

Interview Project Document

- Harry Marsh

Hire car Fleet Management Application

Abstract

This documentation outlines and discusses the skills, methods and understanding required to complete the project of a c# application. The application will have the ability to display a map and icons that represent different vehicles that are being managed by a vehicle hire company. The application will have the ability to display information about the vehicles as well as the customers who are hiring them.

Table of Contents

Abstract	1
list of Assumptions	3
Project Design.....	3
User Interface design	3
Class Diagram.....	4
Project flow diagram	4
Icon design choice.....	4
Task/ time estimations	6
Bugs and issues	6
Map does not change size when resizing window	6
Icons hard to see against map background.	7
Tooltip box after hovering over marker has no format unable to make bold or underline	7
Car icon is not aligned to the centre of the marker	7
Unable to zoom when hovering over tooltip.	7
Icons would not load unless mouse scrolled.....	9
Designer drag and drop window not displaying	9
Test plan.....	9
Cyber/data security considerations.....	10

Design plan review	11
Task/ time estimations Review	14
External tools / research material / lessons learned	15

list of Assumptions

For the application to reach the requirement of the assessment, the below assumptions could be made;

Render a map on a window

For the user to be able to see the location of a vehicle within an application a map will be required. The map needs to be displayed on the window of the application and be large enough for the user to utilise it effectively.

Give ability for user to manipulate map

This added functionality will give the user a better ability to view and locate vehicles on the map. With the ability to zoom, the user will be able to see specific towns and streets that the vehicles may be located. The ability to move the map around will allow the user to search for vehicles on the map.

Place a point on the map to indicate location of vehicle

For the user to be able to look for a vehicle they will need to be displayed on the map. The Points on the map will need to be at the precise location of the vehicles so that the user is able to identify where the vehicle is located.

Use icons to identify vehicle type (Small car, Estate car, Van)

Adding specific icons to each point will allow the user to easily identify the type of vehicle in that location. The icons used should be simple but obvious to the user. Using an image of the vehicle type itself may be the best way to identify the vehicle.

Allow the ability to click on the point to give information about the vehicle's attributes

At this point the user is able to see all the vehicles on the map but is unable to identify the details of the vehicle or the customer who is currently hiring the vehicle. Adding the ability to click on the point and having a pop-up box show on screen, will give the user the ability to view the details of the vehicle within the pop-up.

Add ability to locate vehicle point by entering key details (Registration, Client Name)

When the application has many vehicles displayed it will be harder to locate specific vehicles required. By adding the ability to search based on key details such as the vehicles registration number or the clients name, the user will be able to reduce the number of points on the map to ones that fit the search criteria.

Project Design

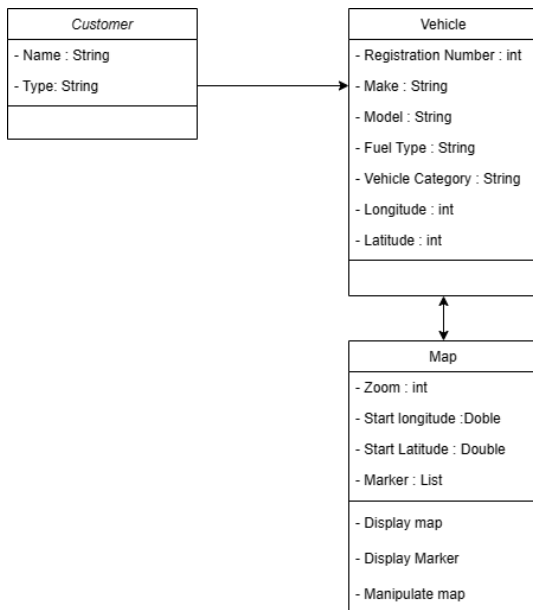
User Interface design

The screenshot displays the 'Main Application View' of a Hire Car Management Application. On the left is a sidebar with search filters: 'Vehicle Registration' (Input Box), 'Vehicle Make' (Input Box), 'Vehicle Model' (Input Box), 'Vehicle Fuel type' (Input Box), 'Vehicle Category' (a list with 'Small Car', 'Estate Car', and 'Van'), 'Client Name' (Input Box), and 'Search Button' and 'Clear Button' at the bottom. The main area shows a map with three vehicle icons: a van, a small car, and an estate car. A pop-up box titled 'Vehicle Details' is visible over the van icon, containing fields for 'Reg #', 'Make', 'Model', 'Fuel Type', 'Category', and 'Client Details Name'.

