PROJECT ON QLIKVIEW :

Stage 1:

We must have complete knowledge of the customer and its requirements.

We must have the answers of the following questions to develop any project:

1. What is the general explanation and background of the project means what reason triggered this project.
2. Who are the audiences of the report?
3. How this report will help them in their day to day business?
4. What kind of KPIs (KPIS means expressions in qlikview) are there in this app?
5. How these KPIs will help to drive the business (the reason to ask this question is because it gives developer control on what kind of UI objects (charts) to presented on front end app that makes the app , Kpis more meaningful)
6. Are there any YTDs (Year To date), MTD (Month to date), QTD (Quarter to Date), Rolling 13 months , YOY (year over Year) calculations.
7. Are there any Variance and Variance % age calculations.

Stage 2:
DATA IMPORT

QlikView is based on the use of flat data loaded in the live memory. The data stored in RAM are retrieved and prepared from a script defined by the user (in pseudo-SQL language). This technique, which is called in memory, can significantly reduce the volume of data (rate of about 10) whilst ensuring the uniqueness of the data fields.

When creating a QlikView project, the very first step is to load data within the application. This step is crucial because it determines the structure of the database that will be created within QlikView.

QlikView can connect to any ODBC data source including clinical tools such as Oracle Clinical and SAS. Note that in the case of ODBC connection with SAS, a SAS server with SAS Share is required.

To complete this step, the loading script must be edited. It can be done manually or with assistance from the user interface.

1) DATA SOURCE

To edit the script go to the File menu then Edit Script (or CTRL+E). An ODBC data source can then be selected in the "Data" tab. After choosing "Excel file" (see Figure 2), the user is invited to select the Excel file containing the data to import.

2) CONNECTION TO DATA SOURCE SCREEN

3) Each tab of the Excel file will be considered as a table, and the interface allows the user to preview the data, to obtain information about columns (format, size, name, data type) and to select all or only some fields in the file.

