# Work report

Wrote some comments after every hour of work. **Total time 28 hours**(not counting breaks and note taking). The project wasn't easy but it was fun, I liked incorporating different systems into the project and learning how they worked like the sunrise-sunset.org api, the map and the graph. Also it's always nice to visually see something work. I was constantly making progress and didn't get stuck on anything for too long. When I did get stuck or found a problem that I couldn't solve then I highlighted it in red. I wish I could have made the app look better but I didn't have time for that.

The basic idea behind my solution is that for stage one I send an api request to https://sunrise-sunset.org/api and get a supposedly accurate response for sunrise and sunset. However for stage three I use a library called SunCalc(https://github.com/mourner/suncalc) which gives a less accurate response but I use it because sending up to 600 separate requests for the graph will overburden the api, is difficult to code, is slow and also accuracy is not that important for the graph. (For a real project I would have to more thoroughly validate the claims about accuracy and inaccuracy, I am currently basing this off some forum posts)

(1h) Got an overview of what needs to be done. Got a general overview of OpenStreetMap.

(1h) Found out how to add a graph to a website, decided for Google Graphs because the documentation seemed to be good. Thought about whether to use a framework like Angular or simple html and javascript, decided to use simple html until I find a problem where a framework would be useful and then change if necessary.

(1h) Found out what is the EPSG:4326 coordinate system. Found a simple api that gives the day length based on coordinates and date (https://sunrise-sunset.org/api)

(1h) Looked at how to implement stage 1

(1h) Decided to use Angular because sending http requests and receiving data is easier with it

(1h) Set up angular

(1h) Looked up how to use user input with angular

(1h) Got user input with dates working, made DaylightResponse interface and daylight-time service to get data from api

(1h) Looked up how to send requests and handle the response with angular

(1h) Managed to get response from api and display it

(1h) Formatted daylight time to show hours, started to implement the map

(1h) Map is now displayed on website, ran into error "Cannot read property 'setLatLng' of undefined" when trying to show a popup when a user clicks the map, it seems that the app doesn't properly create the popup instance for some reason

(1h) Found that the previous popup error might be a bug in leaflet, didn't research further because I realized that using a popup is not a good idea anyway, instead I will just update the input elements with the coordinates of the map click

(1h) Ran into problem where if I try to update the coordinates in the callback function that is called when the map is clicked, then the two-way binding of angular stops working properly

(1h) Solved! It seems that the problem is that leaflet doesn't work exactly the same with angular as it does without, so I can't expect code examples from leafletjs.com to work properly. Used a different code example from https://stackoverflow.com/questions/57255614/angular-and-leaflet-add-marker-on-click-and-get-coord, modified it for my use and got it working. (spent a total of 1h30m on problem)

(1h) Cleaned up code for selecting coordinates from map, found a problem where if the sun never sets and daylight time is supposed to be 24h, then the api returns daylight time as 0h. Looked at the data returned and noticed that when the day is supposed to be 24h long the api seems to always return "1970-01-01T00:00:01+00:00" as the sunrise date, I can check for that and correct the error

(1h) Fixed 24h daylight problem, started looking at how to get date range from user

(1h) Figured out how to get date range from user

(1h) Ran into errors with google graphs. It works a little differently with Angular so can't use google documentation directly. Also the google graphs angular module seems to have made some changes recently so the the tutorials I found were dated and didn't work. Finally found the answer from google graphs angular module github page.

(1h) Was looking for how to send multiple api requests for stage 3 and came to the conclusion that I should calculate the stage 3 daylight time and not use the api, I read that it's less accurate, but sending so many api requests might overburden the api provider and will take more time to get all the responses, also coding this type of thing seems complicated and accuracy is not as important for the graph

(1h) Found a library that calculates sunrise and sunset https://github.com/mourner/suncalc, got it working

(1h) Got a better understanding of how javascript dates work so that the graph would not be skewed by a day because of local time and UTC time differences, decided to use UTC time to get data for the graph just in case

(1h) Used the suncalc library to get daylight time for each day in the date range for stage 3, now need to add it to the graph

(1h) Got the graph working, looks cool, refactored some code

(1h) Found a problem where if the sun never sets(The Midnight Sun) then the suncalc library returns sunrise time as an invalid date, same for when the sun never rises(Polar Night), this means that when the sun is up 24h then the graph shows this as 0h, couldn't find a solution like I found with the sunrise-sunset api. Left problem unsolved because it's not important enough to spend more time on.(1h working on problem)

(1h) Improved UI, did input validation for most common cases

(1h) Displayed sunrise and sunset times as well

(1h) Uploaded project to github, tested the app on different browsers, works on every browser but when running on https://stackblitz.com/github/stenno12/day-length-app then sometimes the y-axis of the graph goes missing, I don't think it matters because stackblitz is just for demonstration anyway