Based on the requirements provided on the brief the User Interface would be a good representation of what the final product may look like. The key components of the application include the map, individual icons based on the category of the vehicle as well as the ability to view the details of the vehicle when they are clicked on.

Interview Project Documentation – Hire Car Management Application
-Harry Marsh

Class Diagram

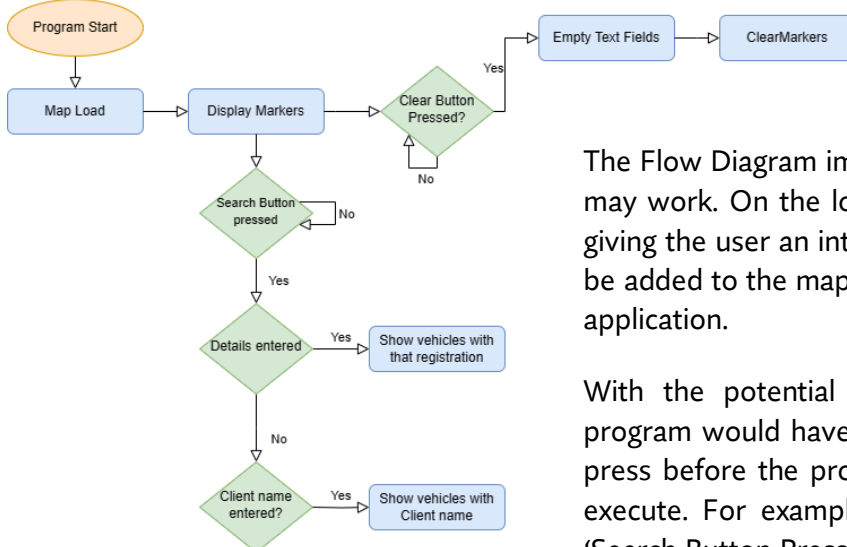


There will be 2 main class objects, the 'Customer' and the 'Vehicle'. These classes will hold the attributes that are required about the objects to be displayed to the user of the application. This includes the 'Registration Number', the 'Make', The 'Model', the 'Fuel Type' of the vehicle and what 'Category' the vehicle applies to, 'Small Car', 'Estate Car' and 'Van'.

The 'Map' Class includes all the attributes of the map object as well as its functionality. The class will need attributes such as zoom level' as well as the location of the map that it is looking at. This can be decided by the 'longitude' and 'latitude' attributes.

The class will have a collection of methods such as displaying the map image, displaying the markers of the vehicles on the map and will have the ability to be manipulated, for example to move the map around on the screen as well as zoom the map.

Project flow diagram



The Flow Diagram image shows how the logic of the application may work. On the load of the application the map should load giving the user an interface to look at. The markers should then be added to the map so that the user can see the vehicles in the application.

With the potential implementation of a search feature the program would have to wait for an event for example a button press before the program progresses onto the next method to execute. For example, in the image we have the question of 'Search Button Pressed?' if this case is true it will move onto the next function which in this case checks whether any data has been entered.

Icon design choice.

Based on the projects brief the user needs to be able to identify the category of a vehicle on the map without viewing the information pop ups. There are 3 different categories of vehicle 'Estate Car', 'Small Car' and a 'Van'. The images chosen to represent each of the category of vehicles is shown below.



These images were chosen due to their simplicity, allowing a user to quickly identify the category of vehicle. An additional benefit of these images is their style as they fit well with the style of common maps such as Google Maps.

Task/time estimations

The project brief states that “the task is to take no more than 4 working hours” with this in mind the table below shows an estimation of how much time should be spent on each component of the application based on how complex each task is to complete.

