

Antonin BOUVRY

Promotion 2020 à l’université de Strasbourg dans l’école Télécom Physique Strasbourg

Spécialité Réseaux et Informatique

Second Year Internship Report 2019

-

Junior Software Developer



Gas & Power

***Company: Total Gas & Power***

***From 03/06/2019 to 31/08/2019***

***Place: 10 Upper Bank Street Canary Wharf, London E14 5AB***

# Acknowledgments

I would like to express my appreciation to Total Gas & Power for the opportunity to develop myself in this rewarding and motivating environment. Thanks to Tristan Sheppard for believing in me and for his support during this internship. I also want to thank the whole IT department's members for their attention, welcome and their guidance. Furthermore, I would like to thank the Front Office team, they gave me great advice during this internship. It was a true pleasure to work with them.

## Abstract

### **English – Realization of features to follow financial transaction**

This report presents my internship in Total Gas & Power at Canary Warf, London. It explains the different work I had to do as a Junior Software Developer.

During this internship, I was in charge of developing an application used by people who work in finance, especially for people who control the risks linked to the sales done by traders. I had to help them detect easily high-risk deals through the application.

I succeeded in creating a graph and a grid with the data they provided me. The graph and the grid contained market prices for a specific product chosen by the users. With a quick look, users were able to compare their prices with other market prices and know if their deals were too risky or not.

I also added various features to improve their experience when using the app.

### **Français – Réalisation de fonctionnalités servant le suivi de transaction financière**

Ce rapport présente mon stage à Total Gas & Power situé dans le quartier de Canary Warf, London. Il explique les différents travaux que j’ai fait entant qu’apprentie développeur d’application.

Pendant ce stage j’ai développé une application utilisée par des personnes du milieu de la finance et plus précisément par les personnes qui contrôlent les risques liés aux ventes/achats faits par les traders. Je me suis chargé de les aider à détecter rapidement les risques à travers cette application.

J’ai donc créé un graphe et un tableau avec les données qu’on m’a fournies. Le graphe et le tableau contenaient les prix du marché de produits choisis par les utilisateurs. Avec un simple coup d’œil, les utilisateurs de l’application pouvaient comparer leur prix avec ceux du marché et finalement valider le prix initialement proposé par les traders.

J’ai aussi ajouté plusieurs fonctionnalités pour qu’ils aient une meilleure expérience de l’utilisation de l’application.

# Contents

**Companies Profiles** 6

Total Gas & Power 6

**Junior Software Developer Intern at Total Gas & Power** 7

IT department and Team organization 7

Position 7

Technologies and management tools used 8

**PRiSM Project** 9

Contents and specifications 9

Development and Deployment Process 13

**Conclusion**  5

## Companies Profiles

### Total Gas & Power

In this section I will present the history of the company. And then, I will explain its activity and how this company creates value.

#### Total Gas & Power history

Total Gas & Power is a business energy provider in the UK. It is an owned subsidiary of French Oil Supermajor Total SA.

This subsidiary was formed as AGAS Ltd during the deregulation of the business gas market in 1987. It became the first independent supplier to contract gas for direct sales into the UK market. It became a wholly owned subsidiary of Elf Aquitaine and it changed its name to Elf Business Energy in 1996. Now its name is Total Gas & Power to symbolize the growing integration with Total.

The industry of Total Gas & Power is part of the industry of utilizes. It means that its goal it is to maintains the infrastructure for a public service. The utilities that the company provides is mainly natural gas and electricity. It supplies over 175 000 sites across all market sectors: from small and medium business to large industrial and commercial companies, as well as the public sector and non-profit organizations.

The vision of the company is to be the UK’s most trusted energy supplier and that the company can deliver what it promises. Its mission is to harness the energy and resources to drive benefits for its customers, its employees, and the Total Group.

The TGP’s head office is at Canary Wharf in London and hires more than 200 employees. TGP is also based in Houston (USA), Singapore and Geneva (Switzerland).

Nowadays, its business covers the entire oil and gas chain, from crude oil and natural gas exploration and production to power generation, transportation, refining, petroleum product marketing, and international crude oil and product trading. The CEO of this French multinational is Patrick Puyanné and the chairman is Thierry Desmarest. The head office of the group is in La Défense in Paris.

The company has a website that provides many details: <https://www.gas-power.total.co.uk/>.

#### Commodities Trading and Information Technology

The company covers the trading and marketing of gas electricity-related products. It also trades the power from renewable energies and coal.

Total Gas & Power creates value by optimizing the production and the marketing assets of the Total Group on a defined set of markets and commodities. These markets have become more open and flexible. TGP has introduced new ways of balancing supply and demand, creating pricing formulae with hedging and decision-making tools.

The needs of analysts and traders are unique. They need customs tools and systems, databases, calculations, services, and scoreboards in order to help them make the best decisions. All of this is providing by the Information Technology department.



Figure 1: Logo Total with a sentence about its everyday goal

## Junior Software Developer Intern at Total Gas & Power

During my internship, I joined the IT department team at Total Gas & Power in London. I worked with the developers of the Front Office team. There I discovered their organization, their structure, and their management. I had also a lot of good interactions. The people I worked with were very kind and they helped me to develop my developer skills. Thank to them I had been able to fulfill my duties as a junior software developer.

