info

4.4.4. Ship Route information

The Ship route information comprises of the following

1. Call sign
2. Time stamp at this position
3. Latitude
4. Longitude

In order to get the ship information the following web pages were crawled http://www.sailwx.info/

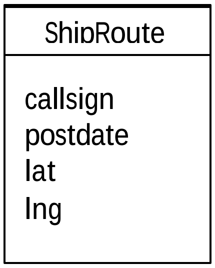
The difficulty in crawling the ship movement is that the information is not explicitly provided and only one site provides the details. The data is collected through get request from the webpage by passing the call sign as the request parameter. The call sign is obtained from our database through the data collected for ship information.

Schema Design in the data base

shiproute

```
+-----------+-------------+------+-----+--------------------+-------+
| Field     | Type        | Null | Key | Default            | Extra |
+-----------+-------------+------+-----+--------------------+-------+
| callsign  | varchar(100) | NO   | PRI |                    |       |
| posdate   | datetime    | NO   | PRI | 0000-00-00 00:00:00 |       |
| lat       | float       | YES  |     | NULL               |       |
| Ing       | float       | YES  |     | NULL               |       |
+-----------+-------------+------+-----+--------------------+-------+
```

Class Diagram



Example

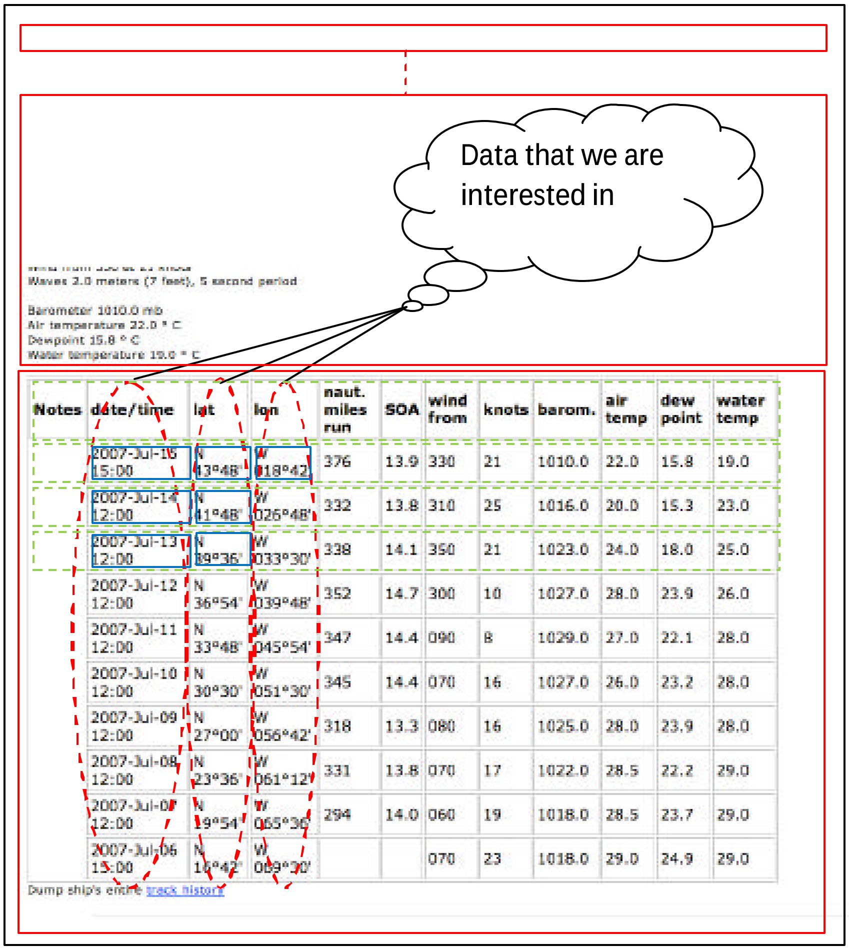
Using the above general protocol on http://www.sailwx.info/shiptrack/shipposition.phtml?call=9HOB7

Ship Route can be collected in the following manner

1. First manually analyze the webpage from which the data to be extracted and deduce a regex that will get the data that we are interested in

Screen shot of page structure for the following page

http://www.sailwx.info/shiptrack/shipposition.phtml?call=9HOB7



Waves 2.0 meters (7 feet), 5 second period

Barometer 1010.0 mb
Air temperature 22.0 ° C
Dewpoint 15.8 ° C
Water temperature 19.0 ° C

Notes	date/time	lat	lon	naut. miles run	SOA	wind from	knots	barom.	air temp	dew point	water temp
	2007-Jul-15 15:00	N 13°48'	W 018°42'	376	13.9	330	21	1010.0	22.0	15.8	19.0
	2007-Jul-14 12:00	N 41°48'	W 026°48'	332	13.8	310	25	1016.0	20.0	15.3	23.0
	2007-Jul-13 12:00	N 39°36'	W 033°30'	338	14.1	350	21	1023.0	24.0	18.0	25.0
	2007-Jul-12 12:00	N 36°54'	W 039°48'	352	14.7	300	10	1027.0	28.0	23.9	26.0
	2007-Jul-11 12:00	N 33°48'	W 045°54'	347	14.4	090	8	1029.0	27.0	22.1	28.0
	2007-Jul-10 12:00	N 30°30'	W 051°30'	345	14.4	070	16	1027.0	26.0	23.2	28.0
	2007-Jul-09 12:00	N 27°00'	W 056°42'	318	13.3	080	16	1025.0	28.0	23.9	28.0
	2007-Jul-08 12:00	N 23°36'	W 061°12'	331	13.8	070	17	1022.0	28.5	22.2	29.0
	2007-Jul-07 12:00	N 19°54'	W 065°36'	294	14.0	060	19	1018.0	28.5	23.7	29.0
	2007-Jul-06 15:00	N 16°42'	W 069°30'			070	23	1018.0	29.0	24.9	29.0

Dump ship's entire track history

Figure 5: screen shot of http://www.sailwx.info/shiptrack/shipposition.phtml?call=9HOB7with layouts shown in differnt color, representing rectangles red: tables, green: tr, blue: td and red circles: Data to be extracted

DOM structure for the page displayed above

```
<html>
 <body marginwidth="0" marginheight="0"bgcolor="#ffffff" leftmargin="0" topmargin="0">
  <div id="LayoutTable">
   <table width="982" cellspacing="0" cellpadding="0" border="0"></table>
   <table width="982" cellspacing="0" cellpadding="0" border="0">
    <tr valign="top"></tr>
    <tr valign="top">
     <td rowspan="9"/>
  <td rowspan="9"></td>
  <td colspan="10"/>
  <td height="7"/>
</tr>
<tr valign="top"></tr>
<tr valign="top"></tr>
<tr valign="top">
  <td colspan="2"/>
  <td colspan="6">
    <p class="Body">
      <span class="BODY">
        <table cellpadding="3" border>
        </table>
    </span>
```

Data we are interested in is present in this table as first 3 columns

acted using Urllib.urlopen

The regex is applied to get the specifics from the webpage

Analyzing the layout

By analyzing the above DOM we can deduce that the data of interest resides in the first three columns of the table enclosed within a span. So in order to extract the data we need to identify the unique feature that could retrieve the table first. From this we can retrieve the first 3 columns and load to our database.

The cache here is that the table we need to retrieve has cellpadding=3 and border none is unique. This can be used to extract the table.

