José Manuel Ossorio
Marco Antonio Pérez
Juan David Aguirre

Engineering method

Phase 1: Identifying the problem:

- Description of the problem context:
    - The automobile is one of the greatest inventions of humanity. In fact, it is thanks to automobiles and other means of transportation that our world is able to work today. People need to go places and using some sort of transportation saves a lot of time.

      However, when everyone wants to have their own car to go where they need to, it can become a problem. First off, today we are experiencing the effects of contamination and pollution. We have reached a point where we have to stop now and make a change; otherwise, there will be no planet for future generations. Automobiles are a big source of pollution, even though most modern cars have filters to reduce the waste they produce.

      On the other hand, cities are more crowded every day because more and more people have vehicles. This causes that using your car is not as fast as it used to be in past years. In fact, in some situations you might find yourself arriving sooner by foot than by car.

      In conclusion, we need to find a way to optimize the use of vehicles, since this will reduce pollution and traffic.
- Description of the problem:
    - There are too many cars in the city which is causing a lot of pollution and traffic
- Functional Requirements:
    - The software must:
        - R1: show all close by objects related to the solution (be it cars or stations)
        - R2: zoom in and out on the map
        - R3: let the user pick a location
        - R4: inform the user of the closest available vehicle to such location
        - R5: inform the user of the distance between the selected location and the closest available vehicle
- Non-functional requirements:
    - The software must:
        - Easily implemented
        - Easily developed
        - Relatively low cost

Phase 2: Compilation of necessary information:

While the document *Does sharing cars really reduce car use?* does not focus primarily on the environmental side of the situation at hand, the evidence indicates toward the idea of ride-shares as a means to reduce the ever growing car usage. The reduction of car usage would translate into a reduction of car-related gas emissions and overall more liveable cities.

We researched how to use the GMap.NET control in order to solve the problem through the use of this tool. We did a demo project which had some markers on the map, some routes and also a polygon. All this to get used to GMap.NET

Key elements in the solution of the problem

- Marker: a marker on the map
- Route: a line between 2 or more markers
- Distance: GMap.NET can calculate the distance of a route. This is key for our problem because we need to find the closest vehicle to a position
- Latitude and Longitude: these are a way to locate points on the map, which would be useful to mark the locations of the vehicles

Some ideas and projects have already been implemented in order to deal with this problem, some examples are:

“Cupos” groups on facebook: this was an idea of some university students to establish a place where people that offer room in their cars and people that needs transport can meet together, for the purpose of organizing the available space on the cars. Normally the people who offer the service post their routes and the interested students write on the post to secure their space on the trip.

“Voy con cupo”: is a government web page that use the same idea of “cupos” groups, make the people who have cars meet together with the ones that need transport. On the web page, as a driver, you can register your trip and the page search interested passengers on the specified route, if you are a passenger you have to log in and specify the beginning and destination and the page finds you a trip.

Phase 3: Creative solutions

We used the brainstorming method to come up with these ideas.
All solutions will be implemented in software form, all of these will allow:

1. the user to visualize a map of his or her surroundings which displays all of the vehicles close by, the service wouldn't be charged.
2. the user to visualize pre-established “car stops” where people willing to share their cars can pick others up.
3. users to share their car for a minimum fee.
4. the user to visualize checkpoints where authorities would enforce a country wide “car-share” policy in which people should always share their car so long there’s less than two people in it

Phase 4: Transition to preliminary design

Before moving forward into the analisis and deep dive into the selected options, some should be discarded. Option 2 won't be taken into further consideration as it would require considerably higher economical funding than any of the other solutions. While some carpooling-only stops could be a proper solution to the issue (that being in connection to incentivising carpooling itself), said stops could entail the government having to take certain lands from people for them to be built, not taking into consideration any expenses related to a considerably large amount of stops. On the other hand, option 4 won't be taken into further consideration either. While a law wouldn't necessarily necessitate physical infrastructure for it to work, forcing the population into sharing cars might not be in a country's best interests. While a considerable amount of people would be supporters of the law, some would consider this a violation of their privacy or opening the possibility to insecurity. Said dichotomy of thought could lead into bigger issues as any of the two position entail some, if ever so slightly, politically fueled followers.

Having discarded options 2 and 4, the remaining options shall be explored further.