I also learned a lot about me, how to position myself, to grow while gaining self-confidence. I built myself by realizing my value. I was recognized, valued, appreciated for my skills and my interpersonal skills.

During this internship, I experienced the busy schedules and the pressure of the objectives and the various ways of managing teams.

In this section I will present the IT department and the organization of the Front Office Tools Feature Team and after I will explain what was my position in this environment.

### **IT department and Front Office Tools Feature Team organization**

Front Office is a team of developers which is part of the IT department. In the IT department, there are also the IT support, the Business Application, the Databases Administrators, the Network Administrators, the Compliance & Control team, and the Security team. As most of those teams are in the same building and some of them are even on the same floor, they can communicate easily and enforce their teamwork. That helps applications to be great and well deployed.

Each team contribute to the development and maintenance of applications. The team I worked with is in charge of providing the traders and analysts new tools for their emerging needs.

IT department is composed of a team manager, KDB developers, .NET SharePoint Developers, and .Net Developers. They are working on various projects, and one developer can work on multiple project at the same time.

The Front Office Tools team is a feature team. It’s mean that its goal is to develop the applications the team worked on. We were using Agile method. So there was a Product Owner who represented the user interests of the application so he had to define the features of the final product. There was also a Backlog manager who manage the backlog, it is a list of duties we had to fulfill to create the product we wanted. Finally there were the developers who modified the source code of the application to install new features or to fix bugs. We were 5 developers.

Front Office Tools team create new features when they saw the needs or when the users of the applications asked them to add a feature. They are directly linked to Front Office and Back Office department. Front Office departments are responsible for sales and trading while Back Office department manages risks.

### **Position**

I have been included in Front Office team as a Junior Developer and I worked on PRiSM project which is used by Back Office to control deals and sales. During this internship, I have been guided by the other developers on the team. I started to work on small tasks to become familiar with the application architecture, the programming languages Angular and C sharp, and the way to deploy my code. I had the opportunity to discuss with the users of the PRiSM application to know what their needs were and I had the opportunity to suggest them various solutions.

In each task, the workflow was the same. People from the Back Office expressed their new needs to the business analyst team. This team is in charge of the communication and comprehension between the users and the developers. They write functional specifications. After validation, this document is used by the development team in order to produce a technical design, compound architecture and the technologies to use. The team also specified the patterns, the framework and the connection to the databases or external tools.

This part is primeval for the project. The needs have to be correctly identified and understood for the acceptance and the efficiency of the application. Not only does the goal the application has to be properly understood but responsivity and security are analyzed at this step.

As soon as the specifications are produced and validated by all the concerned persons, the development begins. After the development, there are tests to do and finally, the application can be deployed.

### **Technologies and management tools used**

During my internship, I used different software and technologies. I developed in Typescript and C sharp using Visual Studio with Angular Framework (figure 1). PRiSM has been connected to two types of databases: Oracle and KDB.

I also worked on various features. Each time I had a task on those features, I needed some different libraries and/or technologies. So I started with small tasks to learn how to manipulate various technologies and libraries. I will explain mainlines.



Figure 2: Angular logo

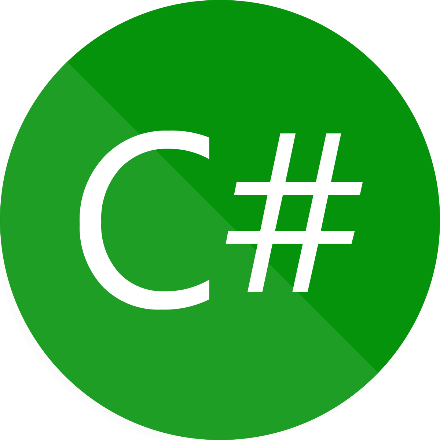


Figure 3: C sharp logo

For confidentiality reasons, this report does not contain any details of architecture or process of the Total Gas & Power Ltd applications.

## PRiSM Projects

PRiSM is an application I worked on. It was a desktop application used by Back Office team. They used PRiSM to verify all the daily transactions done by the traders.

I worked on several application features during my internship. Some of them were shorter than others. With the PRiSM project, I have been familiar with Total framework and I discovered the deployment procedure. In this section, I will explain the content and specification of my internship. I will first explain the main feature I developed in PRiSM Service. After, I will shortly talked about the web design I worked on and the back end of the PRiSM application. Then, I will present two features I developed: URL feature and the creation of a graph and a grid.

### Content and specification

#### The main feature I developed in PRiSM Service

The main goal of the PRiSM application is to analyze quickly and without error the risk for each deal/sale.

I spent two weeks in this project to handle the application, its architecture. The Back Office team needed improvement in the visualization of the data linked to the deals. They spent too much time comparing their prices to the market prices in Excel sheets. Using Kendo library I added a graph and a grid to be able to see data correctly. I had to recuperate data from a database and Excel sheets, process it and use Kendo libraries to create the graph and the grid.

After that, I modified both of them to answer to the user needs.

There were different needs from the users. The main goal was to help Back Office people to compare the deals the traders have made to the market tendency. Each time one of their deal for a product was too risky, they needed to see it quickly and make a decision: keep it or not.