The extracted data is stored in Mysql database to use it for presentation

By analyzing the page we were able to come up with a unique identifier that could be used to retrieve the fields. The following regex can be used on the above DOM to retrieve the first three columns of the table

<table cellpadding=3 border>(.*?)</table>

And incase if there is any <> we can remove them using the following regex

'<[^!>](?:[^>]|\n)*>', ''

Python Code

So the python code to get the above data extraction part will be

```
import urllib2
```

```
data=urllib2.urlopen("http://www.sailwx.info/shiptrack/shipposition.phtml?call="+callsign)
doc=data.read()
 print callsign

 p=re.compile(r'<table cellpadding=3

 border>(.*?)</table>',re.I | re.M | re.S)

     subdoc=p.search(doc).group()

      if subdoc!=None:

         p=re.compile(r'<tr>(.*?)</tr>',re.I | re.M | re.S)

         rows=p.findall(subdoc)

       #print rows[10]

         p=re.compile(r'<(TD|td)>(.*?)</(TD|td)>',re.I | re.M | re.S)

         for i in range(1,len(rows)):

            cols=p.findall(rows[i])

            #print i

           #print cols

            posdate=strptime(cols[1][1],"%Y-%b-%d %H:%M")

            posdate=strftime("%Y-%m-%d %H:%M:%S",posdate)

            print postdate

           con = MySQLdb.connect(host = "localhost",port= 3306,user = "root",passwd = "",db ="seaport")
                                                                                  cursor=con.cursor()
                                  cursor.execute("insert       into       shiproute(callsign,posdate,latitude,longitude)
values(%s,%s,%s,%s)",(callsign,posdate,cols[2][1],cols[3][1]))
```

4.4.5. Ships In port

The Ship In port information comprises of the following

1. Port name
2. Ship name
3. call sign

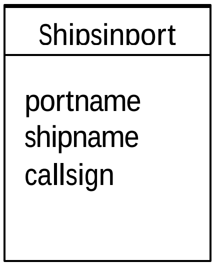
The shipsinport table gives us information about all the ships that have visited the port, ships currently present in the port and the ships that will visit the port in future.

It is constructed by retrieving all the shipname, callsign from shipinfo table and check whether any of this occurs in the port schedule webpage. If it occurs it will be recorded in the shipsinport table stating that this particular ship with this name and call sign has been to this port at some point of time.

Schema Design in the data base

```
shipsinport
+----------+--------------+------+-----+---------+-------+

| Field    | Type         | Null | Key | Default | Extra |

+----------+--------------+------+-----+---------+-------+

| portname | varchar(100) | YES  |     | NULL    |       |

| shipname | varchar(100) | YES  |     | NULL    |       |

| callsign | varchar(100) | YES  |     | NULL    |       |

+----------+--------------+------+-----+---------+-------+
```

Class Diagram

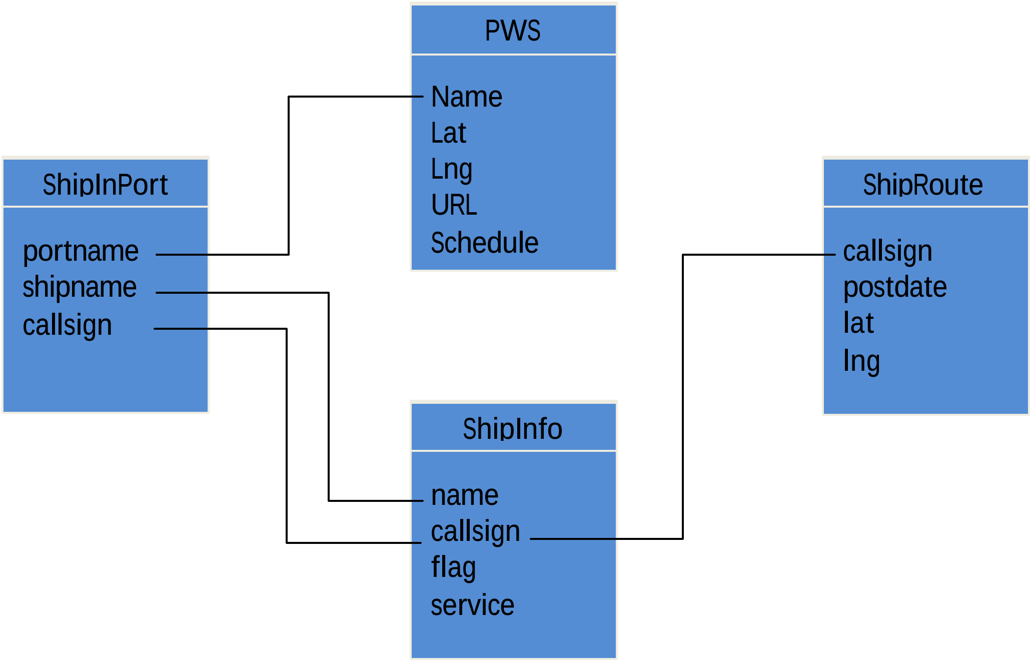


4.5. Data Maintenance and updation

Data Maintenance and updation is said to be the most significant part of a search engine, because this is where new data is extracted continuously from the web to keep the system up to date and the old data is cleansed, archived and also sometimes mined to find if any pattern existed in it.

The data extracted is maintained using MYSQL database and the updation is done by scheduled running of python auto scripts (Crawler) using Crontabs on a UBUNTU machine.

4.5.1. Database Design



4.5.2. CronTabs and How they work

Cron is the name of program that enables unix users to execute commands or scripts (groups of commands) automatically at a specified time/date.

There are a few different ways to use cron.

In the */etc* directory the user will probably find some sub directories called '*cron.hourly*', '*cron.daily*', '*cron.weekly*' and '*cron.monthly*'. Placing a script into one of those directories will run hourly, daily, weekly or monthly, depending on the name of the directory.

If more flexibility is needed, one can edit a crontab (the name for cron's config files). The main config file is normally */etc/crontab*. The crontab will look something like this:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

The first part is almost self explanatory; it sets the variables for cron.

SHELL is the 'shell' cron runs under. If unspecified, it will default to the entry in the */etc/passwd* file.

Now for the more complicated second part of a crontab file, an entry in cron is made up of a series of fields, much like the /etc/passwd file is, but in the crontab they are separated by a space. There are normally seven fields in one entry. The fields are:

minute hour dom month dow user cmd

minute : This controls what minute of the hour the command will run on, and is between '0' and '59'
hour : This controls what hour the command will run on, and is specified in the 24 hour clock, values must be between 0 and 23 (0 is midnight)
dom : This is the Day of Month, that the user wants the command run on, e.g. to run a command on the 19th of each month, the dom would be 19.
Month : This is the month a specified command will run on, it may be specified numerically (0-12), or as the name of the month (e.g. May)
Dow : This is the Day of Week that the user wants a command to be run on, it can also be numeric (0-7) or as the name of the day (e.g. sun).
user : This is the user who runs the command.
Cmd : This is the command that the user wants to run. This field may contain multiple words or spaces.

If the user does not wish to specify a value for a field, he/she can place a * in the field.

Example

```
59 11 * * 1,2,3,4,5 root backup.sh
```

Will run backup.sh at 11:59 Monday, Tuesday, Wednesday, Thursday and Friday,

In a similar fashion this corntab is used to run our python script(Crawler) on a scheduled basis to keep the system upto date.

4.6. Data Presentation

The data extracted and maintained is presented over a Google Maps Mashup upon user request.

