Satellite Vessel Tracker

Ana Gaspar
ana.ximenes.gaspar@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2021

Abstract

The document reports the development of a .NET application that aims to monitor vessels via satellite using a Geographic Information System (GIS). First, as the application needs to be developed in Windows Presentation Foundation (WPF) and having GIS as its main component, a previous research was carried out on the GIS topic and after that, several GIS controls were also researched, obtaining their respective functionalities, advantages and disadvantages. In addition, a research was also carried out on the type of system that the application will be (Vessel Monitoring System) and the existing softwares that have the same type of system and also present its functionalities. Secondly, there was also a survey on WPF, shapefiles, Web Map Service and a brief survey on the Global Positioning System (GPS). After the researches carried out, it was possible to present the best GIS that fulfilled all the company’s criteria for the development of the application (accessible API, GIS without a license or with a single value, edit the map (draw points, lines and polygons), developed in WPF, support shapefiles and WMS). Then, the code was implemented so that the application met the criteria and tests were carried out with the objective that the implementation developed met all the criteria. Finally, it can be said that the application fulfills all the requirements imposed by the company and after feedback from the company and some colleagues it can be concluded that the Application Programming Interface (API) of the application is accessible as intended.

Keywords: Vessel Monitoring System, Geographic Information System, Windows Presentation Foundation, Shapefiles and Web Map Service.

1. Introduction

Geographic Information System (GIS) [1] is a computer-based tool, whose objective is to process, store, manipulate, analyze and display data related to positions on the surface of the Earth. The data is then presented cartographically, graphically or as an report. To locate geographical information on a map, one can use either maps with a coordinate system to allow locations to be read, or shapes (polygons) of the geographic information, where shapes are drawn onto the map. As the purpose of the application is to monitor vessels via satellite, that type of system is a Vessel Monitoring System (VMS) and it is a form of satellite tracking using transmitters on board fishing vessels. A basic VMS consists of a Global Positioning System (GPS) receiver which plots the position of the vessels coupled with a communications device which reports the position. This project will be supported by the company XSealence - Sea Technologies [2] and the motivation is to improve a desktop application that the company has. To this end, a .NET application with the objective to monitor vessels via satellite in GIS will be developed. It is also fundamental that the application is developed in Windows Presentation Foundation (WPF) with the main component being a GIS, where the positions of vessels received via satellite will be represented. In addition, the GIS must support local maps or via Web Map Service (WMS), shapefiles and if the GIS has a license it must be paid only once. Besides that, an Application Programming Interface (API) should be created that allows the reuse of the GIS component in other applications and accessible (user-friendly). Briefly, there are criteria imposed by the company that should be fulfilled. Those criteria are accessible API, GIS without a license or with a single value, edit the map (draw points, lines and polygons), developed in WPF, support shapefiles and WMS.

2. Background

Prior to the implementation and development of the application, it was necessary to carry out some researches on GIS, WPF, VMS, shapefiles, WMS and one on GIS. For the GIS, it is presented the definition of it, the architecture and also the controls of GIS that were found as well as their functionalities. The advantages and disadvantages of each
GIS is also given. As the previous application developed by the company was used WinForms and not WPF, it is presented a definition of WinForms, how it works and also the advantages of which one with the objective to show why WPF is better to accomplish the company’s criteria (accessible API, GIS without a license or with a single value, edit the map (draw points, lines and polygons), developed in WPF, support shapefiles and WMS). In terms of VMS as it is the type of the system, it will be presented the definition and the architecture. After, it was done a research with the objective to get some softwares and their functionalities. The research on GPS had the objective to present its definition, describe how it works, what components it has and each functionality. Besides that, it is also shown the architecture of it. For shapefiles it was done a research with the objective to describe its functionalities, how this type of files can be read and also a technical explanation for each file that is necessary to obtain the final file. Finally, for WMS it was done research in terms of describing of how it works, to show the architecture of it and explaining it.