4) QlikView detects and automatically manages the associations between the tables and when the choice is validated, SQL code is automatically generated in the script.
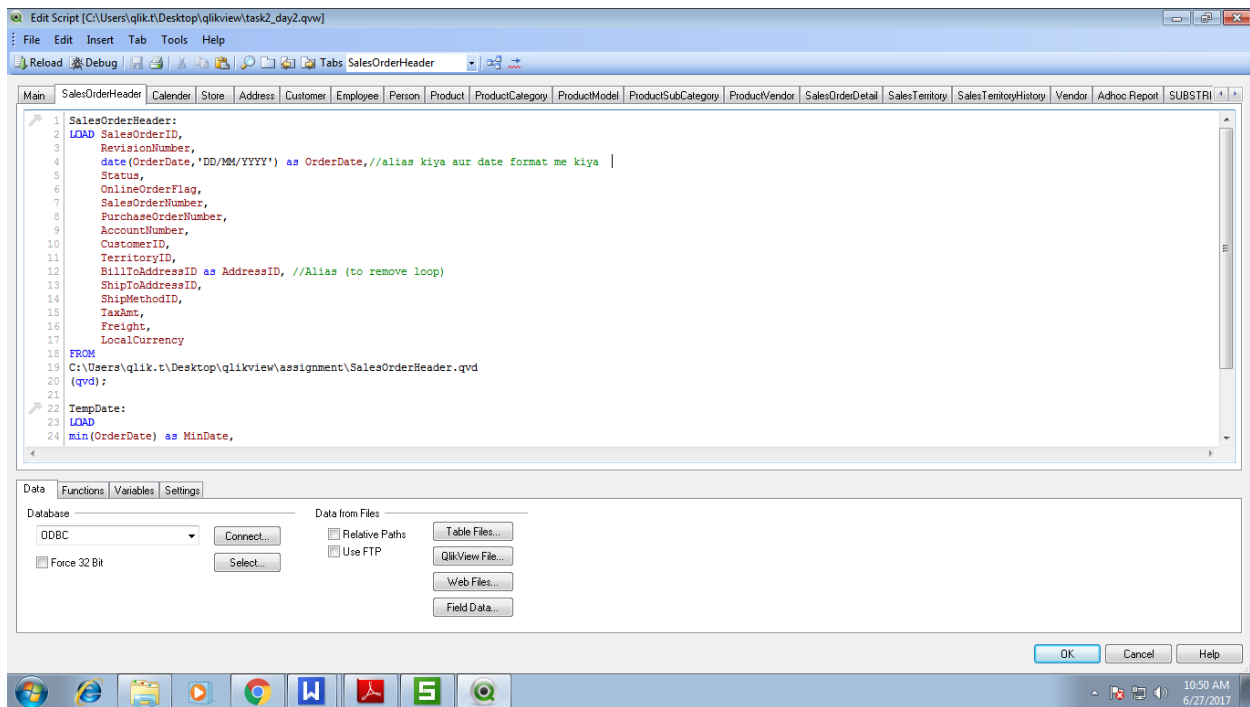
5) The embedded SQL engine in QlikView allows tables and fields to be renamed, restrictions to be made and functions such as „min()", „max()" and „distinct()" to be added.
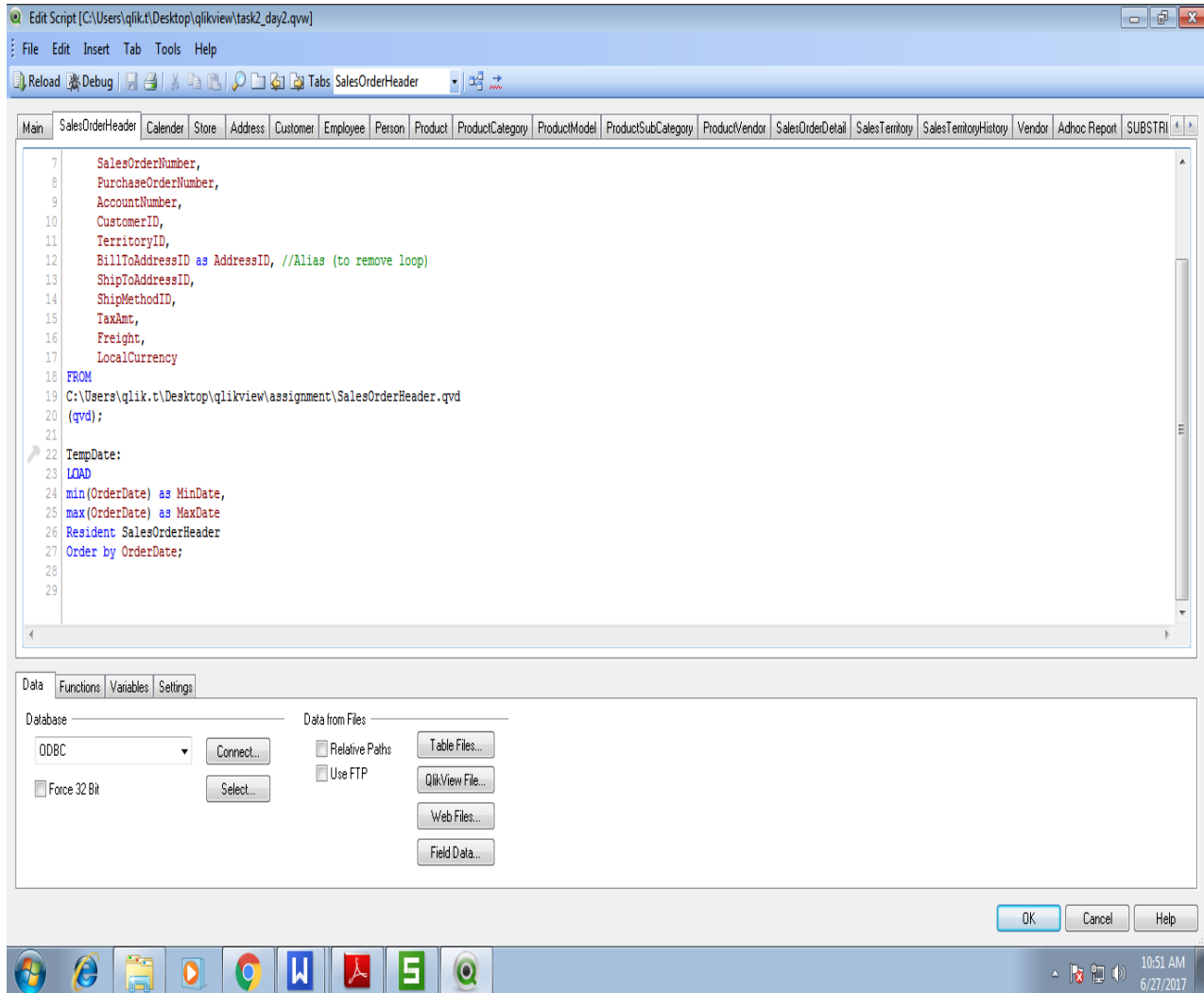
6) The SQL engine also allows temporary working tables to be created or erased and junctions to be made