The users were given the following options to interact with the mashup and retieve the result

1. Search with Callsign [Both in DB and online]
2. Search with Ship name [Both in DB and online]
3. Plot the ports on moving the map
4. Retrieve the information about the ship on mouse over the plotted ships

5. Retrieve the information about the ports on mouse click over the ports
6. Easy navigation over the map using the GOverviewMapControl
7. Display the details pertaining to the user search on the side bar
8. Get the list of ports for the portion map displayed
9. Get the list of ships present currently for the portion of map displayed
10. Clearing the previously plot
11. Able to see multiple ship routes in the same map
12. The maps also adjust dynamically to the route of the ship so that the whole route is visible to the user

4.6.1. How it is Different

The greatest advantage of this search is the way data collected and presented. By going through the state of Art it can be stated that the data is mostly presented in a raw format like excel or CSV. Even if some provide the data on Maps, the Maps on which the data is provided is not interactive and use the classic web application model to fetch the data and present it. This leads to higher retrieval times there by frustrating the user.

On the contrary this search engine tries to present the data on Google Maps which is very much interactive at the same time uses AJAX for request to the server. AJAX is used to retrieve the results asynchronously from the server based on user request and thereby avoiding page refreshes, providing an user friendly environment, decreasing retrieval time and user frustration. The system also provides an interactive environment with essential details and querying options.

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

Every user action that normally would generate an HTTP request takes the form of a JavaScript call to the Ajax engine instead. Any response to a user action that doesn't require a trip back to the server — such as simple data validation, editing data in memory, and even some navigation — the engine handles on its own. If the engine needs something from the server in order to respond — if it's submitting data for processing, loading additional interface code, or retrieving new data — the engine makes those requests asynchronously, usually using XML, without stalling a user's interaction with the application.

The following is the comparison of AJAX architecture with traditional web architecture

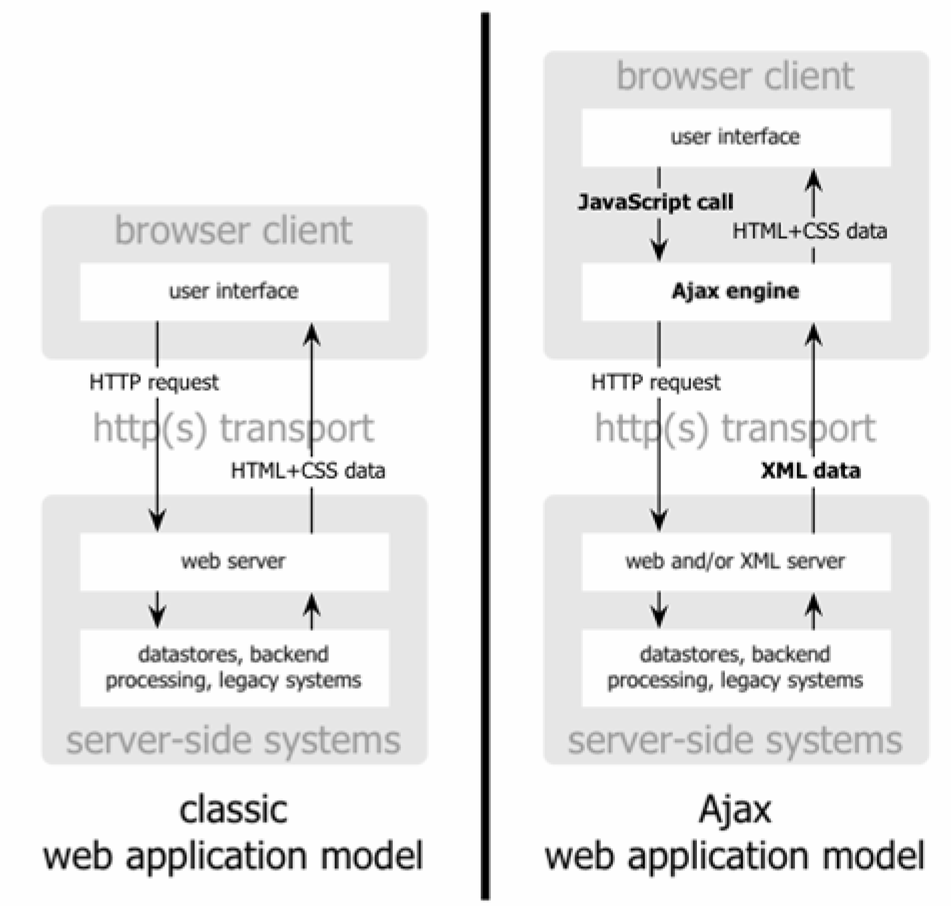


Figure 6: Comparison of AJAX architecture with traditional web architecture

4.6.2. Packages, modules, APIs and Tools used

LAMP, PHP, Python and Google Maps API

Auto setup using CRON jobs

Curl

4.6.3. Significance of Crawling

Another important aspect of this project is that the crawler used during data extraction is a multi threaded crawler. Thus it facilitates crawling of multiple pages at the same time. that can crawl millions of pages very quickly and there by can always keep the database up-to-date and can easily by run through Crontabs for scheduled crawling without human intervention.

4.6.4. Ship Movements Tracking

The ship movement is tracked by the data extracted and stored in the shiproute and shipsinport tables. This data from shproute table is then plotted on Google map to trace the path the ship has taken. The data from shipsinport table is also plotted on Google map to determine the ports the ships are currently present, the ports the ship has already visited and the ports the ships will visit next.

Ship Movements is plotted in two different methods

Plotting from DB - that is by extracting the data from tables shiproute and shipsinport

Plotting real time from Web – that is extracting the data from sailwx site realtime on user request

4.7. Functional Description of the GUI

The following are the features that are present in the proposed ship search engine.

<<Can you add a section describe all the query capabilities, what can be queried, user inputs, and which box to submit, etc.>>

<<Which site do you do online query to get the ship locations and tracking information?>>