2.1. GIS
A Geographic Information System is a conceptualized framework that provides the ability to capture and analyse spatial and geographic data. Furthermore, GIS operates on many levels, being on the most basic level, geographic information systems technology is used as computer cartography, that is for straightforward map making. The real power of GIS, however, is through using spatial and statistical methods to analyze attribute and geographic information. The end result of the analysis can be derivative information, interpolated information or prioritized information. GIS mapping produces visualizations of geospatial information. The four main ideas of GIS are: create geographic data, manage it in a database, analyze and find patterns, visualize it on a map. These main ideas can be shown in the Figure 1.

![Figure 1: Architecture of GIS.](image)

The first main idea create geographic data is shown in the Step 1 when the data from the real data is converted to raw data with the objective to be able to be stored and managed it in a database (second main idea) which is shown in the Step 2. For the third main idea (analyze and find patterns) the data from Step 2 is manipulated according the user’s necessities. So, the forth main idea (visualize it on a map) the data manipulated on Step 3 is shown as a output data on Step 4. As conclusion of the observation of the figure, the main components of a GIS are: maps - these are the geographic container for the data layers and analytics a person wants to work with -, data - GIS integrates many different kinds of data layers using spatial location. Most data has a geographic component. After these research, it was necessary to carry out another research to choose the most appropriate GIS. Thus, the GIS found were: TatukGIS 10, GMaps .NET, SharpMap, DotSpatial, Bing Maps, ThinkGeo UI, topoXpress, ArcGIS and QGIS. After the previous research, it was necessary to choose a GIS that was the best for the development of the application. For this, it was necessary to take into account the company’s criteria (accessible API, GIS without a license or with a single value, edit the map (draw points, lines and polygons), developed in WPF, support shapefiles and WMS) and the advantages and disadvantages of GIS. These process will be presented in Chapter 3.

2.2. WPF
This project is motivated to improve a desktop application that the company already had. The application was developed in WinForms but it will be changed to WPF as being one of the company’s criteria. Windows Presentation Foundation (WPF) is a UI framework that creates desktop client applications and also it is Microsoft’s latest approach to a GUI framework, used with the .NET framework. The WPF development platform supports a broad set of application development features, including an application model, resources, controls, graphics, layout, data binding, documents, and security. The framework is part of .NET and uses the Extensible Application Markup Language (XAML) to provide a declarative model for application programming. When developing an application in WPF, there are two files that are generated: .xaml (where the view is developed) and the .xaml.cs (where the logic is developed). As said previously, the company had a desktop application that was developed in WinForms but it was changed for WPF in order to improve the application.

2.2.1 WPF vs WinForms
The single most important difference between WinForms and WPF is the fact that while WinForms is
simply a layer on top of the standard Windows controls (e.g. a TextBox), WPF is built from scratch and doesn’t rely on standard Windows controls in almost all situations. WPF is more suitable for the development of the application since it has one feature (data binding) that allows to accomplish the company’s criteria.

2.3. VMS

The Vessel Monitoring System (VMS) is a general term to describe systems that are used in commercial fishing to allow environmental and fisheries regulatory organizations to track and monitor the activities of fishing vessels. Moreover, it is defined as a communication system used to monitor, control, and survey fishing activities. The data is collected by fishing authorities and it is made sure that the vessels are following all international fishing regulations and are within fishing limits. So, in the Figure 2 represents the architecture of a VMS.

2.4. GPS

The Global Positioning System (GPS) is a space-based radio-navigation system consisting of a constellation of satellites broadcasting navigation signals and a network of ground stations and satellite control stations used for monitoring and control. Moreover, it is a global navigation satellite system that provides location, velocity and time synchronization, also it is navigation system using satellites, a receiver and algorithms to synchronize location, velocity and time data for air, sea and land travel. The Figure 3 shows the architecture of GPS and the elements used to make GPS works.