in order to provide solutions for the creation of some indicators. Once the script is entirely edited, data need to be loaded (File > Reload or Ctrl + R). The application will produce the structure of the data base (File > Table viewer or Ctrl + T to view it)
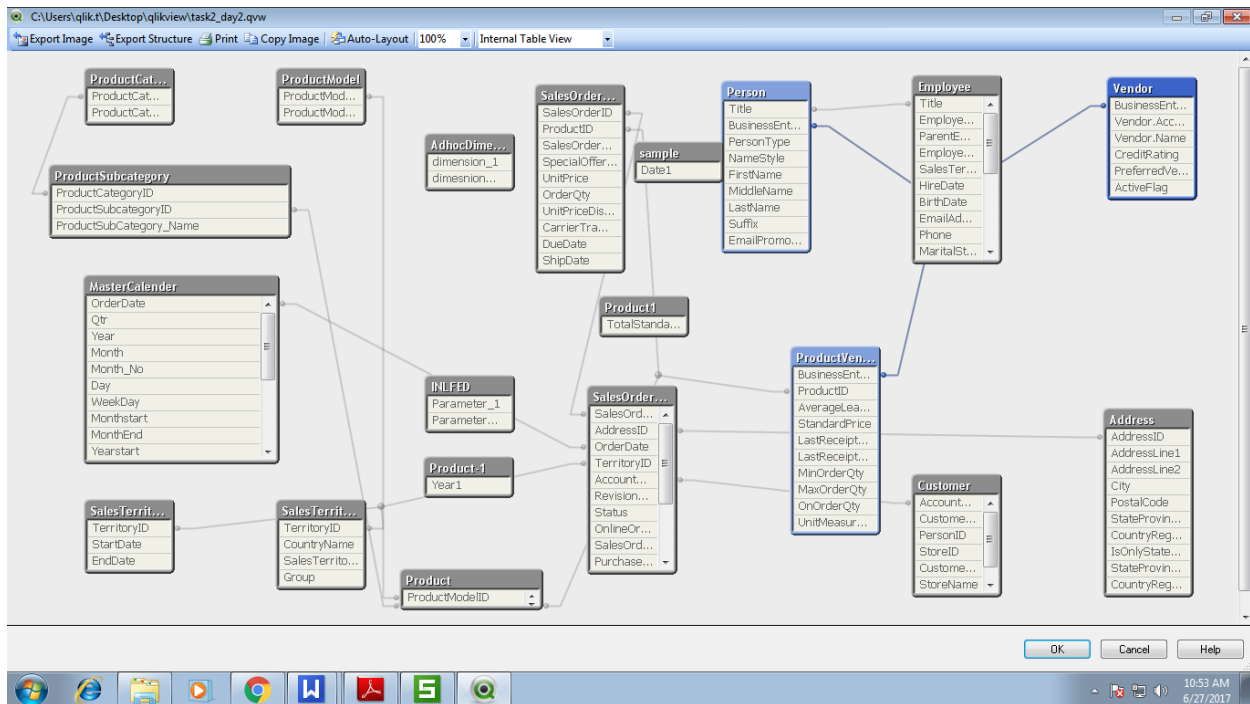
The application determines the key tables based on the field names of the loaded tables. Thus, if some tables have fields in common but have no link between them, the fields must be named differently in the loading script.