General Layout

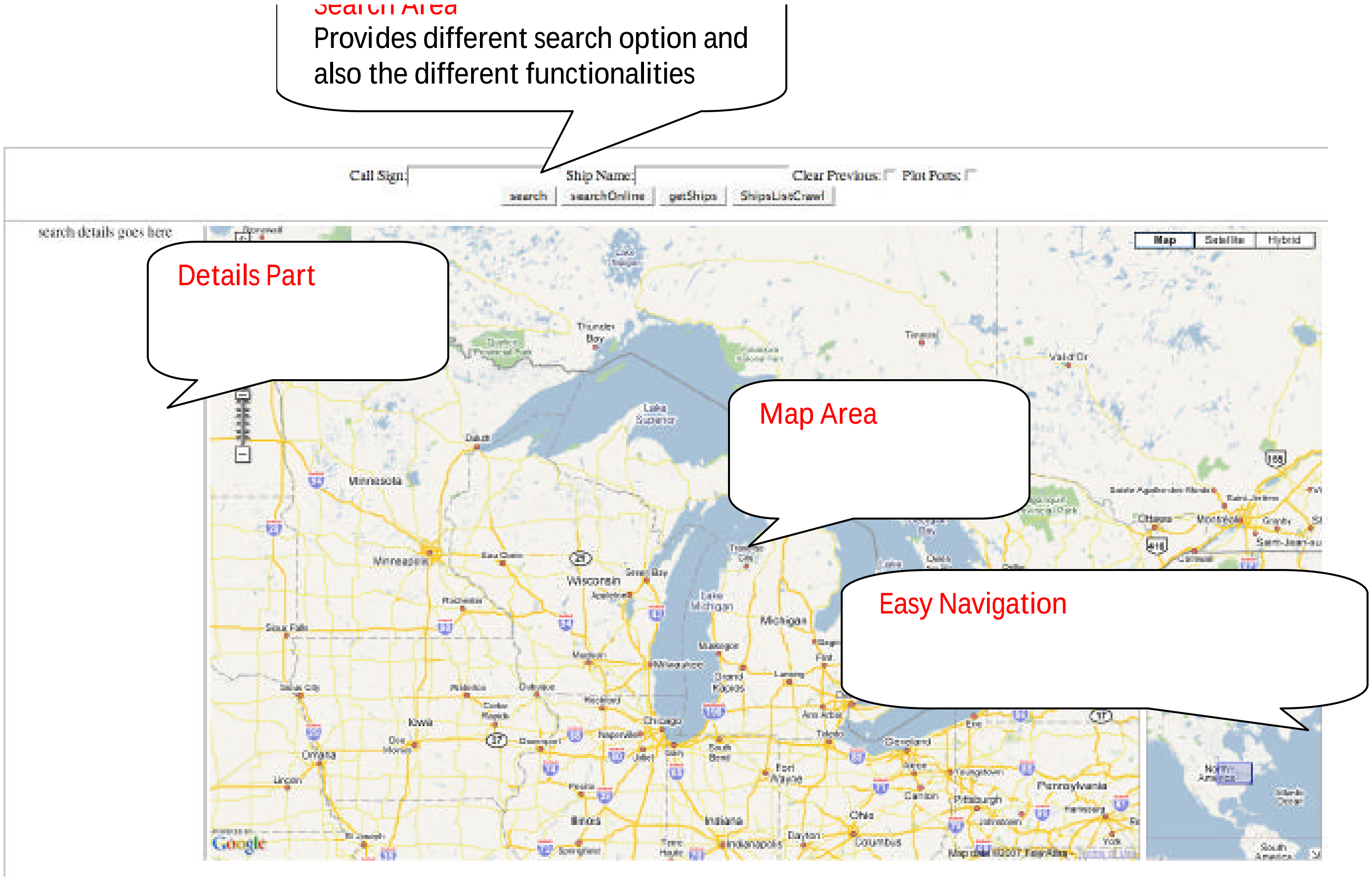


Figure 7: General Layout of the webpage

Introducing auto-suggest for Call Sign and also Ship Name - there by making the search very easier