So I implemented a code color to help them detect the main information.

Thus, I learned to share my codes to other developers with Team Foundation Server and I learned how to implement a Kendo graph with Angular.

#### Design part

Because the users are using the application on a daily basis, I had to work on the design to improve their experience when using the app. I used what I learned at the university to create a simple but well-looking design. The language I used was CSS and HTML.

So I took some online lessons to get better in those programming language. I also used Kendo libraries to create graphs and grids.

Kendo UI is a software that helps developers to create responsive application. The libraries that the developers integrate in their development environment help them to create graphs or charts very quickly and with a lot of options. So it was a great tool for me and it helped me to create beautiful interfaces and responsive graphs that were used by Front Office team to check the trader’s transactions of the day.



Figure 4: Logo of CSS

#### The database implementation

PRiSM is connected to databases. The software used to implement the database was MongoDB for the developer teams. It is a cross-platform document-oriented database program. MongoDB uses JSON documents. I had to understand the basic features of MongoDB to use it properly. This database were used to create mock data or real data and to collect data that was used by the applications we developed. This database was fulfill by a other database: kdb+.

The Database team used kdb+. It is a column-based relational time series database with memory abilities. This software is developed by Kx Systems. The Database team used this software to collect all the data of the company. For example, the market prices of the product the traders traded were stocked in the kdb+ databases. Those database had a high level of security and a 7 people on a full time job were working on it to make data secure and available.

Moreover, to use correctly the MongoDB, an Application Programming Interface (API) had to be set up. The API was used to retrieve and modify data in the MongoDB databases. So there were parts of the API source code that contained request in the SQL programming language for example. There were also the object creation part which enabled the developer team to use properly the data in their application.

When I arrived, there already was an API. So I needed to learn how it was working. It took me a while because the source code of the API was written in C sharp programming language and this language can be difficult to understand. To study this programming language, I used the website <https://app.pluralsight.com/>. The company created me an account to be able to have access to online lessons. So I studied during a week the fundamentals of this programming language.

After learning C sharp basics skills, I had to understand how the architecture of the API worked. The API is composed in different modules. Each modules are linked to fulfill the main goal of the API: create a connection between the desktop application PRiSM and the database. This connection has to be secure and reliable to ensure the policies of the company and to provide essential data to the application. The total source code lines was about 10 000, so, sometimes, I had to ask for help to my coworkers.

When I understood its architecture, I have been able to complete tasks for new features. For example, to retrieve specific data from the database, I had to ask a database administrator some permissions. Then I modified the code source of the API. More precisely I changed the request to the database in the request module of the API. Sometimes I had to process data to display it as I wanted to. So I had to understand how the object was retrieved and how to use it.

For example, one of the feature of the desktop application was that the users were able to change the colors of the graph and of the grid accordingly to its color code. So I had to search in the database how the color code was stocked. Then I had to create the request that enabled the retrieve of the code color. The colors were written in RGB color model. But the front end of the application that used Kendo libraries needed the colors to be in hexadecimal. So, I had to process the data in the API code source.



Figure 5: Kx Systems logo (company that developed KDB software).



Figure 6: MongoDB logo

#### Kendo Issues

Two times, I could not add a feature as I wanted because I founded an issue with Kendo technologies. So I had to report the bugs to the Kendo developer team with my GitHub account (figure 7 and 8).

One of the two bugs was that the negative values prevented from zooming. I first thought that I had an issue with the code lines I wrote. I searched different ways to correct my code. It is only after I tested all the different solutions that I consider there was an issue with Kendo libraries.

Firstly the Kendo developer team acknowledged me that they identified the error and that they will correct it. Secondly, they explained me that the correction of this issue will be in a release they planned in the end of July 2019. So I had to wait until this date to provide a solution to negative value issue on my graphs. The bug report I made in on the figure below.

The second bug report (as shown in the next figure) is also about graph.

Indeed, the labels on the X axis of a Kendo graph disappeared when we used the zoom feature. This feature was very useful since it helped a lot the Back Office to identify which trade was too risky for the company. Indeed, the Back Office users of the application were able to precisely see the difference between a threshold (the price limit of a trade calculated with value market) and the trade price.

