This last week has been one of the more challenging ones, for more than one reason. Firstly this week was the week where I had the least amount of available time because of the workload of the other course (as well as personal reasons), resulting in me having to use the weekend to feel like i got enough work done.

The second reason was that this week had a focus on developing the back end (our server setup and the main routing algorithm). This made it harder to both estimate how much effort every user story would require, but also to set up good user-stories and tasks that followed the INVEST/SMART model.

A consequence of the tasks not being that independent was that it was hard to know exactly how to write code for the tasks that I was assigned as this required me to know what the other team members had done and how they had written their code. Combining this mentioned issue with the fact that I was unable to put time into this course during Tuesday and Wednesday made it so that I had to put in some hours on Thursday just trying to understand what had already been done, and how I could continue working from the existing product.

One thing that on the other hand helped and was a good way to better understand the interfaces between the different parts of our application was the UML-diagrams. So this is something I think can be worth focusing more on the next time. To make sure that we have a good structure, and that we define all the interactions that all different modules of our code require before we actually start writing the code. This way it would be easier to work independently on a task as it does not require as much synchronizing with the other team members all the time, which is useful for weeks like this, where I had a tight schedule.

For the next sprint I think it's important that we spend a bit more time on planning. This is so that we can actually create smaller, more independent tasks by taking time defining all the parameters and returns of all methods we think we might need so that every single method or class can be developed without understanding everything that all other team members do every second.

Despite this being a quite stressing week, and that our planning maybe wasn't as efficient I still think it was a fun week where we all developed. And as this was the week where I think we had the most issues, I also think that this week was the one that helped us learn the most also. As "failing" can sometimes be the best way to learn how to actually do things in a good way.