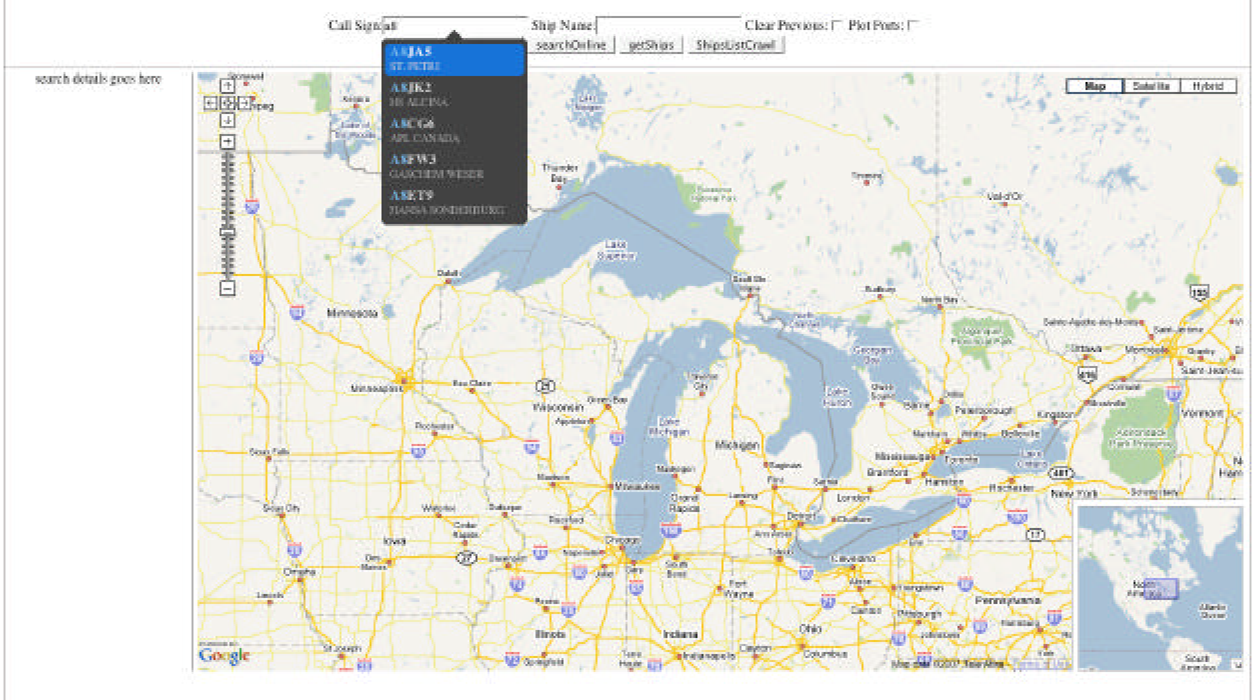


Figure 8: Showing the auto suggest feature for call sign

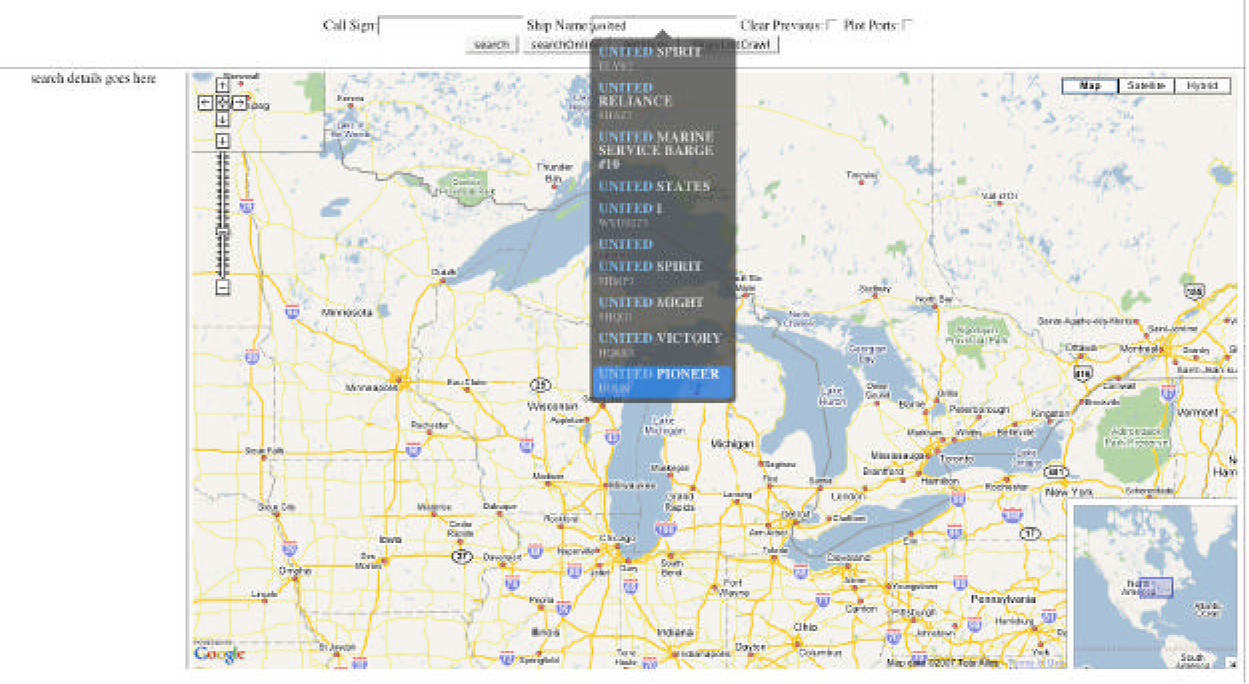
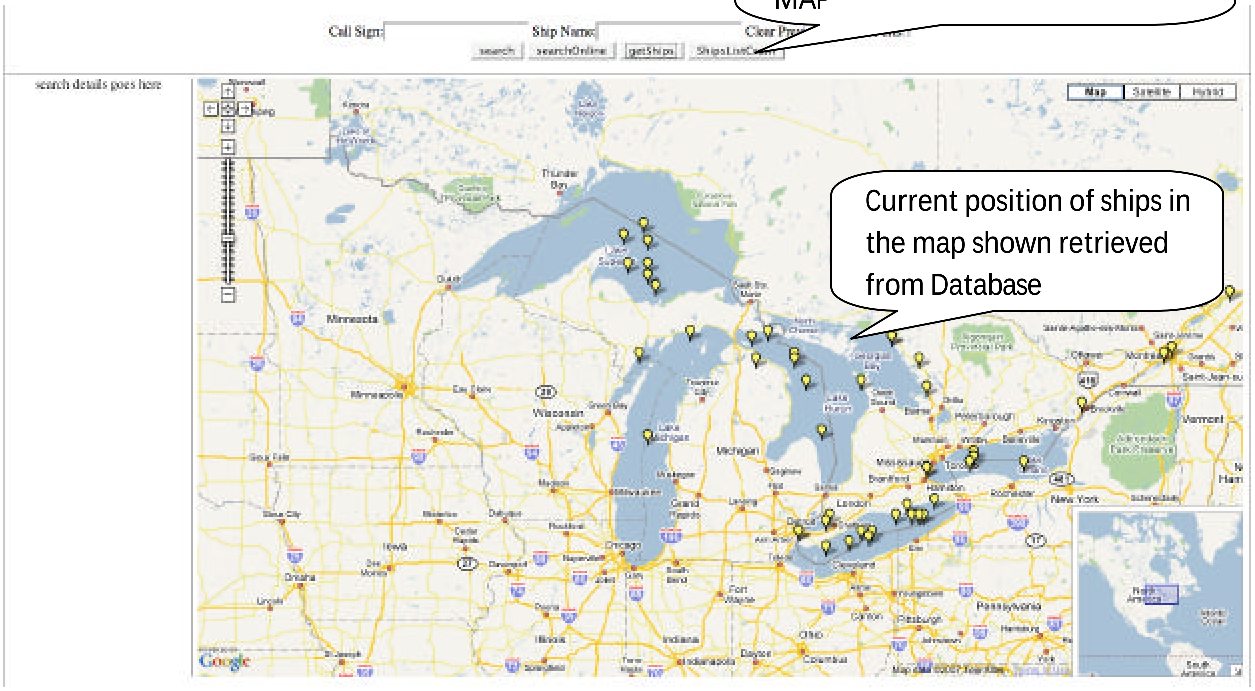


Figure 9: Showing the auto suggest feature for ship name

Plotting the present position of the ships from the database

Yellow markers represent the present position of the ships as retrieved from the database

On clicking this [Getships] the following data is displayeds on the MAP

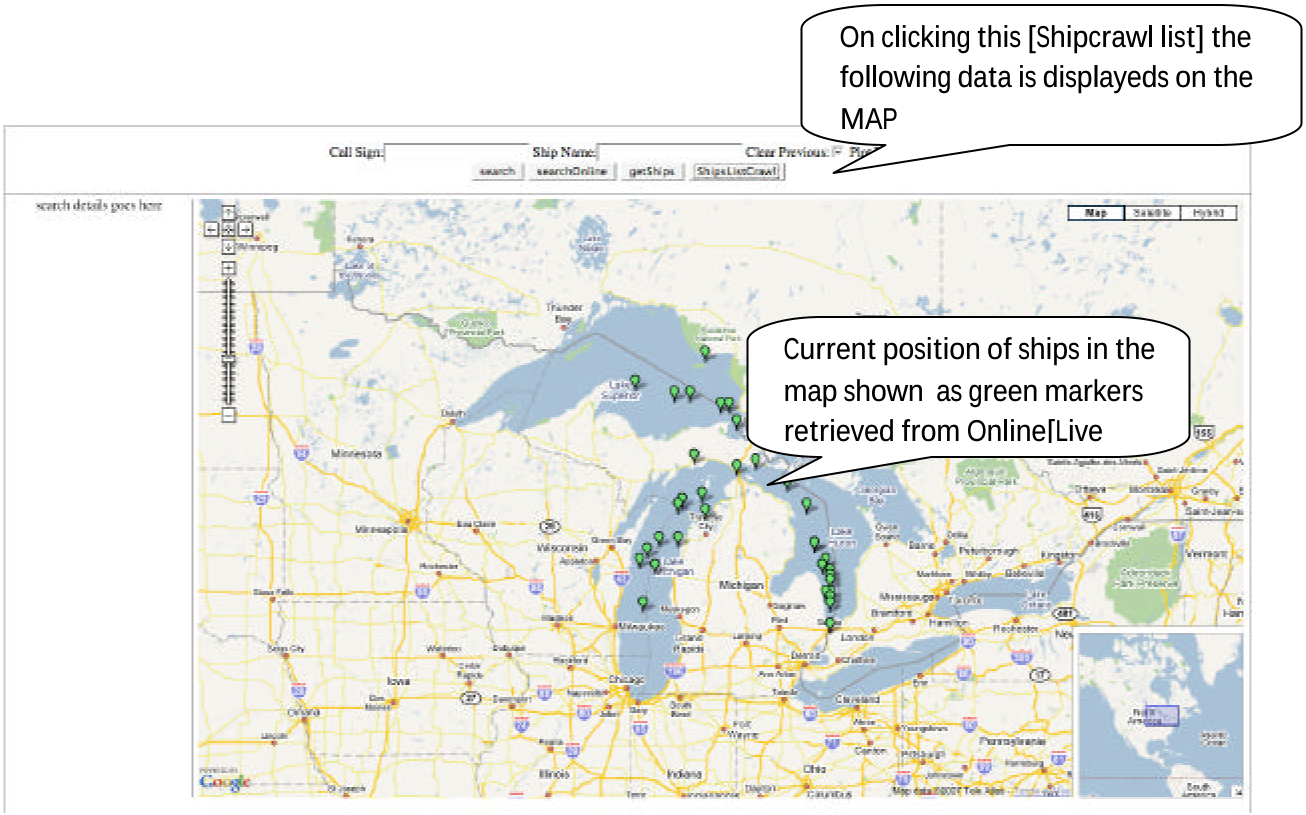


Current position of ships in the map shown retrieved from Database

Figure 10: Explaining how to get the list of ships for the region of the map shown from database

Plotting the present position of the ships from the Online

Green markers represent the present position of the ships as retrieved from Online