Once this is done, it is possible to create various objects to develop the application.

Stage 3:

DAR METHODOLOGY

The Dashboard, Analysis, Reporting (DAR) methodology is a foundation you can build all of your applications on while still having room to be creative and meet the varying requirements of individual clients/prospects.

 In a nutshell you lead with a Dashboard page, followed by Analysis pages, and finish with Reporting pages. The system works on a few levels but to understand some of why it works we have to discuss how people interact with computers and how we perceive information.

Dashboard

The Dashboard page gives just the most important information and has the least amount of interactivity/clicking. It is mostly to help users scan for status updates. If your users have only a minute or two to get the overall status of their business this is where they would do it. They can check in and see if things are working or not. It's a starting off point, like a table of contents - you get an idea of what is available and then head off to other parts of the application based on what you have see here.
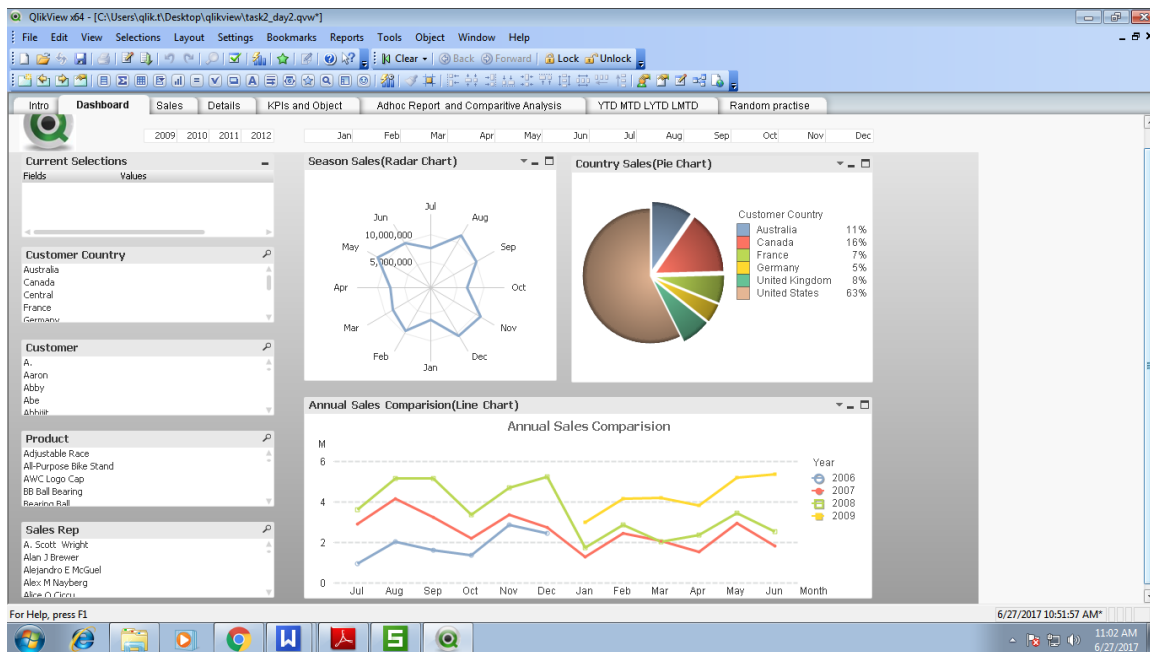
This is where the majority of high-level perception takes place. Users are getting a general first impression and the overall status of things.