![Figure 2: Architecture of VMS.](image)

![Figure 3: Architecture of GPS.](image)

As shown in the previous figure, the information transmitted from the satellites can be interpreted by receivers to precisely identify locations on earth by measuring distances from the satellites. GPS has three elements: space segment (satellites) - the satellites orbiting the Earth, transmitting signals to users on geographical position and time of day -, control segment - monitors the health and position of the satellites in the space segment and transmits correction information back up to the satellites -, and user segment - GPS receivers and transmitters, which means, handheld or other receivers to interpret the messages broadcast from the satellites. While there’s only need three satellites to produce a location on earth’s surface, a fourth satellite is often used to validate the information from the other three. This last satellite allows to calculate the altitude of a device. GPS works through a technique called trilateration, which is used to calculate location, velocity and elevation, trilateration collects signals from satellites to output location information.

2.4.1 Trilateration

Trilateration is a mathematical technique used by a global positioning system (GPS) device to determine user position, speed, and elevation. By constantly receiving and analyzing radio signals from
multiple GPS satellites and applying the geometry of circles, spheres, and triangles, a GPS device can calculate the precise distance or range to each satellite being tracked. All the GPS devices require three satellites for an accurate calculation of position. Data from a fourth satellite — or even more than four satellites — further enhance the precision of the point’s location, and also allows factors such as elevation or, in the case of aircraft, altitude to also be calculated. GPS receivers routinely track four to seven satellites simultaneously and use trilateration to analyze the information.

2.5. Shapefiles
A shapefile is a vector data file format commonly used for geospatial analysis that stores the location, shape, and attributes of geographic features. It is stored in a set of related files and contains one feature class. Shapefiles are binary files that store data as points, lines, or polygons. These three feature types form the basis of geospatial vector data analysis. All shapefile files must have the same name but must have different extensions. There are three required files that, at a minimum, make up a shapefile. These three minimum files are: .shp (shape format) - main file (mandatory), direct access, variable-record-length file in which each record describes a shape with a list of its vertices (the feature geometry itself) -, .shx (shape index format) - positional index of the feature geometry to allow seeking forwards and backwards quickly; in this file, each record contains the offset of the corresponding main file record from the beginning of the main file and contains a 100-byte header followed by 8-byte, fixed-length records -, and .dbf (dBase IV format) - attribute format; columnar attributes for each shape, in dBase IV format. In this reference [4], a technical description of ESRI Shapefiles can be found where describes the shapefile technical, the organization of the main file and as well as record contents, organization of the index file and the organization of the dBASE file.

2.5.1 ESRI
Environmental Systems Research Institute (ESRI) is the world’s leading developer of geographic information systems (GIS) software, including programs that plot ZIP codes and addresses, demographic information and detailed, color-coded data. ESRI has several different lines of software that organizations benefit from. Their flagship product is ArcGIS Pro. This is an extremely powerful mapping tool that allows people to create maps, analyze data, compile and share geographic information, and more. This tool operates on desktop, server, and mobile platforms.

2.6. WMS
A Web Map Service (WMS) is a computer program that produces maps of spatially referenced data dynamically from geographic information. It is a simple interface for web based mapping applications. It defines an interface that allows a client to get maps of geospatial data and gain detailed information on specific features shown on the map. A “map” is defined here as a visual representation of geospatial data, not the geospatial data itself. A Web Map Service can: produce a map – as a picture, as a series of graphical elements, or as a packaged set of geographic feature data -, and answer basic queries about the content of a map; and Tell a client what maps it can produce and which of those can be queried further. The Figure 4 represents the system architecture for web mapping.

As shown in the figure, on the server side there is a Web server, Web map server, database, and files. A Web server represents a computer on which the appropriate software is installed with query interpretation capabilities sent via HTTP (but also HTTPS, FTP, FTPS, SMTP, etc.). Web map server represent a specialized software application comprising part of the Web server model and having an implemented functionality to receive and respond to requests for geo information. Web map clients represent libraries (typically in JavaScript) which enable visualization, manipulation, and analysis of geospatial data relying on Web standards.