Tasks to be completed.	Estimated time (Minutes)	Reason behind estimated time.
Create User Interface	30	With the design idea already completed it should be straight forward to implement the design.
Add Map to application with controls implemented.	60	Adding a map to the application will most likely require a 3 rd party library so may take longer to research and implement the design into the application.
Create the vehicles and customers to be added to the map	30	This should involve the creation of 2 classes and creating objects with attribute to create the example vehicles.
Create markers with required icons for the location	45	This may take longer as it requires the ability to create markers with the vehicle's coordinates as well as have the ability to display custom icons based on its category.
Add ability to click on markers to display information of vehicles.	30	Will need to create a pop-up window to display all the information about the vehicle and client.
Add the ability to search for each marker based on its attributes	45	This will require all the vehicles to be sorted based on the data entered in the search. The markers that do not fit the search need to be hidden as well.

Bugs and issues

Below is a list and description of bugs that were present in the development of this project. The bugs have a description of the issues present and how they were found, the solution to the bugs if one was found and the result of the bug whether fix, unfixable or needed further attention to resolve.

Map does not change size when resizing window

Issue: When full screening the application, the map or any other components do not change size when the window is resized. All components stay in the same position when resizing the window and if resized over any components of the form they will disappear.

Solution: After following the Documentation and online forms, this issue can be resolved by setting up auto scaling on the form. And setting the lower limits of the window.

Result: Solution found, not Implemented yet.

Icons hard to see against map background.

Issue: One of the problems that was noticed almost instantly was that the Icons chosen do not stand out with Google maps colour theme and can be very difficult to spot.

Solution: With this in mind the icons had to be changed to icons with backgrounds to help easily spot them at a glance. The new icons are much easier to see on the maps the colours do not blend in with the background, The new icons are displayed below.



Result: Solved

Tooltip box after hovering over marker has no format unable to make bold or underline

Issue: When hovering over a Tooltip box there is no format to the text. The text was a different font to the road names and other text on the map making it seem out of place. The text was not very clear in the box and needed underlining.

Solution: After doing some research on the GMap.Net controller I found out that there is a small amount of formatting that could be done. This includes setting fonts, text paddings, background and text colours. Unfortunately, I was unable to find the ability to underline any of the text to help the data look clearer. To fix this issue I added a string of “-----” after section titles in the details box string, using the “\n” to create new lines.

Result: Partial solution

Car icon is not aligned to the centre of the marker

Issues: When viewing the map, it can be obvious that the marker icons do not perfectly centralise too the location of the vehicle. When hovering over the icon the detail box does show an arrow that indicates the exact location of the vehicle. This issue is most likely caused by the use of custom icons.

Solutions: To solve this problem I added an offset to the marker. The original marker was designed to be at the bottom of a push pin due to this the icons were being displayed as that potion. To fix this issue I did some research and found that if you divided the width and the height of the icon by -2 the icon would be central to the position of the marker.

Result: Solved

Unable to zoom when hovering over tooltip.

Issues: While the application is running, the users is unable to zoom in or out with scroll wheel while the mouse cursor is over a Vehicles details box.

Solutions: No solution was found to fix this bug. As the user can continue to zoom in and out once the vehicle details box is not under the cursor, the bug only causes a small inconvenience to the user. On the other hand, not having the ability to zoom while hovering over the details box could be seen as a benefit as it prevents accidental scrolling which would refresh the map and reload the markers. With this in mind, I decided to move on to better utilise the time I have on the project.

Result: Not solved.

Icons would not load unless mouse scrolled.

Issue: After implementing the use of lists to store both objects as well as the markers, the icons on the map would not load until the user zoomed in or out of the map.

Solution: This appeared to be due to the map controller waiting for an update before it added the markers. I struggled to find a solution to fix this bug. I tried refreshing the map controller and making it update after Every marker was created but this did not fix the issue. I decided to force the map to zoom in and out to render the icons. Although this is having not fixed the cause of the problem the outcome was effective and due to the speed, the program runs, the zooms are unnoticeable in the program. To fully fix this issue more research will need to be done into the GMap.net package.

Result: Temporary solution

Designer drag and drop window not displaying

Issue: At some point in the development process, I was unable to open the form designer page. When is double clicked on the .cs file the code would open instead of the form designer as it should. I found this issue after trying to make some modification to form for the search function.

Solution: The issue was due my object classes being outside the scope of the form class so when visual studio tried to read the code it would see the object class first. To fix this I placed the object classes inside the form class and this has fixed the issue that was preset. I am now able to access the form designer.