Analysis

Analysis pages are more interactive, they help users explore their data and look for answers to questions they may have formed on the dashboard page. Analysis pages are where you come to spend more time and interact with the application on a deeper level to explore the data. Typically each page has a part of the business it is exploring or a specific technical objective (such as a page just for Comparative Analysis).
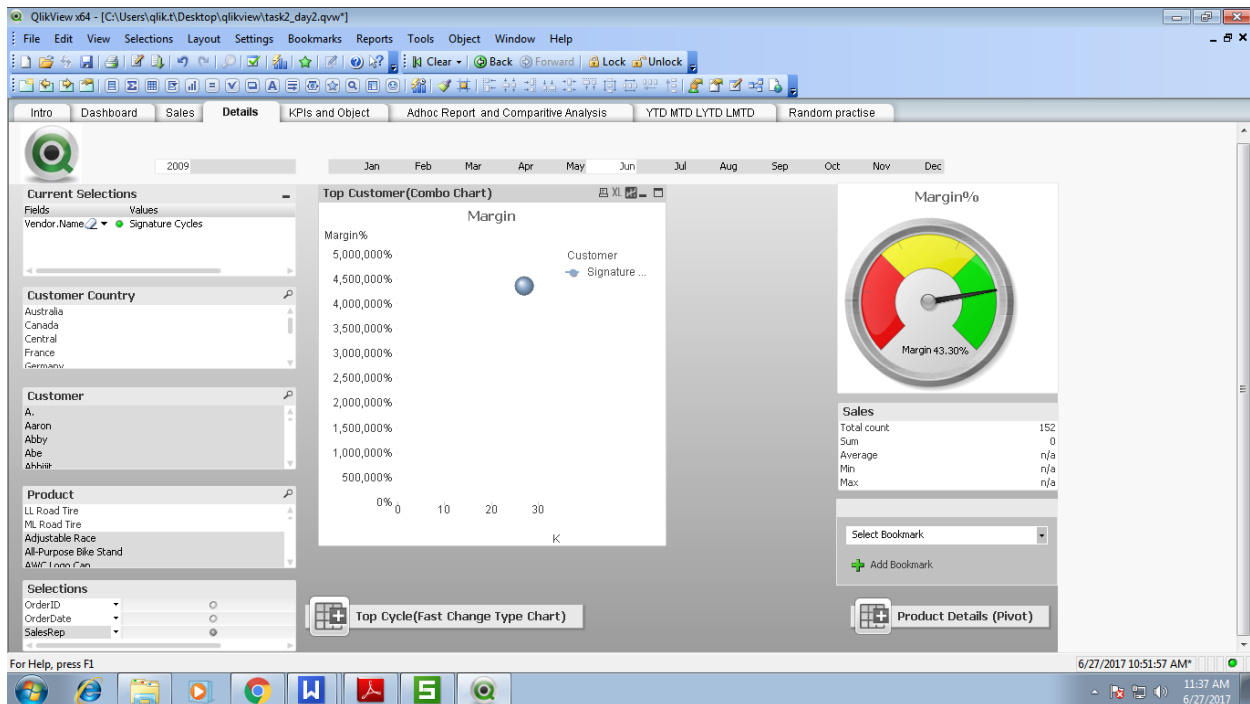
Reporting