3. Implementation
This chapter presents the GIS chosen, the respective reason for it and also it is described the general functional architecture of the application. It also presents a small overview of a WPF application before taking a look at the code developed for the achievement of some of the company’s criteria (edit the map (draw points, lines and polygons), support shapefiles and WMS) and for some extras features developed in order to improve the application. In addition, an explanation of the displayed code is presented. It was taken on account the inputs that
the application could have, one is being a shapefile or no input (the data’s information is done on the map) for the development of the application.

3.1. GIS
After the research on GIS, it was necessary to choose the best one for the development of the application. For this, it was necessary to take into account the company’s requests and the advantages and disadvantages of every GIS found during the research about GIS and the existent softwares. The company’s criteria were: accessible API, which means an application that is user-friendly; if the GIS had a license, the company didn’t want to pay it annually but only an unique value, that it is, only paid once; it should be possible to edit the map, more specifically draw points, lines and polygons on it; the GIS should be support WPF, WMS and shapefiles. Thus, taking into account the criteria presented above, it was easy to remove some GIS. In the next figure, it is possible to present them and the respective reasons for their elimination or not.

<table>
<thead>
<tr>
<th>GIS</th>
<th>Reasons for the elimination</th>
</tr>
</thead>
<tbody>
<tr>
<td>TatukGIS10</td>
<td>Does not have a license and supports WPF, WMS and shapefiles. The only main down side is to</td>
</tr>
<tr>
<td></td>
<td>have no access to performance and usability, so it is a poor fit for the application</td>
</tr>
<tr>
<td>OGIS</td>
<td>the GIS does not meet the minimum requirements of the company, so it is not easy to use for the application</td>
</tr>
<tr>
<td>SharpGIS</td>
<td>it has a complex API and it needs many plugins and so the performance is not fantastic, so</td>
</tr>
<tr>
<td></td>
<td>it is eliminated from being used for the application</td>
</tr>
<tr>
<td>Bing Maps</td>
<td>it has a slow pan and zoom with many polygons, so it is not easy to use for the application</td>
</tr>
<tr>
<td>Thinlits</td>
<td>for the license, the company didn’t want to pay it annually but only an unique value, that it is, only paid once, so it is eliminated from being used for the application</td>
</tr>
<tr>
<td>AMSIS</td>
<td>it has an unusual license fee. It is eliminated from being used for the application</td>
</tr>
<tr>
<td>GIS</td>
<td>it is eliminated from being used for the application</td>
</tr>
</tbody>
</table>

Figure 5: Elimination or not of GIS and the respective reasons.

By looking at the table, the GIS that meet the criteria are TatukGIS10 and Bing Maps. For the process to choose the best one, it was shown both GIS to the company and it was decided to use Bing Maps because the pan and zoom was less slower than the ones from TatukGIS10. After choosing the GIS, which is the main component of the application, it was possible to make a representation of the general functional architecture of the application, show in the Figure 6.

By looking at the figure, the application will have as GIS the Bing Maps which will have 3 servers: tracking, database and web. Tracking server will receive the coordinates of the vessels through the ground station; database server is where the data is stored and where it will be processed; web server is where the data will be presented. These servers are in synchronization and the application can be implemented as a mobile application or a desktop application. The project application will be a desktop application. After choosing the GIS and having the general functional architecture of the application, it was possible starting developing it.

3.2. WPF with Bing Maps
The feature of WPF, data binding, is important for the development of the application, since it will be developed extra functions that the user needs to view and edit data. After the data is accessed and loaded into an application’s managed objects, the hard work for WPF applications begins. Essentially, this involves two things: copying the data from the managed objects into controls, where the data can be displayed and edited and ensuring that changes made to data by using controls are copied back to the managed objects. So, after choosing the GIS, it was necessary to find how to use it with WPF, since the application will be developed in it. In order to use the Bing Maps WPF Control, a Bing Maps Key is needed to authenticate the application. The process to obtain Bing Maps Keys can be found in the reference [5].