Result: solved

Test plan

The use of a test case system can be used to preform basic user testing. The test case works by identify the different features and functions that the application has to preform and manually interact with those features to see if any issues become present. The test case will give me the ability to document any issues and make comments to acknowledge and potentially solve the issues. Below is a Table of the user test cases that I performed on my program.

Case No	Case Name	Expected Result	Actual Result	Comment
1	Map Movement	Be able to Drag the map around with the left mouse press.	As intended	This is a base Feature of the GMaps.net package.
2	Map Zoom	When the scroll wheel is used the map should zoom in and out showing more or less local data based on zoom distance.	As intended	This is a base Feature of the GMaps.net package.
3	Icon positioning	The icon should be at the location of the vehicle	Icons are slightly offset to the actual markers position	Need to change image padding to fix this issue

4	Icon images	The icon for the marker should be in correct for the category of vehicle its displaying	As intended	Used an if statement to take data form the Vehicle. Category attribute.
5	Icon data accuracy	The data shown in the markers pop up should be relevant to the selected vehicle sand position.	As intended	The data is based on the object including locational data
6	Search Button Null Entries	When sear pressed and no entries are fill nothing happens	As intended	No Comment
7	Search Button Partial Entries	Show all vehicle markers that match the criteria even if null entries are different	As intended	If there is a spelling mistake no entries will be shown.
8	Search Button Full Entry	Show all vehicle markers that have the exact entry.	As Intended	If there is a spelling mistake no entries will be shown.
9	Clear Button	Empty all textboxes and redisplay all vehicle markers.	As Intended	No Comment
10	Window Resizing	Components should scale with the window being resized.	No components of the application changed when the window is resized.	Time restraints lead to this feature not being implemented.

The use of unit testing, the process of isolating individual parts of a program or units can be very beneficial if implemented at an early stage of programming. The unit testing allows the application code to be tested in section. The benefit of this is to ensure that the program continues to function as new features are added. Another benefit of using unit testing is that features of the program are designed and tested to work independent of each other allowing the option to use the code across other applications.

Cyber/data security considerations

General Data Protection Regulations (GDPR)

The application with real life implementations would be handling a collection of different data. It is crucial that is data is handled correctly, used lawfully and stored securely. These are some of the basic requirements for the data protection act 2018. There is stronger legal protection for personal data for example with the project could include the 'Customer' or even the 'Vehicles' Details as these contain names as well as locations. With this in mind there are a collection of methods that can be utilised to keep this data safe as well as used in its lawful and intended way.

Protecting data such as coordinates and personal details will protect and reduce the chances of malicious attacks to take place. For example, if the coordinates of a hire car vehicle where easy to get a hold of there would be a high risk of that vehicle being stolen.

Right of access controlled through the use of accounts and passwords.

One method that could be implanted to protect data is to moderate the data's access. With the implementation of a login system, the data should only be accessed by users with authorised accounts. In addition, it provides the additional ability to restrict availability depending on the access level provided to a user's account.

With the implementation of the login system another layer of protection to users' data can be added. This system can prevent unauthorised user for example a person with malicious intent, from physically accessing the data through a hire company's computer.