Reporting pages give the most granular information with lots of tabular data and should ultimately lead to action. It's where a user can spend a lot of time sorting and filtering through the details. This is a large part of the cognition phase of perception.
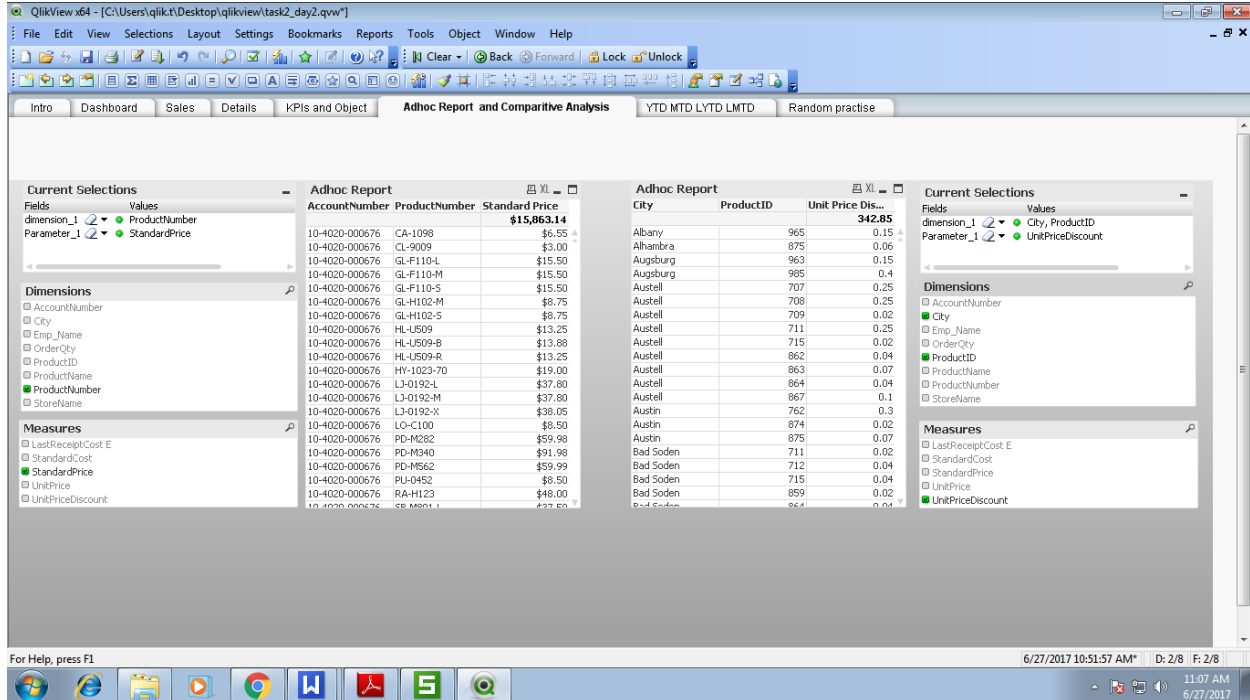
Before obtaining this result the developer must define the indicators and selection criteria. In QlikView these elements are called objects. As an example we will describe how to build a curve.

QlikView Objects

To add indicators to a page, a variety of objects is available. All these objects can be grouped by functions:

☐ Application of filters on the data,

☐ Display of data as table, chart or gauge,

☐ Design of application.

Among them, the chart object offers a variety of chart types (histograms, curves, grid, pie chart etc.) and also the opportunity to make a pivot table or a straight table which could host calculated fields or coloured indicators. Moreover, a function named „ast type chart"offers to the user several different representations for the same graph and allows the user to switch between them using a button in the object header.

Build a curve

During the collection of clinical team requirements, we were informed of the need for a theoretical inclusion curve, which allows evaluation of the inclusion rate within the various investigator centres. This highlights the way in which the developers have enhanced the user-friendliness of the tool by providing a desired indicator.

The following flowchart describes the different steps of curve building.