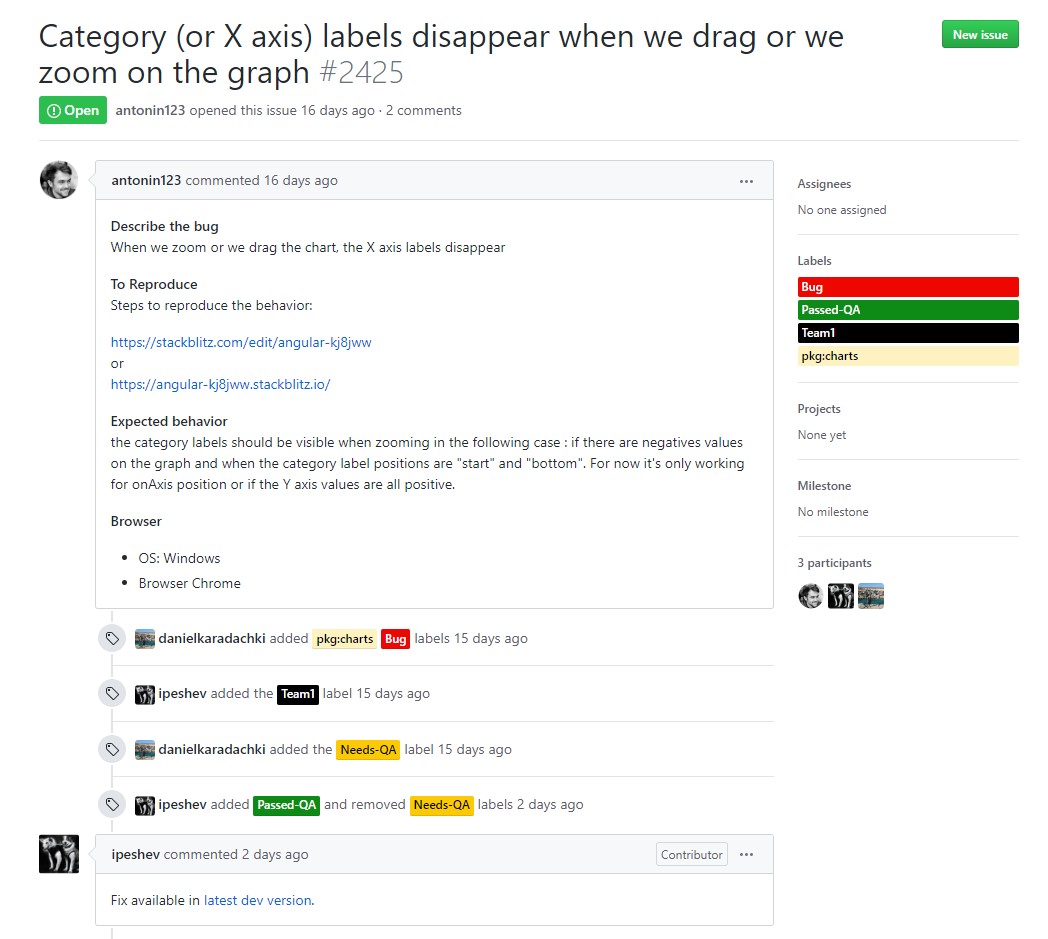


Figure 7: report 1 of Kendo bug

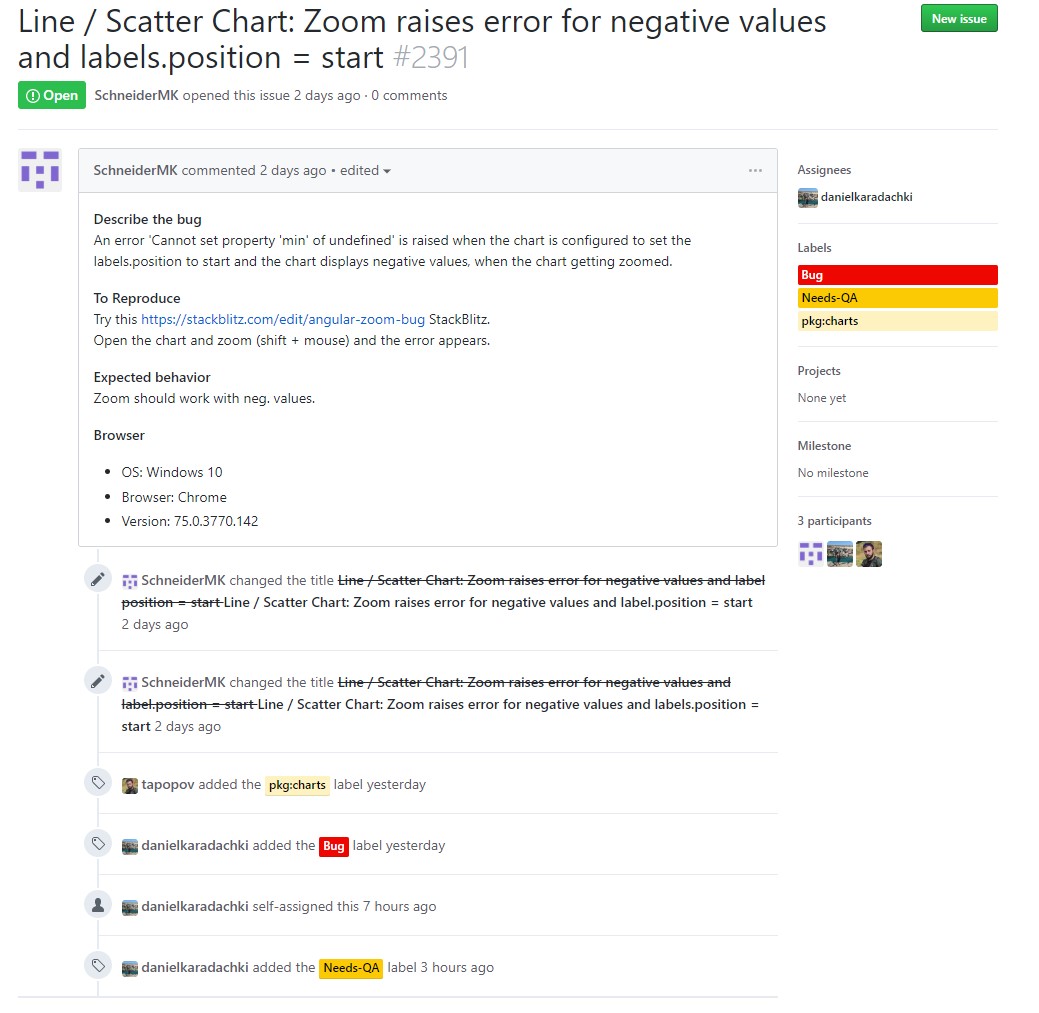


Figure 8: report 2 of Kendo bug

#### URL feature

##### Analysis of the need

Back Office users wanted to be able to share specific data between them. For example, one person who found a high sale risk on the graph and he wanted to notice a colleague. And because the application has a lot of parameters to choose before displaying the graph, I had to find a solution to share what he found without specifying all parameters to his coworker.

So I decided to add all the parameters in the URL each time a new one was selected.

##### Realization

I had to implement a robust way to navigate through the application. It means that I had to verify that the shared URL between two Back Office users was correct for all the views of the applications.

So I used tools to add parameters in the URL, retrieve the URL, and process it. The difficulty of this task was in the fact that I had to perfectly understand how a trade is defined. I had to know how all the parameters of a trade (date, location of the trade, department where the trade has been done…) are stocked in the database, how to request it and process it to fit its in an URL.

Moreover I had to know how a URL worked and all the writing specification for an URL. I used libraries to stock URL and process it. I had to test every single with all the parameters of a trade to detect bugs. For information, there were 7 parameters to each trades. That means that in each of different views (or pages) of the applications, I had to test if the parameters were well process by the view.

I needed to do it as simple as possible to keep the application reactive. It means that I didn’t have to load a view if it was not absolutely necessary. I also had to see if precedent page button worked properly.

There were a lot of different features I worked on but the URL feature was one of the most important ones. In the figure number 13, there is an exhaustive list of the item I worked on.