Data Integrity

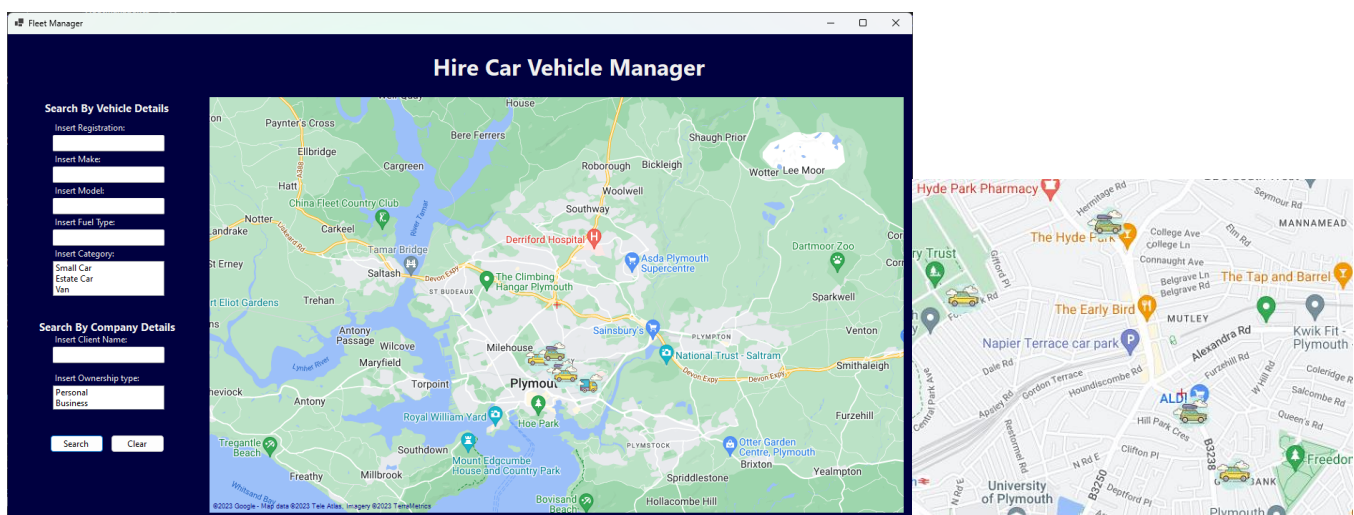
For the application to work as intended and to provide correct information the integrity of the data is crucial. Data can lose integrity in multiple different areas for example, Human error. Making a mistake when adding data to a dataset such as spelling mistakes or incorrect formatting can lead to the application not working correctly or showing the wrong information.

The program can be affected by malicious intent in many ways. In relation to the data's being used, it is important that the data is encrypted before transferring between the data source such as a database, to a user's device. Using this system on an insecure network and without any use of encryption could lead to the data being intercepted and stolen.

Design plan review

User interface Review

The design of the user interface did not differ much from the original design. The main differences to the design were the implementation of titles to help show what each part of the application does. In addition, I added more textboxes and another list box to allow the user to search for vehicles based on the customer's details. This provides additional functionality to the application and allows the user to view all of the vehicles that are owned by a company. Below is an image of the final application design.



The design could be improved by making changes to style of the application for example rounding the edges of the map to make it look sleeker. In addition, the icons could be made even more visible by adding a shadow to help elevate them visually above the map.

Functionality Review

For the application to reach the requirement of the assessment, the below assumptions were made;

- Render a map on a window
- Give ability for user to manipulate map
- Place a point on the map to indicate location of vehicle
- Use icons to identify vehicle type (Small car, Estate car, Van)
- Allow the ability to click on the point to give information about the vehicle's attributes
- Add ability to locate vehicle point by entering key details (Registration, Client Name)

With the implementation of the GMaps.Net package all aspects of the map such as rendering, the ability to move and manipulate the map and the ability to place markers on the map could be implemented. The controller was able to provide the base functionality of the map and gave me a platform to develop on. The map worked exactly as I assumed it would and exactly as I needed.

The implementation of custom markers on the map to identify the different vehicles also worked as per the original assumption

The process of view the details of the vehicles was performed very differently than what was assumed. The original assumption was that when a marker was clicked on a popup box would show with all the information related to that Vehicle. In the creation of the application a different method was used called a tooltip. This is a built-in option for the Gmaps.net markers allowed me to add custom text when hovering over a marker on the map. I decided to use this feature as it was already partially implemented so only needed a small amount of work. The system uses a mouse hover rather than a mouse click which frees up the opportunity to use the mouse click for a different functionality.

Future capabilities

Implementing the use of a database for large dataset

Within the project only a small amount of data entries is used to demonstrate the programs capabilities. With the deployment of a project like this it can be assumed that it may be used by all sizes of companies. With this in mind the implementation of storing data for the program in a database allows the application to expand having the ability to utilise a large quantity of organised data. In Addition, the data can be more secure in a database adding encryption and password locks.

Implement the ability to register new vehicles and new customers

In the scenario provided it could be assumed the company will want to add more vehicles to their fleet as well as add more customers. Adding a new form in the application to provide the user with the ability to create new entries allows the client to increase or decrease their datasets as they need it.

In addition, it would be beneficial to have the ability to assign customers to vehicles instead of having them preset. This allows the hire company to easily change which customer is using which vehicle giving the user the flexibility to manage and maintain their own fleet without directly editing a data set.