1. Application without fees:

   For the user input:

   a. Have a field for the user to introduce the latitude and longitude of his current position
   b. Have a click listener for the user to click anywhere on the map and get that position's latitude and longitude
   c. Have fields for the user to introduce and address
   d. Implement a MouseHover method that would be indicating at all times what the closest vehicle to the current mouse's position on the map is

   For finding the closest vehicle:

   e. Go through all the markers (vehicles' position) comparing distances and finding the shortest one
   f. Make a graph with the user's location and all the other vehicles
   g. Make a reasonable area around the user's input location so that we don't have to go through all the possible markers, since some of them will clearly be too far and wouldn't need to be considered when finding the closest one
2. Application which uses fees:

   This application would be considerably similar to the other one proposed with the only difference being that it should implement the following:

   a. A payment system
   b. Strong security to manage the payment information

Phase 5: Evaluating and selecting the best solution

For the sake of better judgement into the evaluation of solutions, the following rubric was designed:

1. Development difficulty
   a. 1 - High difficulty
   b. 2 - Medium difficulty
   c. 3 - Low difficulty
2. Implementation difficulty
   a. 1 - High difficulty
   b. 2 - Medium difficulty

    c. 3 - Low difficulty
3. Governmental implications
    a. 1 - High implications [Such as regulations which could lead to illegality]
    b. 2 - Medium implications [Such as regulations as to how many people per car or traffic]
    c. 3 - Scarce implications [Involvement of the government in marketing situations]
    d. 4 - No implications

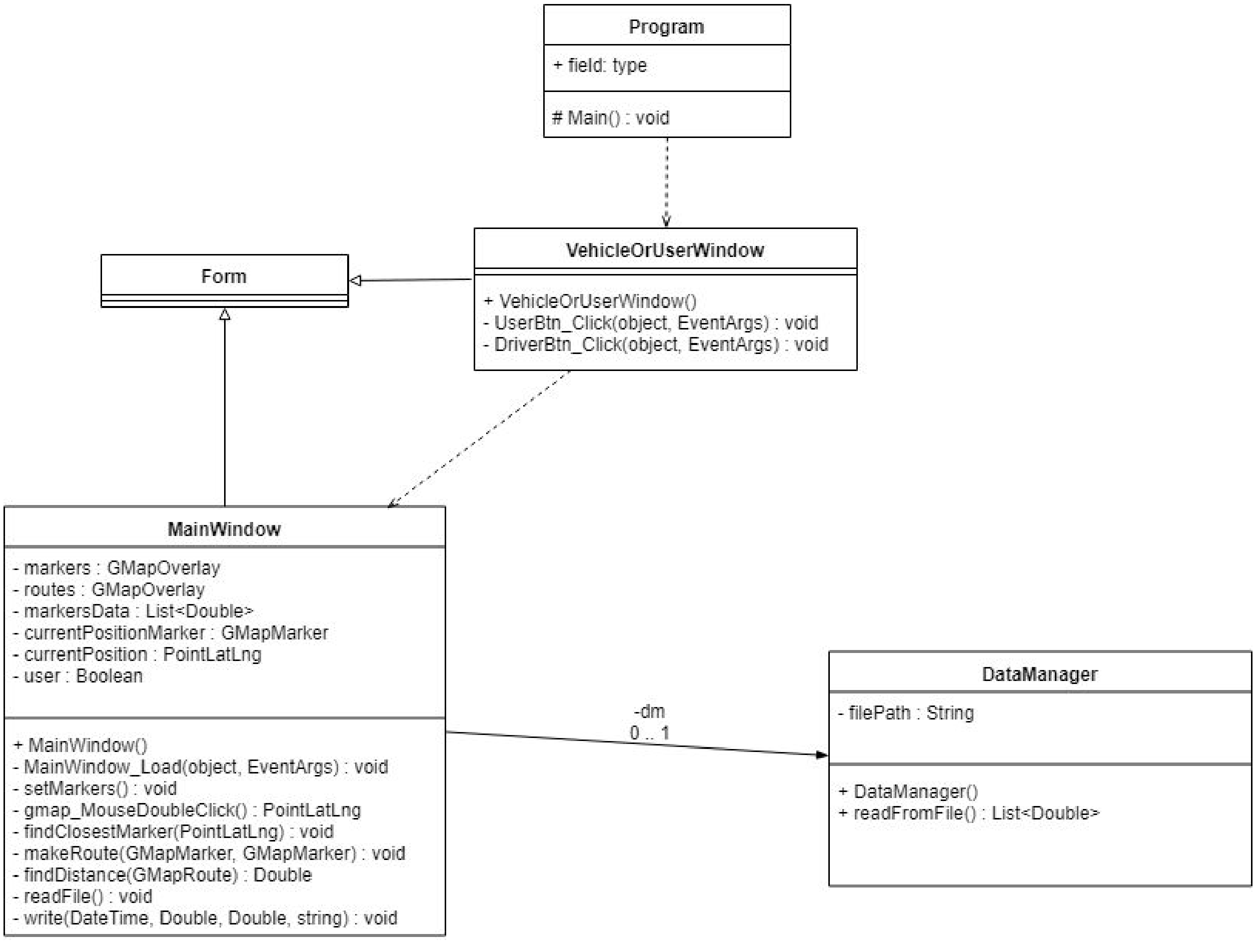
Evaluation:

Solution	**1**	**2**	**3**	**Total**
1	2	2	3	**7**
2	2	2	1	**5**

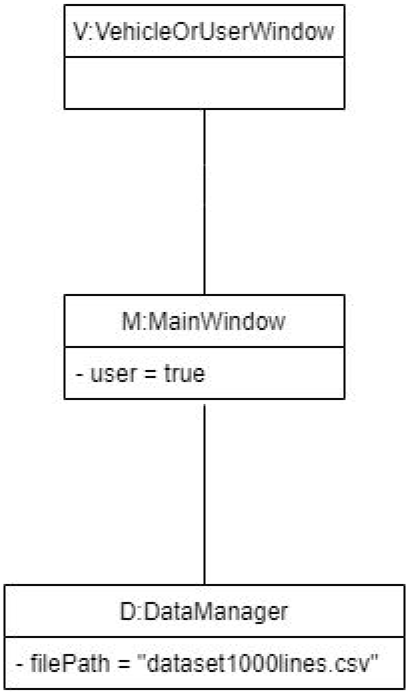
The results direct us towards the design of an application which does not require a fee. This result is heavily favored by solution 2's performance on item number three, this being governmental implications. Said point shall not be taken into lightly consideration by any proposer of a transportation service. Depending on the country, any application which requires users to pay any fee is considered as a service similar to that provided by cabs. This has lead to applications such as uber (which for all intents and purposes is substantially different to option 2) being banned in several countries around the world, this directs any proposer of similar apps to strive from governmental support. Either way, governmental support wouldn't necessarily translate into success as Taxi lobbyists are powerful entities in most countries and potentially have monetary means to oppose both the government and tech-companies. Taking all of this into account, solution 1 will be fully developed.

<u>Phase 6: Preparation of reports and specifications</u>

Class Diagram:


Program
+ field: type
# Main() : void
VehicleOrUserWindow
+ VehicleOrUserWindow()
- UserBtn_Click(object, EventArgs) : void
- DriverBtn_Click(object, EventArgs) : void
Form
MainWindow
- markers : GMapOverlay
- routes : GMapOverlay
- markersData : List<Double>
- currentPositionMarker : GMapMarker
- currentPosition : PointLatLng
- user : Boolean
+ MainWindow()
- MainWindow_Load(object, EventArgs) : void
- setMarkers() : void
- gmap_MouseDoubleClick() : PointLatLng
- findClosestMarker(PointLatLng) : void
- makeRoute(GMapMarker, GMapMarker) : void
- findDistance(GMapRoute) : Double
- readFile() : void
- write(DateTime, Double, Double, string) : void
-dm
0 .. 1
DataManager
- filePath : String
+ DataManager()
+ readFromFile() : List<Double>

Object Diagram:



Reflexive Synthesis

The engineering method, when taken with the amount of dedication it deserves, can lead into finding the best solution for any problem at hand one may have. This occurs if one properly follows the steps proposed by it. Contextualizing the issue at hand allows the team to have a better view on the conditions surrounding it and how some of the solutions to be proposed have potential to affect some related areas. Following this, and in unison with the compilation of information, proposing ideas and evaluating them becomes an smoother experience considering the previous steps provide a clear view on the situation and what is and isn't viable, taking most factors into account.

When it comes to Gmaps, there's one significant setback. While coding, the group noticed that computers whose native language is English will have an easier job of integrating with CSV-based information as the coordinates are natively stored with dots (.) in said systems. In contrast, Spanish based systems use commas (,) when referencing coordinates, this would cause conflict with the way the coordinates are stored into the files as CSV natively uses commas to separate data. While in this case, the solution is to be implemented mainly on Spanish based systems and proper corrections were made; if the solution was to be implemented in non Spanish based computers, other versions of the software would have to be developed.

Finally, it's crucial to point out that the solution will not work by itself. While this is purely an engineering work, centered around the development of a software solution, it becomes clear that no solution will be successful if there's no strong foundation surrounding it. Yet again, this should come as no surprise to the reader as no solution to the environmental impact of car use is easily achieved.

References

- Transport & Environment (2017). Does sharing cars really reduce car use? Obtained from: **https://www.transportenvironment.org/sites/te/files/publications/Does-sharing-cars-really-reduce-car-use-June%202017.pdf**
- International Transport Forum (2016). Shared Mobility: Innovation for liveable Cities. Obtained from: https://www.itf-oecd.org/sites/default/files/docs/shared-mobility-liveable-cities.pdf
- La Nación (2018) ¿En qué países Uber es ilegal? Obtained from: https://www.larepublica.net/noticia/en-que-paises-uber-es-ilegal
- Independent Software. GMap.NET Beginners Tutorial Adding Polygons and Routes to Your Map. Obtained from: **http://www.independent-software.com/gmap-net-beginners-tutorial-adding-polygons-and-routes-to-your-map-updated-for-vs-2015-and-gmap-net-1-7.html**