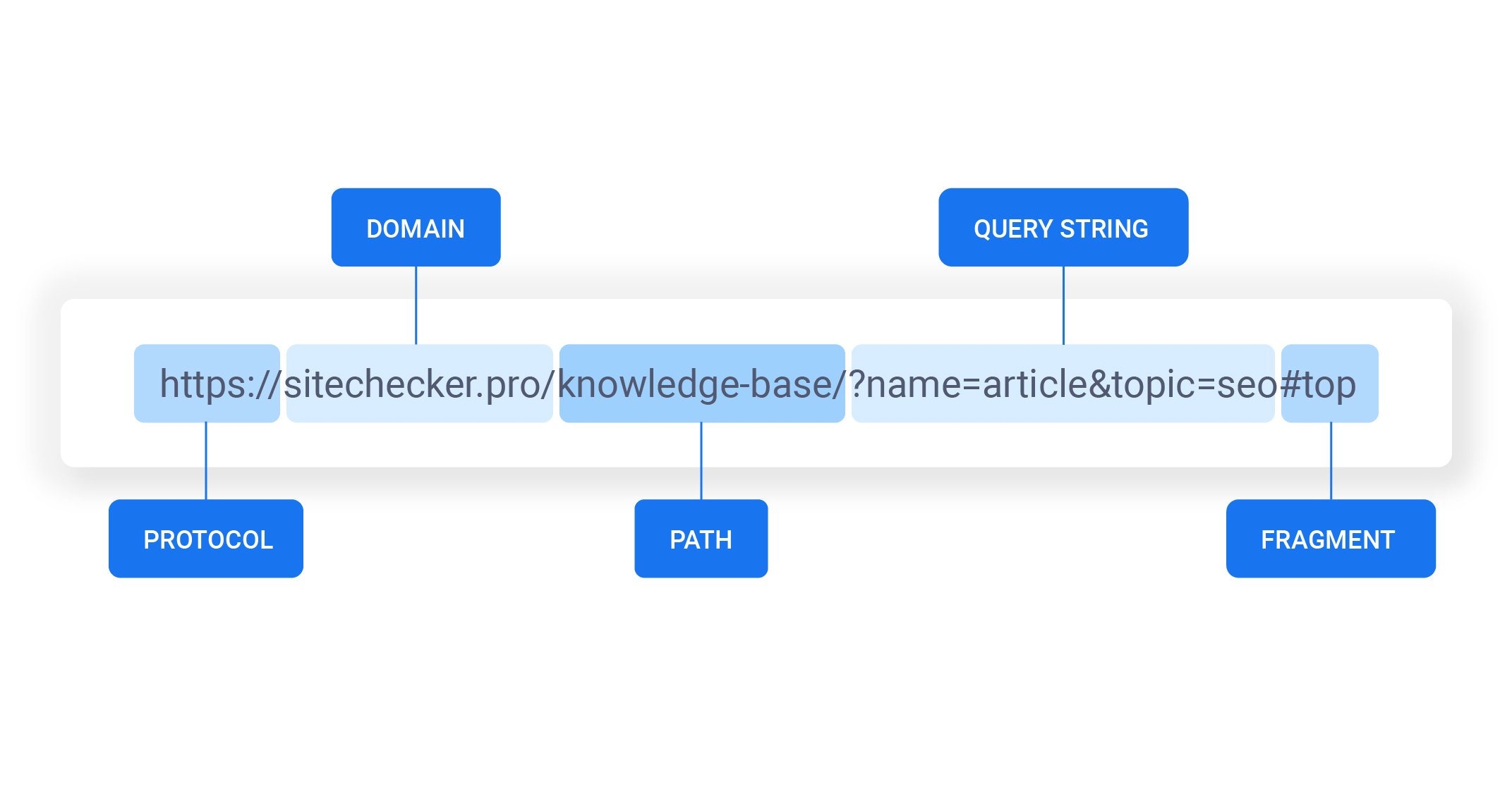


Figure 9: composition of an URL

#### Creation of a graph and a grid to follow trader transactions

##### Analysis of the need

The goal of the Back Office team was to evaluate the risk of a transaction that has been made by the traders. Before I arrived, the Back Office team used Excel to compare the product prices between the market prices and the prices that the traders estimated. If the team detected too risky transactions or trades, they asked the traders to justify the risk they has taken. If the justification was not appropriate, the team canceled the transaction.

There were two problems with this method. Firstly, it took too much time for the Back Office team to compare all the prices with an Excel file. Secondly, because the Excel files had a lot of lines, the Back Office team missed too risky transactions.

In this context, PRiSM application was created by developers of the Front Office Feature team where I was. This application helped measure difference between prices. But the application wasn’t develop enough and simple enough. So the Back Office team still used Excel files.

So I proposed to use Kendo libraries in order to create interactive graph and grid to simplify the reading of prices.

##### Realization

To realize this goal I had to understand how to use Kendo libraries. For this I used the online documentation: <https://demos.telerik.com/kendo-ui>. The programming language of the front end of the PRiSM application was in Angular. So I also took lesson on <https://www.pluralsight.com/> to know the fundamentals of this programming language.

The complexity of the task was to clearly identify the better ways to underline a highly risk transaction in a graph and in grid. I had to choose the right proportion of the page, use good feature to see correctly the difference between prices.

Firstly, I had to retrieve the data from database, I had to understand the object and the properties of a transactions. I had to go through MongoDB and the API to retrieve the data and process it to fit it in the grid and the graph.

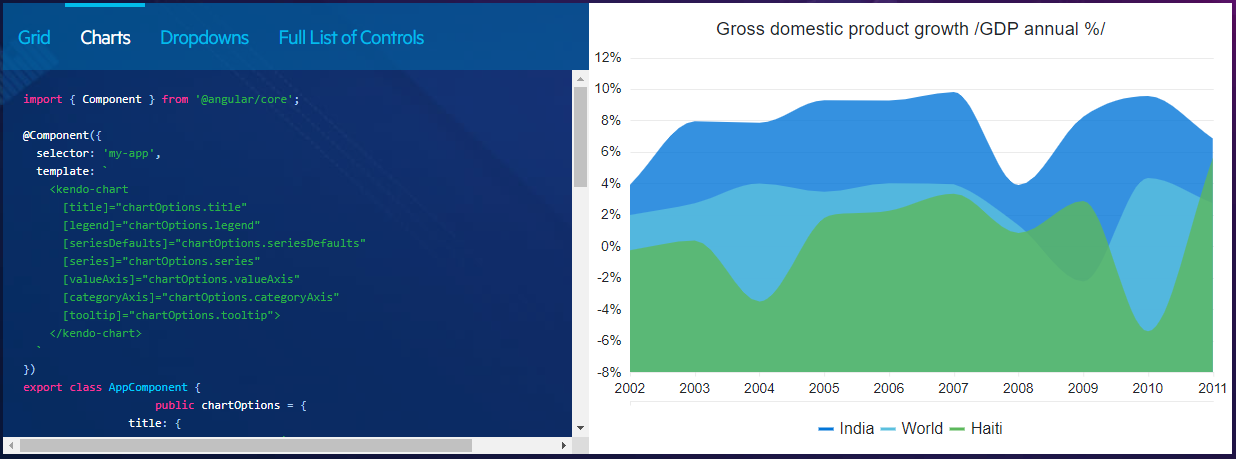
Secondly, I had to ensure that features of the graph and the grid enable the Back Office team to have, with a quick check, all the important information of the trader’s transactions of the day.

For example, for the graph, I implemented a code color with the zoom and pop-ups features. The zoom feature enable the user to look closer to the graph. The pop-ups feature enable the user to see the prices and the minimum and maximum limits in a shortest page when he clicked on a value of the graph.

For the grid I added filters to help the Back Office team to search what they wanted. Moreover I highlighted in red the transactions that were over or under the limit of the market price.

During all this work, I had a lot of meetings with users to understand their needs, their work. That help me to understand the complexity of the financial jobs such as trader or the Back Office jobs.

I could not take the graph or the grid in screenshot I worked on because I didn’t have the permission. But the grid and the chart looked like the figure 10 below.