Call Sign: Ship Name: Clear Previous: ☑ Plo

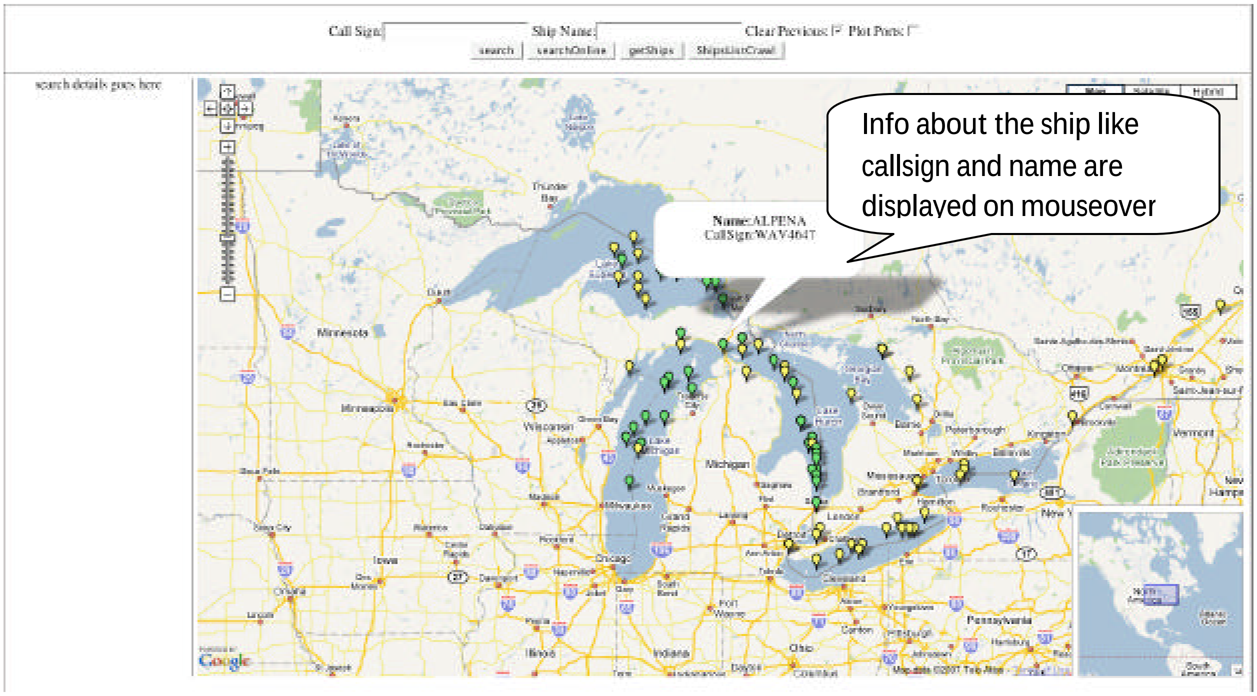
search details goes here

On clicking this [Shipcrawl list] the following data is displayeds on the MAP

Current position of ships in the map shown as green markers retrieved from Online[Live

Figure 11: Explaining how to get the list of ships for the region of the map shown from online

Getting Information about the plotted ships on mouseover the markers on the map



Call Sign: Ship Name: Clear Previous: ☑ Plot Ports: □

search | searchOnline | getShips | ShipsListCrawl

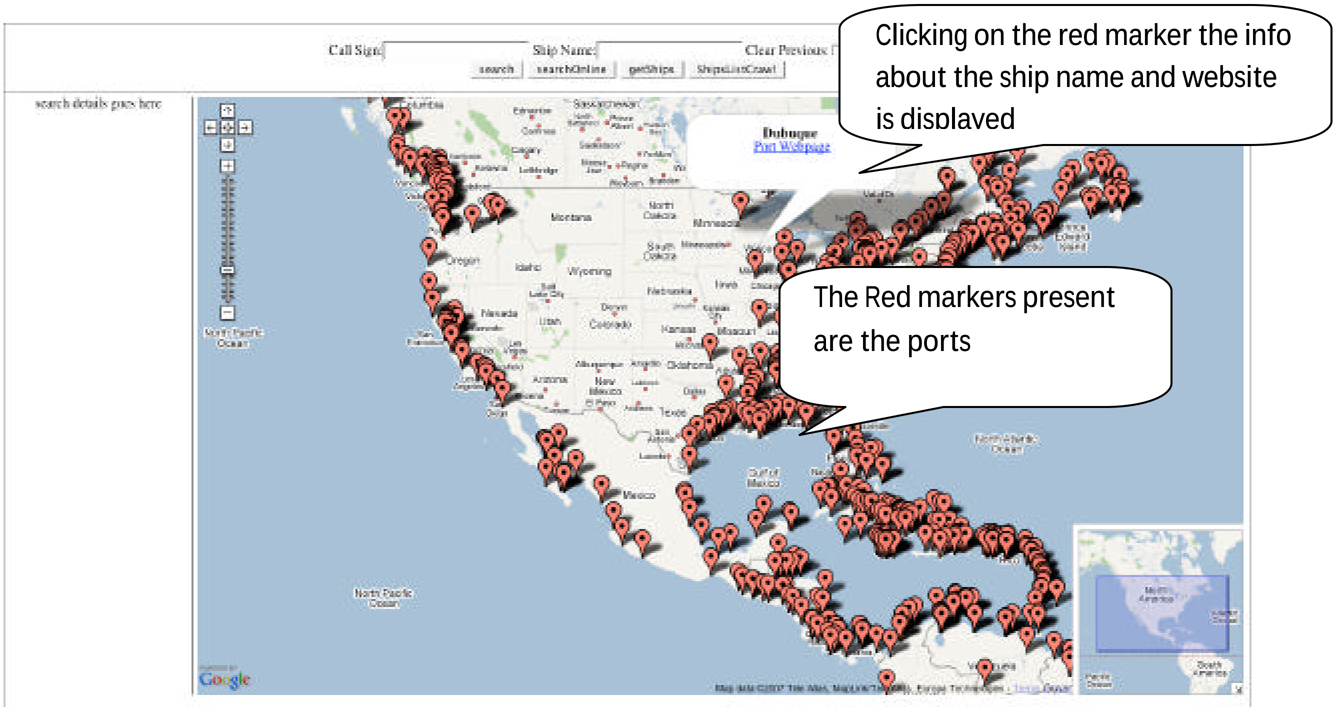
search details goes here

Name:ALPENA
CallSign:WAV4647

Info about the ship like callsign and name are displayed on mouseover

Figure 12: Shows how to get info about a particular ship. Yellow markers: Ships plotted from DB, Green Markers: Ships plotted from online

