For many people, working at CERN would be the greatest honor and a dream come true, I can say I am not the exception. Since I first heard of CERN, thanks to a teacher in High School, I felt that this was a place that I needed to experience for myself and not only read about. At my University I started searching for summer programs, and when my physics professor suggested the CERN Summer Student programme, I knew this was the opportunity I had been waiting for.

After arriving at CERN I learned that I would be working on the Detector Control System (DCS) group of the ALICE experiment. This group does pretty much what the name suggests: it creates control systems for the ALICE Experiment. This is done using a vast number hardware and software components. Given that I am majoring in Computer Engineering, I believe that being assigned to this group was the best opportunity to develop my skills and learn from other fields such as particle physics.

After meeting with my supervisor, Ombretta Pinazza, for the first time, I learned that I would be working on a program called WinCC Open Architecture. This represented a challenge for me because I had never heard any mention of said program. WinCC OA is a Supervisory Control and Data Acquisition system (SCADA). This type of system is basically used to gather and analyze data in real time, which can be used to monitor and control a system remotely, in this case the ALICE Experiment.
WinCC OA is specifically used for “visualizing and operating of processes, production flows, machines and plants in all lines of business”. Most of the work in WinCC OA is done in the Graphics Editor (GEDI).

Figure 1: WinCC OA Gedi

In this interface you will work creating panels and scripts that will serve as a “bridge” for the Man-Machine communication. When working on a panel you first use a graphical object, such as a button or text field, and then program it to work as you desire. To program in WinCC OA a customized version of the C language is used, this played to my advantage because I already had some knowledge on this language. My first task with WinCC OA was to get familiarized with the interface, something that was made more difficult due to the lack of documentation. In this matter I am glad that my supervisor was always ready to help me with any doubts I had about the program. I started by learning about datapoints (dps) and how I could connect them to different
elements to achieve a result. These datapoints were values, of different types, that you could analyse and change depending on your needs.

After getting familiarized with the GEDI and how to manage the datapoints I was ready to move forward in my work. While using the dps I learned that one could add different configs to them. The config that was used the most in my panels was the one called Alarm handling. With this config one could set an alarm every time the datapoint reached a determined value. There were three different type of alarms used with the dps: **Warning**, **Error**, and **Fatal**, these are in increasing order of their severity. Before creating any panel, I created a floating-point type datapoint called X and a script that would change X to any value between 0.0 and 10.0. Afterwards added the alarm handling config to X and assigned the limit values for all of the different alarm types. After all of this was done, I was ready to create my panel. My first panel, was designed to monitor any datapoint value in real time, in this case we used X. The panel consists on a trend that shows the current and past values of the datapoint for a user determined period of time. There is also a time stamp, below the trend, showing the current date and time. There is also a display showing the exact value of X and some other elements that warn of any alert and what type it is.

![Figure 2: My First Panel](image-url)
It is worth mentioning here that there was another panel created that was very similar to the previous one. This new panel just consisted of a trend and a timestamp, but it was designed to monitor the activity of the datapoint over a 24 hour period.

After creating the previous panels I was presented with the task of making a script that took screenshots of the panels. For the first panel, the screenshots would happen every time there was an alarm. For the dailyTrend panel, as the name suggest, the screenshot would be taken every 24 hours. This task seems easy enough when you discover that WinCC OA has a function for that purpose, the problem was that the function only worked when implemented in a panel. This means that it was impossible to use the function in a script, as was intended initially. To battle this problem we came with the solution of creating an User Interface manager (similar to the control managers used for the scripts) and used it to run the panel in Silent Mode. This basically provided us with the same outcome as using a script. Then again, we had the problem that the previous solution did not work that well with the daily panel. For this, my supervisor provided me with some code used before by the DCS group to take screenshots. After using the provided code I was able to take both screenshots with no problem.

After working with WinCC OA for a while, I was assigned the other part of my project: creating a website where all the elements I had created would be displayed. After having a website generated from the CERN web services I was set to work. It is worth noting that, even when I had some html knowledge, this was my first website. I started modifying my website using Microsoft Expression Web 4 but quickly changed to
Notepad. For this website I ended up working in HTML, CSS and XML. The first two, were to structure and style the website, which contained the screenshots taken from my previous work. The XML was used to create an RSS feed that people would be able to subscribe to in order to automatically receive any update from the website.

The website was pretty much set and the panels were showing, but we still needed a way to update the website remotely. The first step I took was modifying the panels so that the time and date of when the screenshots were taken would appear on the XML file and, therefore, on the RSS feed. Of course, the screenshots were always updating themselves so the website always had the latest alarm registered. The last panel I worked on, had me going back to the beginning of my work, as I had to work with datapoints. This panel was different from the others because it was made to receive some input from a person. In the panel you were able to select a specific datapoint from a list (the datapoint type would appear on the side) and specify an alarm condition for it. In addition to this, the user could create their own XML entry that would then be added to the RSS feed. The third part of this panel worked basically like the previous one but this time it was the HTML file that was modified.