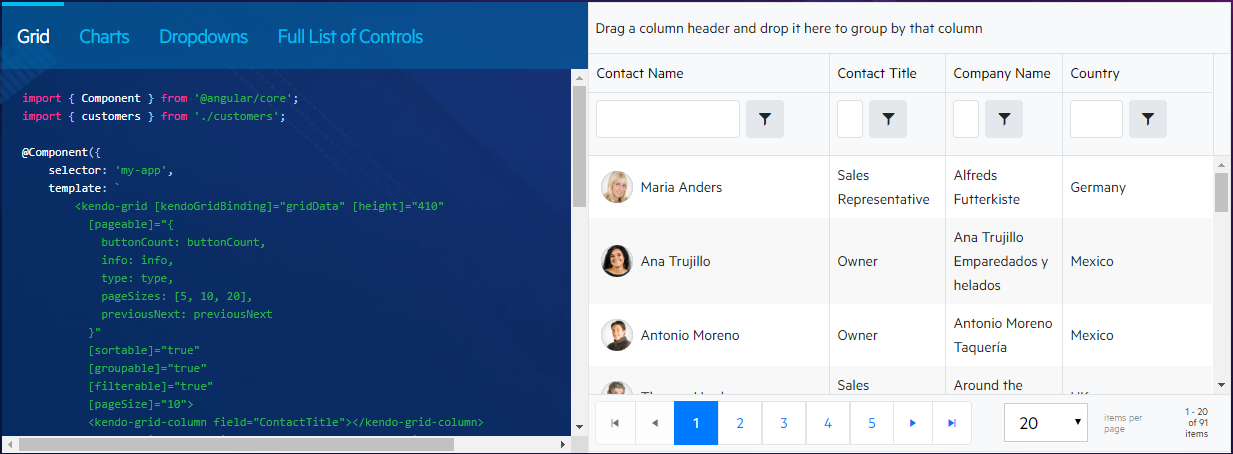


Figure 10: example of a Kendo grid and a Kendo graph

### Development and Deployment process

During my internship, I have been used to the development and the deployment of an application. So I had to understand the various steps to create and maintain an application. In this section, I will explain the different aspects of testing and the deployment procedure. After, I will present how I worked with the Front Office team.

#### Testing and deployment procedure

It is important to test our application, our source code before deploying it in a production environment. The FOT team has 3 different environment type.

The first one is the local environment. It means that we have to test our source code on our machine. For this first test we used the integrated development environment Visual Studio Code. To test my code I ran it on my computer. If they were compilation bugs, some texts were written by the compiler on my console in red color.

So I had to understand the issue and correct my error. But if the code compilation was right, I had to use the application and test it. I clicked in every button for the URL feature for example and I wrote right URL to see if the displayed view linked to the URL was the good one. I also had to write wrong URL to see if the source code of the application processed also the wrong URL.

So, as I said, the first tests were on my machine with Visual Studio but it is not enough.

To be independent of the workstation configurations, we used external a pre-production environment. This environment was used by Business Analysist team. This team used an application in this pre-production environment as the final users will use it. If a bug was found by the Business Analysist team, they notified developers and the developers tried to fix it. Finally if the application with the new features didn’t have any bug, the application was able to be deploy in the production environment and used by the final users.

In our case, the final users were the Front Office team or the Back Office team. It depended on the application we were working on.

Sometimes, the new feature of the application was not exactly what the final users wanted. In that case, we had a meeting with a user to discuss what we missed or we asked him to precise his demand.

The figure below represent the 3 sites where the application can be. The development site represents the local computer where the developers test their own codes/application. The staging site is the pre-production environment. Those two sites are in the internal network of Total Gas & Power in London. But the production site that contained the final application release was accessible by all the users who had not have access to the internal network.

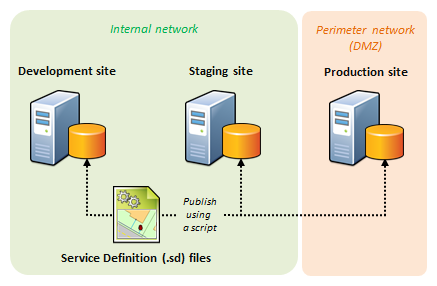


Figure 11: various site for testing the application

#### Working in a developer team

To work in an efficient way in the FOT team we use the Agile method. This allows us to implement easily new features. We fixed goals every two weeks and had exchanges with users and Business Analysis.

Each time we finished a feature we did a pull request. It means that, with Git technology, we shared our code without conflict issue.

We did every day 15-minutes meetings to know what we worked on the previous day on what we will work on the current day.

There were features on the TFS site and we chose what we wanted to work on. See below a little part of what I worked on (figure 13).