Plotting the Ports on moving the map



Call Sign: Ship Name: Clear Previous: □

search | searchOnline | getShips | ShipsListCrawl

search details goes here

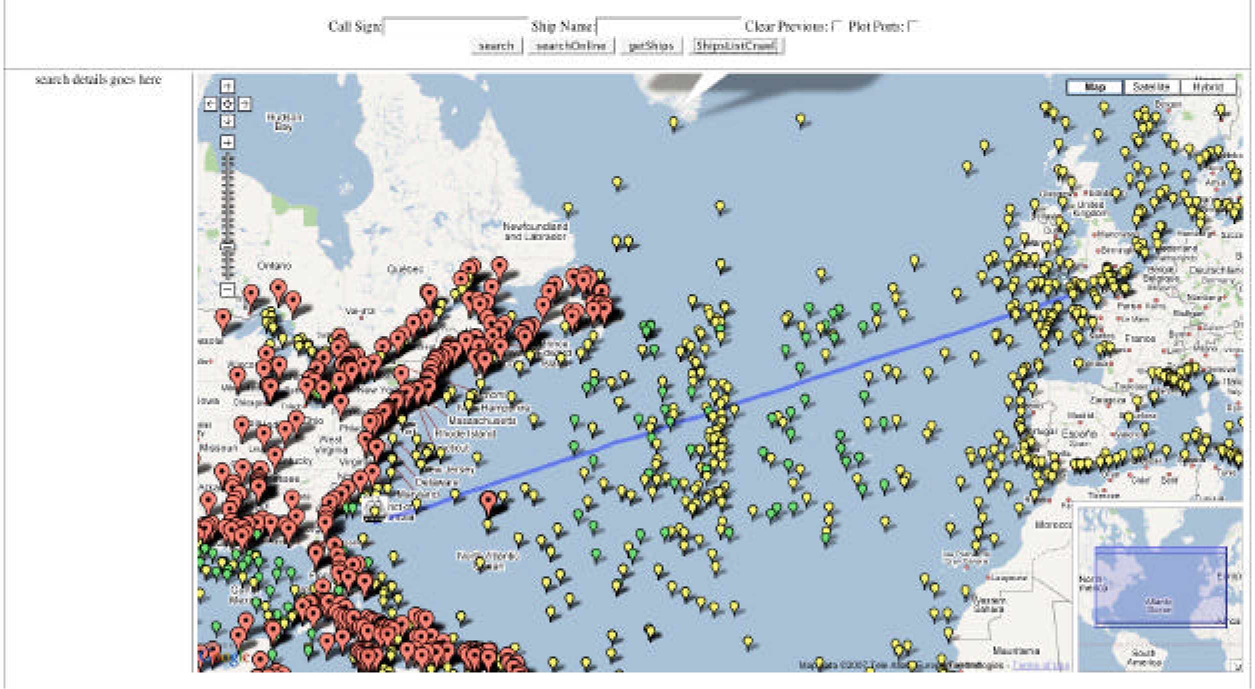
Dubuque
Port Webpage

Clicking on the red marker the info about the ship name and website is displayed

The Red markers present are the ports

Figure 13: Shows how ports are plotted and how to the information about the ports on clicking on the markers

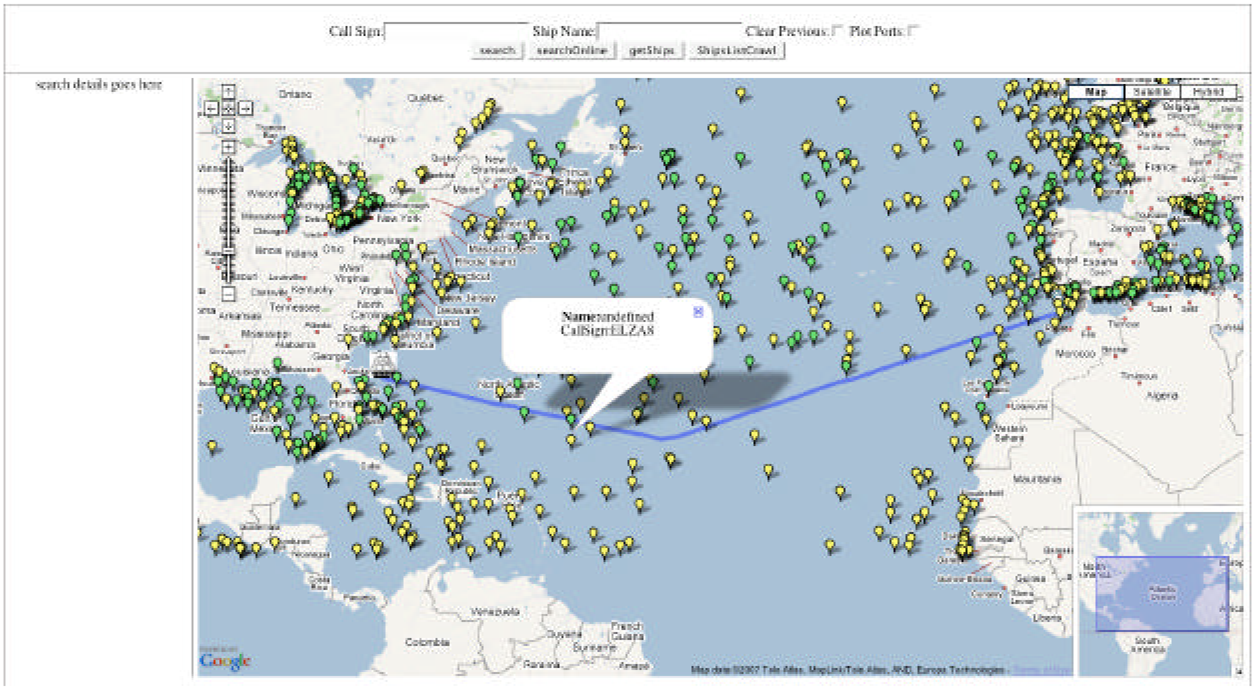
Plotting the route of the ship from online and from DB by just clicking on the ship displayed on the map



Call Sign: Ship Name: Clear Previous: □ Plot Ports: □

search details goes here

Figure 14: Shows the route of the ship along with ports and ships in that area of the map



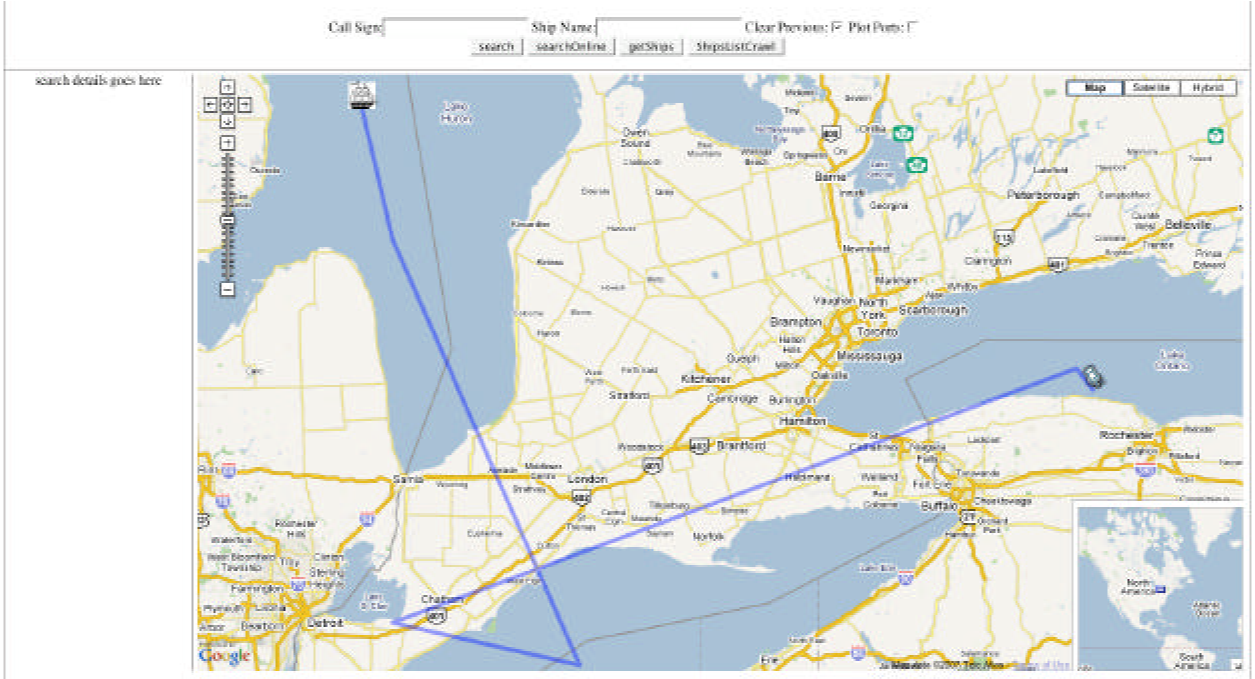
Call Sign: Ship Name: Clear Previous: □ Plot Ports: □