Graph Selection: First of all, the developer must choose the chart type by clicking on the object tab, then „New sheet object" and finally „Choose chart". The window shown in Figure 6 appears and displays the kinds of graphics available in QlikView.
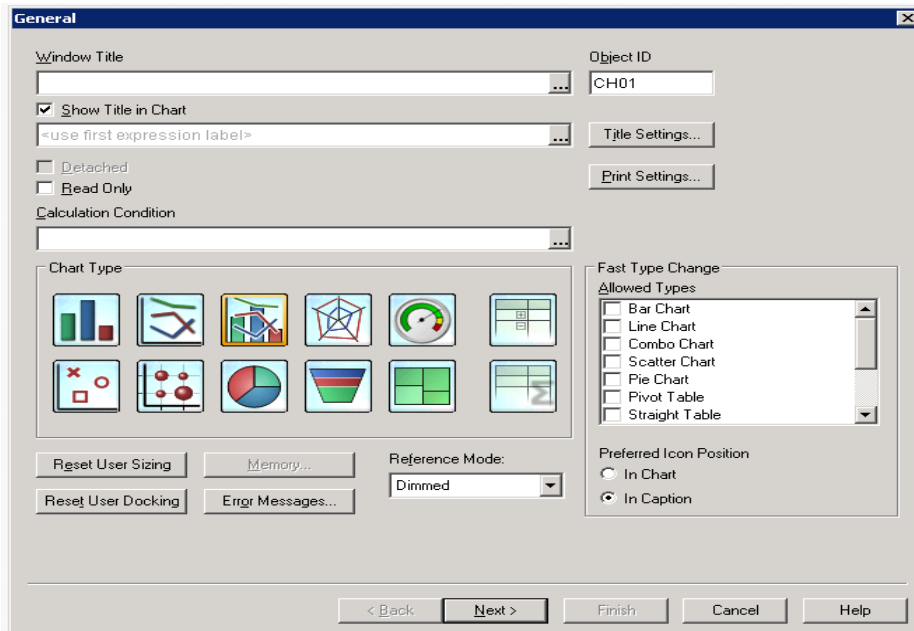
Dimension Definition: selection of the field "date_min_inclusion" which is the date of inclusion for all the patients who have been included within the study.

Formula Expression: completion of the formula to draw the desired curve (within expression): definition of the formula of the curve (see Figure 7) to display the number of intended inclusions. The code used is specific to QlikView, but has some similarities with formulas found in Microsoft Excel. In this field it must be defined whether the current filters should apply to the formula or not.

Curve Design: sets design and display of the curve such as the label (Number of intended inclusions), the colour (red), the symbol (none), the font and the size to be used for the text (arial / 12 pts), etc..

Other curves can be added by clicking on „Add Expression" and restarting the process.

For this application we simply needed to generate the theoretical curve versus real curve. It is of course quite possible to create any type of curve using this method.

All the objects available in QlikView are created according to the same process. The monitoring information tab presents the links between objects.

Stage 4:

PUBLISHING AND WEB DEPLOYMENT

One of our customer's need was to have a centralised application which is available to everyone, whilst limiting the access of information and data depending on the user profile.

The QlikView server can meet these requirements. A management console allows the developer to run the server and set the repository folder from which the applications will be displayed.

Registered users connect via a portal dedicated to QlikView and can have access to only certain information depending on their profile.

The management of user groups is available within the application, allowing access to certain tabs or certain data to be defined according to the user profile (studies, financial information, centres etc.). The developed application is filed in the Qlikview server repository folder and user rights are defined.

The QlikView Management Console also offers the possibility to define in what form(s) the application is available (AJAX, Java, IE plug-in or download). The various choices can cover a wide range of configurations at both the operating system level and web browser level.

CONCLUSION

In choosing QlikView to create this application we ensured that development was rapid and that maintenance costs are low. The customer now has an intuitive application which allows them to have an overview of each of their studies: either globally or in detail. Moreover, with the management of access rights, the use of this application is consistent with the pre-requisites of quality assurance.

QlikView has been proved to be an easy tool to use, both from the developer"s and user"s point of view, and is efficient and user-friendly. The solution provides interactive dashboards which are easily adaptable depending on requirements. However, improvements can still be made to the selection and movement of objects during the design of the application. Also, it is unfortunate that the command „where" is absent in the programming of expressions because it forces the developer to use pre-recorded filters.

In addition to the solutions given to the users, QlikView also has the advantage of being available on the Web and can therefore provide clients with information on their projects.