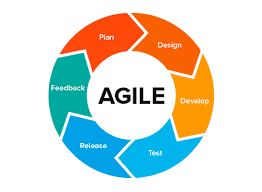


Figure 12: representation of the various step of Agile method

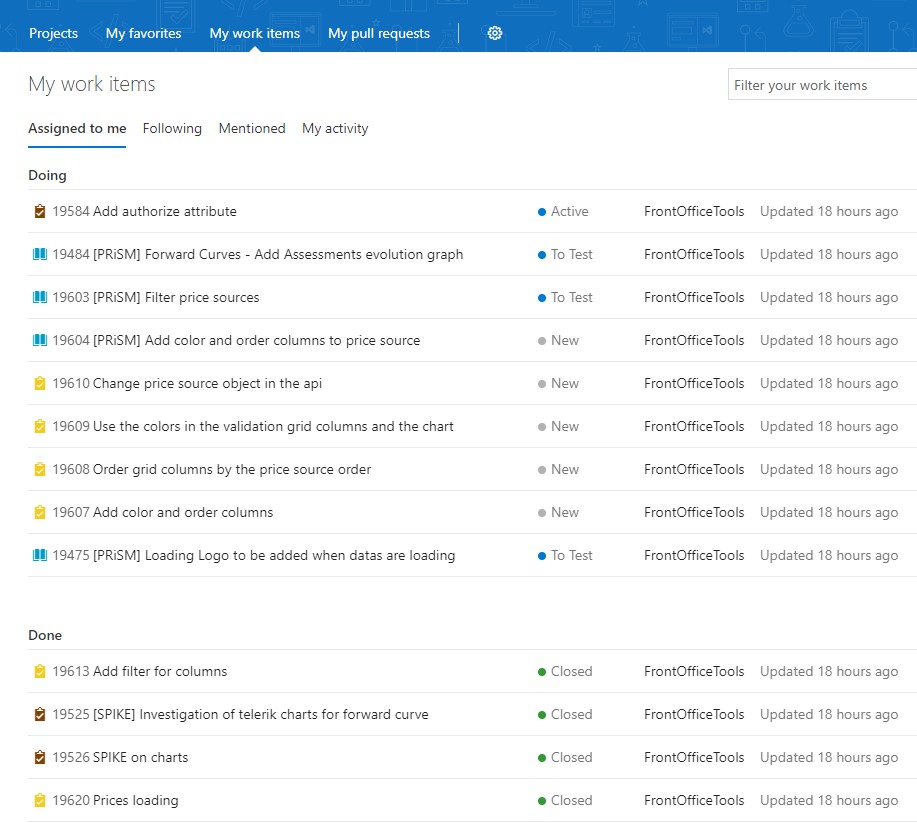


Figure 13: some work items I worked on

## Conclusion

This internship has been fully constructive. I am further fortified in my career choice. I greatly appreciate designing complex, rigorous and critical applications. It was also very gratifying to put into practice knowledge and concepts seen in school such as encapsulation.

This internship is in the continuation of my last internship. Surrounding by experts, I solidified my programming skills. I learned how to use Angular, C# and many other technologies. I learned how to share my code and work on a big application with a developer team.

I helped creating new features in applications and those features are useful and used in a daily bases. I made easier the treatment and the identification of highly risky trader’s transactions.

With the on-going dialogue with Business analysts, users, and other developers, I got efficient feedback on my work. It helps me to improve the way I work.

## Bibliography

Total Gas & Power logo: <https://www.gas-power.total.co.uk/>

Telecom Physique Strasbourg: <http://www.telecom-physique.fr/>

University of Strasbourg: [http://www.unistra.fr](http://www.unistra.fr/)

Figure 1: - C sharp <https://docs.microsoft.com/en-us/dotnet/csharp/index>

* Angular <https://angular.io/>

Figure 2: - kx<https://kx.com/>

Figure 3: -MongoDB<https://www.mongodb.com/>

GitHub <https://github.com/>

Figure 5: Team Foundation Server <https://azure.microsoft.com/en-us/services/devops/server/>

Figure 7 , URL : <https://www.google.com/search?q=URL&tbm=isch&ved=2ahUKEwi5l6WavtjoAhUU-BoKHbJBAbIQ2-cCegQIABAA&oq=URL&gs_lcp=CgNpbWcQAzICCAAyAggAMgIIADIECAAQQzICCAAyAggAMgIIADICCAAyAggAMgIIADoECCMQJzoFCAAQgwFQkt4gWNHhIGDq5CBoAHAAeACAAW2IAb4CkgEDMC4zmAEAoAEBqgELZ3dzLXdpei1pbWc&sclient=img&ei=S5WNXvnFBpTwa7KDhZAL&bih=615&biw=681&rlz=1C1CHBF_frFR891FR891#imgrc=AqqFh1uJHHvB3M>

Figure 11, test environment : <https://www.google.com/search?q=application+test+environment&tbm=isch&ved=2ahUKEwjn0aOPvtjoAhVLpBoKHXyUASoQ2-cCegQIABAA&oq=application+test+environment&gs_lcp=CgNpbWcQAzoCCAA6BAgAEB46BAgAEBM6CAgAEAgQHhATOgQIABAYUL98WImpAWCCqgFoAHAAeACAAaMBiAGHC5IBBDAuMTKYAQCgAQGqAQtnd3Mtd2l6LWltZw&sclient=img&ei=NJWNXudny8hq_KiG0AI&bih=615&biw=681&rlz=1C1CHBF_frFR891FR891#imgrc=pLVYPEdF1HlutM&imgdii=Gs7mim9r-HlqVM>

Figure 5, CSS : <https://www.google.com/search?q=CSS&rlz=1C1CHBF_frFR891FR891&sxsrf=ALeKk02OW0TcL5SE7uxeA4zRvvuCQvGeQw:1586339015180&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjU0PirxdjoAhUU9IUKHQI3BUMQ_AUoAXoECBYQAw&biw=681&bih=615#imgrc=m3C3rUcbt3bdsM>