Social Contract and Effort

- **Your social contract Links to an external site. (Links to an external site.), i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project (this means, of course, you should create one in the first week and continuously update it when the need arrives) There is a survey (Links to an external site.) you can use for evaluating how the team is perceiving the process and if it is used by several teams it will also help you to assess if your team is following a general pattern or not.**

We have still not seen a use for significantly altering the social contract already established. Small changes that may be put in the social contract have been discussed but are more flexible week by week and therefore not something we found to belong in the social contract.

In the future we will begin evaluating our performance more formally as a group during the team reflections as well. This is to create a valuable KPI and to consolidate our opinion as a group on how the sprint went. The attendance of these reflection meetings to the best of our abilities will be put in the social contract.

- **The time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)**

Last week one user story took way too long to complete (26 hours compared to the 3.33 estimated time).This week however, even though we spent some extra time redesigning the structure of the list of cars and how it worked to more efficiently complete our user stories, the time spent does match much better with our expected time. This may be thanks to how we extended some estimated effort for some user stories as we now knew more accurately how much effort it would actually take given our experience with the last ones. We wish to continue this flow and to do that we will continue to update our user stories to achieve that.

Design decisions and product structure

- **How your design decisions (e.g., choice of APIs, architecture patterns, behavior) support customer value**

Not much has changed since we last brought up this topic during the team reflections. We have begun to focus more on speed of the code and the user stories surrounding that aspect. Some restructuring has been done to increase speed and this is always something we keep in mind especially when working with the list of vehicles that should be generated, filtered and sorted. Not having large loading times are standard for today's websites and having a website load for a minute straight would be quite concerning for the customer. We will keep this aspect in mind as we are working to complete the MVP.

- **Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)**

We currently use code documentation as our only means of documentation. We had communicated with the TA/Teacher about this and was informed that it was sufficient. We too believe this is enough and wish to continue working like this.

- **How you use and update your documentation throughout the sprints**

Well, currently we are in a situation where we have done a lot of documentation in the code we have committed every week (A). Everyone should continue making documentation for their own part of the user stories before committing in Github (B). As a group, we decided to make it an usual part of our work every time someone commits their work in GitHub(A->B). If any particular code part is difficult to understand, then we often talk about it and go through it together and try to help each other out to understand.

- **How you ensure code quality and enforce coding standards**

Currently, we have worked ourselves up compared to the first weeks. It feels that we have everything under control. All the unoptimized areas that we mentioned in the first week are now optimized and changed. For example css being in the index.html file are solved by dividing the html and css file to respective files. Or a whole temporary list in the controller is no longer necessary. By using the MVC model standard already from the start, we are enforced to use that structure, and a lot of the code quality follows along quite nicely by correctly applying MVC. Of course, sometimes it is hard to deliver a good code quality, if there is someone that thinks that the code is not of sufficient quality, then we are encouraged to communicate within the group and try to improve it.