Task/ time estimations Review

Tasks to be completed.	Estimated time (Minutes)	Actual time (minutes)	Reason for time difference if any?
Create User Interface	30	30	The time taken to complete this task felt fairly accurate. As I had already come up with the initial design it was simple to follow and re-create. A bit of this time was taken making modifications to the layout and adding additional data entry fields.
Add Map to application with controls implemented.	60	45	The implementation of the map was much easier than I first anticipated. Following a Documentation, adding the map to the project was as simple as importing the GMap.net Package to the program and dragging the map controller onto the form. A small amount of time was taken to configure the map to have a max and min zoom level.
Create the vehicles and customers to be added to the map	30	60	<p>Creating the classes for the vehicles and customers was simple. It did take a good amount of time to figure out how to create objects with the classes and add values.</p> <p>I spent about 15 minutes researching how to create these objects and found a method called constructors. I go into more Detail about them in the lesson learned section.</p>
Create markers with required icons for the location	45	115	<p>The creation of markers required a lot of different components to work well. I had to do research about how markers and overlays work with the Gmap.net system.</p> <p>I then needed to figure out how to create multiple markers for all of the vehicles present. I started by adding all the markers manually but realised there would be no scalability for the application and was tedious to implement. This was resolved by using foreach statements to loop through the list of objects.</p> <p>Finally, I had to learn how to add custom marker icons. After doing the initial research I was tempted to use different colour pin icons and add a key to help the user identify each type. After doing more research I learn that you can create a custom marker using a bit map. This had issues as the marker was not positioned central to its location.</p>
Add ability to click on markers to display information of vehicles.	30	30	The original plan was to create a popup window with all the information on a click of the marker's icon. While setting up the map on the program I found out that the markers have a tooltip option which is a box that shows

			details of a marker on mouse hover. These tooltips needed to have formatting implemented so that they were more readable for a user.
Add the ability to search for each marker based on its attributes	45	90	<p>The original idea for this was to sort the list by the attributes entered in the text box. I found this method to be too complex and ended up implementing a loop that would compare each attribute's value of a vehicle with the entered details of the textboxes. I then needed to create a new list containing all of the matching values to be displayed on screen.</p> <p>I needed to add the ability for markers to hide if they were not a match. To do this I had to clear the map overlay and create new markers with only the matching vehicles.</p>
Finds and fix bugs within the application	N/A	90	<p>Within the program there were many bugs and issues. The issues can be found in the bugs and issues section.</p> <p>These bugs required a lot of research to figure out and implement with more time-consuming bugs such as the not updating until scrolling issue. Overall, I found the bug fixing to be more time consuming than most parts of the project.</p>

Overview

As can be seen from the time taken table above it has taken much longer to complete the application than originally planned. The results of the actual time taken are not perfectly accurate as I did not set any timers between tasks. Some of the tasks took longer to implement due to the amount of research required to set it up as well as time taken to follow the Documentation. In addition, the development process was not continuous spending time to work on this documentation as well as a lunch break.

External tools / research material / lessons learned

For the implementation of maps into the application I did some research and followed some Documentations that all utilised a NuGet Package Called GMap.NET. The package has a collection of features and abilities that are very beneficial for the project. The most important benefit was that it created a tool that could be dropped into the form from the visual studio toolbox. This tool handles the display, movement and zooming of the map when correctly configured.

When adding the markers to the map I wanted to store all the vehicles in an array so that I could create a foreach loop to generate the markers. Due to the limitation of an array I decided to use a list instead. With an array in C# the total amount of entries is fixed or has to be declared. By utilising lists you can more dynamically add entries, or in this project vehicles, giving the opportunity to have scalability in the program.

When creating each of the objects I was manually telling each attribute of the vehicle for example doing Vehicle.Make = "Ford";. Creating the objects this way took up a large number of lines of code. I decided to use constructors instead which is a more compressed way to generate new objects. By using the 'new' keyword and

defining the class I want to create I can use the constructor to create an object. For example, 'Car1 = new Vehicle("DE34 LMN", "Ford", "Focus", "Petrol", "Compact Car")'. The benefit of using this method is that it reduces the number of lines needed to create an object compared to my previous method, as well as gives me the ability to more easily add new objects into a list.