3.3. Mandatory Functions
After acquiring the necessary credentials for Bing Maps Control and introduce those in the Map control presented in the list ??, it was possible to start the design of the menu toolbar taking into account the functions that the application needed to execute, imposed by the company (draw points, lines, polygons, read shapefiles and support WMS layers) plus seven more (color line/border, fill color, thickness, hand, ellipse, undo and clear). In the following subsections it is presented the steps, the code and the respective justification for drawing points, polygons and lines. Each of these functions will have a color line/border, a fill color and a thickness attributes that can be set by the user in order to draw different styles of the same shapes. These attributes are simple select options which will be applied on drawing by the respective function. The description of the extra functions from the toolbar will be presented after the presentation of the mandatory functions.
3.3.1 Points

When drawing points (pushpin) the events MouseLeftButtonDown and MouseLeftButtonUp are relevant. Upon the first event the coordinates from the position where it happened are stored. Then whenever the second event occurs it will create a Pushpin UIElement and add it to the map on the stored position. The position’s information is contained in the event object that is passed to the listeners. Although one must convert the coordinates of the position in the application’s window to the location in the map using ViewportPointToLocation function that is provided by the map control instance.

3.3.2 Polygons

When drawing polygons the events MouseLeftButtonDown, MouseMove, MouseLeftButtonUp and MouseRightButtonDown are relevant. Drawing polygons consists of a series of polylines. This means that the user will click on the map to make the vertices for the polygon so the cycle of the mouse events is going to be left button down, move, left button up and when the user wants to finish he presses right button down. The first event will store the coordinates of the position clicked on the map. The second event will draw and redraw the current polyline while the user moves the mouse. The third event will persist the previous polyline and update the start position in order to draw the next polyline. Lastly when the user finishes he will press the right button which will trigger the last event and stop the drawing of the polygon.

3.3.3 Lines

When drawing lines, the function that is triggered is the polygon function due to the error presented in the reference [6]. So the line will be a specific scenario of a polygon which will be a single polyline.

3.3.4 Shapefiles

The application uses a library named Catfood.Shapefile [7] which provides a ESRI Shapefile Reader. Shapefiles consists of three binary files (.shp, .shx and .dbf) that store data as points, lines, or polygons. Having this, whenever a user wants to load information from a shapefile into the map he uses the shapefile button which will prompt the user for the respective files. Then the file names are supplied to the library function which will return a reader that can be used to obtain each of the shapes information contain in the files. This information is then used to create the UI Element shapes that will be inserted in the map. In order to create the shapes one must check which type of shape is returned by the library. For points it is drawn a Pushpin in the map. For lines and polylines it is used the MapPolyline shape. Finally for the polygons it is used the MapPolygon shape. This way the code developed simply maps the shapes read from the files by the library into UI shapes.

3.3.5 WMS

In order for using a WMS layer in the application, one must implement a TileSource class that provides a way to get URIs for the requests which will be made by the infrastructure to obtain the needed tiles for the current view of the map, based on the XY coordinates and a zoom. After converting the XY coordinates and the zoom to latitude and longitude values for the tiles BBox the implementation will then produce the requests URI and the infrastructure will obtain those tiles.

3.4. Extras Functions

After presenting the steps, code and the respective justification for drawing points, polygons and lines, it will be presented the extra functions from the menu toolbar. Before stepping into each, one must understand how those are handled. To implement the functions it is needed that the application listen for mouse events because it is with those events that each function can be done. The available mouse events are: MouseLeftButtonDown and MouseRightButtonDown - these events are triggered when the left or right button, respectively, of the mouse is pressed down in the context of the application’s window - , MouseLeftButtonUp - this event is triggered when the left button while pressed is released in the context of the application’s window - , MouseMove - this event is triggered whenever the mouse moves in the context of the application’s window, whether pressed or not. Only one function should be ”active” at each time so it was created an enumeration MyShape that contains all the types for the available functions in order for the respective mouse events to be handled in different ways according to the selected function.