search searchOnline getShips ShipsListCrawl

search details goes here

Name:undefined
CallSign:ELZA8

Getting a closer and clear look on the map



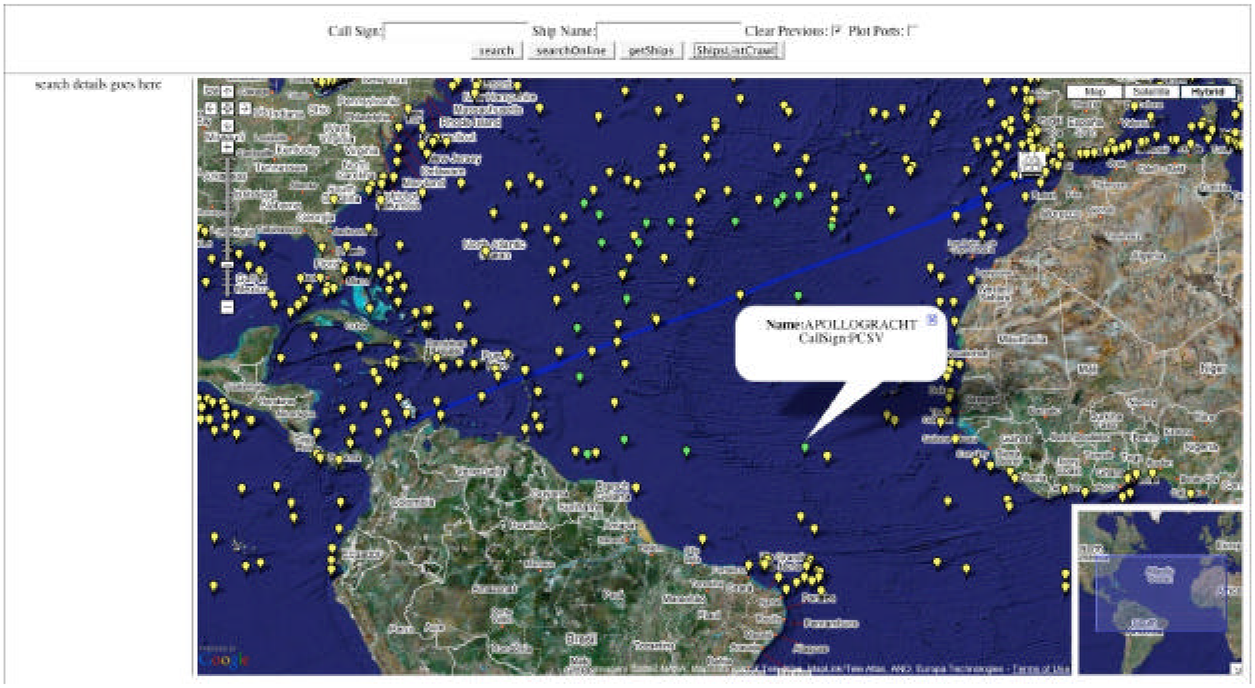
Call Sign: Ship Name: Clear Previous: ☑ Plot Ports: □

search searchOnline getShips ShipsListCrawl

search details goes here

Figure 15: Shows a closer look at the route the ships been through

Different view of the map



Call Sign: Ship Name: Clear Previous: ☑ Plot Ports: □

search details goes here

Name:APOLLOGRACHT
CallSign:PCSV

Figure 16: Another view of the map

5. Results and Analysis

The database comprises of the following information

Number of port information in database: 2037
Number of unique ships details in database: 365913
Number of unique ships with call sign in database: 131280
Number of ship movement information: 1549 [This can be increased if we run the crawler for some long time]
Number of ports that have websites: 1458
Number of ports that have schedule information: 190

On analysis the following is found as an advantage over other ship information providers

1. The response time for the server to respond to the users request is very fast
2. The overall user interface design is so far the best for a ship search
3. Easy access to the required information and interactivity of the user is best in this search compared to others
4. No subscription or sign up to search through the database

6. Conclusion and Future work

6.1. Conclusion

The goal of the project to create a simple yet powerful search engine is achieved by mashing up the shipping data collected from the web with Google Maps. By incorporating the new AJAX technology the interactivity of the user is increased and in the same time the data is presented in a understandable manner. The advantage of this search engine is that it tries to project the data in a visual form on a Google Map and presents limited options that are capable of getting only the most essential data.

In future if more data is collected and if up to date information is maintained this would become a defacto standard search engine for ships.

6.2. Future work or Ways of Improvement

The project can be improved in many ways

1. By mashing up the system with the local news , predictions can be made about the diseases carried by the ship from one place to another
2. This model can be extended to predict the stock availability of a particular product the ship is carrying based on the time the ship will reach a particular port
3. This model can further be improved by crawling several other websites than what is crawled now to extract more information
4. This model if generalized can also be extended to other modes of transportation providing information about their current location.
5. Can incorporate advance search option like determining the route of a ship for a specific date range, comparing two ships (route, service etc.)

7. References

Installing LAMP on UBUNTU
http://www.howtoforge.com/lamp_installation_ubuntu6.06

Cheat sheet for Mysql commands

http://www.pantz.org/database/mysql/mysqlcommands.shtml

writing Mysql Scripts with python DB-API
http://www.kitebird.com/articles/pydbapi.html

Python and Mysql
http://dustman.net/andy/python/python-and-mysql

Python and Html processing
http://www.boddie.org.uk/python/HTML.html

Using full text indexes in Mysql
http://www.databasejournal.com/features/mysql/article.php/1587371

Python tutorial
http://www.python.org/doc/current/tut/tut.html

Pear Package browser
http://pear.php.net/packages.php

Google Maps API blog
http://googlemapsapi.blogspot.com/2006/07/speed-improvements-custom-cursors.html

using PHP/Mysql with Google Maps
http://code.google.com/support/bin/answer.py?answer=65622&topic=11369

Google Maps API tutorial
http://www.econym.demon.co.uk/googlemaps/

An unofficial Google Maps blog tracking the websites, mashups and tools being influenced by Google Maps
http://googlemapsmania.blogspot.com/

Google Maps API Mashups
http://www.programmableweb.com/api/google-maps/mashups

Google Maps API
http://www.google.com/apis/maps/

Newbie into to cron jobs
http://www.unixgeeks.org/security/newbie/unix/cron-1.html

Crontab quick reference
http://www.adminschoice.com/docs/crontab.htm

Running PHP scripts with Cron

http://www.htmlcenter.com/tutorials/tutorials.cfm/155/PHP/

www.piers.com
http://www.boatnerd.com/

http://www.sailwx.info/

http://cgmix.uscg.mil/PSIX/VesselSearch.aspx

http://www.shom.fr/cgi-bin/meteo/list_vos_country.cgi?country=US

http://www.lloydsmiu.com/lmiu/index.htm

http://www.ais-live.co.uk/

Gives the list of world ports based upon country selected
http://www.mgn.com/worldports/ports.cfm

http://www.worldportsource.com/

http://e-ships.net/ports.php

Get the details about all the research vessels with respect to country
http://www.researchvessels.org/