3.4.1 Color line/border

This function is only available for lines, polygons and ellipses since for the pushpins it will not be necessary because the objective of this is to change the color of the line/border. To do so, the user needs to click it and change for the color he wishes for and draw for the shape chosen. To edit it, after drawing, it will be explained later since there will be a button to allow the user edit this feature.
3.4.2 Fill color

Like the function color line/border, it will be only available for polygons and ellipses since for the pushpins and lines it will not be necessary because the objective of this is to change the color of the polygons’ and ellipses’ fill. To do so, the user needs to click it and change for the color he wishes for and draw for the shape chosen. To edit it, after drawing, it will be explained later since there will be a button to allow the user edit this feature.

3.4.3 Thickness

This function is only available for lines, polygons and ellipses since for the pushpins it will not be necessary because the objective of this is to change the thickness of the shape. To do so, the user needs to click it and change for the number of thickness he wishes for and draw for the shape chosen. To edit it, after drawing, it will be explained later since there will be a button to allow the user edit this feature.

3.4.4 Hand

The hand function will allow the user to navigate the map without editing it. This means that after drawing on the map and if the user wants to navigate towards another point of the map without editing it, he can by clicking on the hand function.

3.4.5 Ellipses

When drawing ellipses the events MouseLeftButtonDown, MouseMove and MouseLeftButtonUp are relevant. The first event will store the coordinates of the position clicked on the map which will mark the center of the ellipses. The second event will draw and redraw the ellipse while the user moves the mouse. Then whenever the third event occurs it will create a DrawEllipse UIElement and add it to the map on the stored position. The position’s information is contained in the event object that is passed to the listeners. Although one must convert the coordinates of the position in the application’s window to the location in the map using ViewportPointToLocation function that is provided by the map control instance.

3.4.6 Undo

The goal of the undo function is to rewind an action that the user did and all the properties related to it.

3.4.7 Clear

The clear function is to clear all the shapes that were drawn into the map and it will also clear the properties of all the shapes drawn on the map.

Besides the functions represented in the menu toolbar, it was developed some extras functions which are available for each shape on the map that complements the ones already created. In order to implement these functions for each of the current shapes in the map one must had to have a list of the shapes inserted in the map. For that it was added a list view where each entry represents a shape on the map. The extra functions are available in each entry of this list and are then presented. These extras functions use data binding to bind the respective data to each view.

3.4.8 Hide

The aim of this function is to hide a certain shape that the user wants. It becomes useful when there are a lot of shapes and the user wants to hide the ones that are irrelevant for him. This is implemented via a button which toggles if the shape is hidden or not.

3.4.9 Name of the element

This function only shows the name of the element (shape) created by the user into the map.

3.4.10 Center On Item

The objective of this function is to just center the map in a certain shape that the user wants.

3.4.11 Edit

This functions allows the user to edit the color line/border, fill color and the thickness of the respective shape selected. This is implemented via a button which toggles if the shape is being edited or not.

3.4.12 Description

This function allows the user to add a description to the selected shape and it is shown on the map. This is implemented via a button which toggles if the shape will have a description or not and prompt the user for a textual description). When the user saves the description it will write it into the center of the shape on the map.

3.4.13 Remove

This function is to remove the target shape that was created by the user and all the properties related to it.
4. Results

In this chapter is presented the results of the tests done with the purpose to check if the implemented code is correct and if the functions implemented can meet the company’s criteria. It was also described the steps taken until the results of the implemented functions were shown. It is also points what could be done better in terms of the implementation of the application.

4.1. Problem Description

The purpose of the project is to improve a desktop application that the company XSealence has. To this end, a .NET application with the objective to monitor vessels via satellite in GIS will be developed. It is also fundamental that the application is developed IN WPF with GIS being the main component. The company’s criteria for the application it was to be possible to draw points, lines, polygons on the map and also to be able to support shapefiles and WMS.

4.2. Baseline Solution

For the next subsections it is shown the results for the functions implemented that were presented in the Chapter 3. The functions of color/line border, fill color, thickness, undo, clear and extra functions will be tested in each shape and will be shown. However, functions such as the undo, clear, hand and center on item won’t be shown here since it will be easier to demonstrate them while presenting the thesis.

4.3. Points (Pushpins)

The number of pushpins drawn on the map should be the same as shown in the list view. The button “Edit” on the list view, should allow the user to choose a different option for color/line border, fill color and thickness, however, for the pushpins the features is not available. When the user click on ”Description” it shows a text box with a button next to the button of ”Clear” in the menu toolbar that allows the user to save the changes. The user needs to click in the text box to write the description and then click the button ”Save”. The description for each pushpin will be ”Element” plus the number of each one of them, moreover, the first pushpin will be ”Element 1”, the second will be ”Element 2” and the third will be ”Element 3”. After adding the description for the lines, it was tested if the buttons ”Hide” and ”Remove” were working with this shape. So, in this scenario, the turquoise ellipse will be removed and the ellipse that has the lime border will be hidden.

4.3.1 Lines

As in pushpins, the number of lines drawn on the map should be the same as shown in the list view. Do not forget to draw a line is necessary to active the button ”Polygon” on the menu toolbar. It will be drawn two lines on the map, the first line will have a blue color line/border and a thickness with a value of two, the second line will be brown and will have a thickness of ten. The lines won’t have a fill so it won’t be necessary to choose a color for it. After clicking in the ”Edit” button, the blue line will change only the thickness to 4 and the brown line will change the color for gold. Then, the description for each line will be ”Element” plus the number of each one of them, moreover, the first line will be ”Element 1” and the second will be ”Element 2”. After adding the description for the lines, it was tested if the buttons ”Hide” and ”Remove” were working with this shape. So, in this scenario, the blue line will be hidden and the gold line will be removed from the map.

4.3.2 Ellipses

As in pushpins and lines, the number of lines drawn on the map should be the same as shown in the list view. It will be drawn two ellipses on the map. The first will have a blue border, a pink fill and a thickness of one. The second ellipse will have the same color border and thickness as ellipse one but will not have a fill color, it will be transparent. After clicking in the ”Edit” button, the pink ellipse will change the color of border to yellow and the fill color to turquoise, the blue ellipse will change the border color to magenta and the thickness will change to fourteen. Then, the description for each ellipse will be ”Element” plus the number of each one of them, moreover, the first ellipse will be ”Element 1” and the second will be ”Element 2”. After adding the description for the lines, it was tested if the buttons ”Hide” and ”Remove” were working with this shape. So, in this scenario, the turquoise ellipse will be removed and the the ellipse that has the lime border will be hidden.

4.3.3 Polygons

As in pushpins, lines and ellipses the number of lines drawn on the map should be the same as shown in the list view. It will be drawn two polygons on the map, the first will be a triangle that will have a border with hot pink color, a white fill color and a thickness of three. The second polygon will be a hexagon with an orange border color, aqua as fill color and a thickness of eleven. After clicking in the ”Edit” button, the triangle will have a chocolate color for
fill and the hexagon will be transparent. Then, the description for each polygons will be "Element" plus the number of each one of them, moreover, the first polygons will be "Element 1" and the second will be "Element 2". After adding the description for the lines, it was tested if the buttons "Hide" and "Remove" were working with this shape. So, in this scenario, the triangle will be hidden and the hexagon will be removed.

4.4. Shapefiles
When clicking on the "ShapeFiles" button, the file reader pops-up so that the user can select the three files. After selecting the files, on the map it will show the data from the shapefiles and also it will show the respective list views (Figure 7).

4.5. WMS
When clicking on the "WMS Layers" button it will toggle the WMS layer over the map according to the configured server (Figure 8). If the user wants to see the normal layers it is enough to click again on the button.

4.6. Enhanced Solution
After implementing the functions and testing them, the application met the company’s criteria. However, some functions could have some improvements. For example, the WMS function could be developed better in terms of letting the user input its own server URI. Also, the center on item (extra function) is not centring the shape correctly on the map instead it centers against the whole application’s window.

4.7. Feedback
When the development of the application finished, it was shown to the company to check if the application met the company’s criteria. After doing it, it was possible to determine that the application met the company’s criteria and had some extra functions that wasn’t required by the company. So, knowing the functions required by the customer, in this case the company, it was necessary to determine if the extra functions developed were practical for the user or if they could be disposable. With this objective, some colleagues were asked to test the application and if possible give feedback on whether the extra functions were indispensable or not. In addition, the opinion was also asked that the API was accessible or too complicated. After testing the application with several colleagues, it is possible to conclude that the extra functions help the user to edit the map so that it is possible to navigate it without losing useful and essential information and the API was very accessible and that it was user-friendly.

5. Conclusions
In this chapter, reflection is made on a general conclusion of the entire project carried out. This project is motivated to improve a Desktop application that the company XSealence has. To this end, a .NET application with the objective to monitor vessels via satellite in GIS will be developed. It is also fundamental that the application is developed in Windows Presentation Foundation (WPF) with the main component being a GIS, where the positions of vessels received via satellite in GIS will be represented. Finally, the GIS must support local maps or via Web Map Service (WMS), raster and WMS formats. First, it was carried out a previous research on GIS and consequently it was possible to find out several GIS. After knowing the advantages and disadvantages of each one of them and also in mind the ones that would meet the criteria from the company it was chosen Bing Maps. Then, it was realizes a brief research on GPS, as one of the inputs of the application being a shapefile it was also needed to do a research on it and one one WMS.
Afterwards these research, the implementation of the application itself began and it was developed the code for the functions imposed by the company (drawing points, lines, polygons, support shapefiles and WMS) and extras to complement the others functions. Lastly, it was presented the results of the implementation, moreover, testing if the functions were working and what functions could have some improvements.

5.1. Achievements
This thesis allowed me to learn that the GIS used by the company was not the more suitable for what the company’s wanted, for example, when drawing a lot of polygons the pan, zoom and animations would have a low performance and would impact on the application. Besides that, WinForms was also not the most suitable because it is simply a layer on top of the standard Windows controls so when the application needed to be developed in WPF being one of the company’s criteria was the best solution since WPF is built from scratch and doesn’t rely on standard Windows controls. A great example of this is a button with an image and text on it. This is not a standard Windows control, so WinForms doesn’t offer the user this possibility out of the box. Instead the user will have to draw the image himself, implement the user own button that supports images or use a 3rd party control. With WPF, a button can contain anything because it’s essentially a border with content and various states (e.g. untouched, hovered, pressed). After developing the application, I learnt that sometimes what the client (in this case the company) asks is not what the user needs. For example, the company only needed functions to draw points, line, polygons and functions to show shapefiles and WMS but for the user it needed more functions. So, it was developed extra functions to turn the application’s API user-friendly and to turn the user’s experience better.

5.2. Future Work
For a future work, could improve some functions. For example, the WMS function could be developed better in terms of letting the user input its own server URI. Also, the center on item (extra function) is not centring the shape correctly on the map instead it centers against the whole application’s window. Could be implemented more functions that the application could do. For example, save the data that was done on the map, if the shapefile or WMS was edited the user was able to save it as well. Further, the window of the application could be a window responsive, which means, the window can render well on a variety of devices and window or screen sizes from minimum to maximum display size. The size of the list view could be reduced so that it won’t occupied half of the application’s window.

References