



# EXPRESSDESK REPORT

<http://www.express-desk.com/>

Résumé

Application développée en tant qu'étudiant durant les vacances d'été 2016

Adrien MOUSTY  
[moustyadrien@hotmail.com](mailto:moustyadrien@hotmail.com)

## Table des matières

Exemples d'écrans des différentes applications développées.....	2
Gestion des groupes.....	2
Objectif.....	2
Images.....	2
Query builder.....	3
But.....	3
Image.....	3
Code - Exemple de fonctions.....	5
JavaScript.....	5
C#.....	6
Résumé.....	7

## Exemples d'écrans des différentes applications développées

### Gestion des groupes

Durée de développement : une semaine.

#### Objectif

Les utilisateurs peuvent visionner les différents groupes existants dans la base de données. Quand ils cliquent sur un groupe, cela affiche toutes les personnes comprises dans celui-ci.

Un bouton permettant de changer de vue interverti les deux écrans.

#### Images

The screenshot displays two main sections of a web application. The top section is a table listing users with columns for email, first name, last name, and a language code, along with an 'Edit' button for each row. The bottom section is a 'Group' management interface. It features a table with 'Group Name' and 'Signed in' status, a search bar, and a dropdown menu for selecting a group. The footer contains links for 'Support' and 'Legal'.

Email	First Name	Last Name	Language	Action
sip.christer.gunnarsson4@ikea.com	Christer	Gunnarsson		Edit
sip.christopher.de.west@ikea.com	Christopher	de West		Edit
sip.david.camitz@ikea.com				Edit
sip.enes.alisa@ikea.com				Edit
sip.eva-lena.nystrom@ikea.com				Edit
sip.flavio.palacios@ikea.com				Edit
sip.ignacio.castro@ikea.com				Edit
sip.karan.honovar@ikea.com				Edit
sip.marika.hagstedt@ikea.com				Edit

Group Name	Signed in
English-GroupFunctions	false
English-Infrastructure	false
English-Multichannel	false
English-Other	false
English-Range	false
English-Retail	false
English-Supply	false

Footer: Support Legal

## Query builder

Durée du développement : six semaines.

But

Cette page permet au personnel de créer ses propres recherches dans la base de données via un système de SQL déguisé. L'utilisateur clique sur les éléments qui l'intéressent, ensuite, si la requête est correcte il peut choisir d'en afficher les résultats voire même d'afficher ceux-ci sous forme de divers graphiques.

Image

The image shows a web-based Query Builder interface with the following sections:

- Top Bar:** Contains four buttons: "Show" (with "Drag a value here"), "Number of" (with "CallAssigneeSIP"), "Average" (with "callDurationInService"), and "Minimum" (with "Drag a value here"). To the right is an "Add new boxes" button with checkboxes for "Average" (checked), "Minimum" (checked), and "Maximum" (unchecked).
- Available values:** A list of database fields including CallStartTime, CallEndTime, CallTargetTime, callStartTimeInQueue, callEndTimeInQueue, callStartTimeInService, callEndTimeInService, idCall, CallStatus, CallCatId, CallMainPurposeld, CallTargetExceeded, callDurationInQueue, CallDuration, CallerSIP, RemedyRefNr, RemedyTcktType, CallOnBehalfOfUID, CallAssigneeGroup, CallerSite, CallerDeviceId, CallGUID, originatorIPAddress, toastmessage, and Everything.
- Time unit:** A section with a "Show query result by" dropdown set to "On". It includes a checked "Add a time unit" checkbox, a "Hour" dropdown, and a "CallStartTime" dropdown.
- Filtering options:** A large section with several options:
  - Duration option:** Checked. Includes a "Lasting" dropdown set to "CallStartTime", a "Between two dates" dropdown, and two date input fields: "2016-February-01" and "2016-March-31".
  - In depth option:** Unchecked.
  - Having:** Includes a table with columns: Operation from, Element name, Operation, and Your value. The row shows "COUNT", "CallAssigneeSIP", "IS NOT NULL", and an empty dropdown.
  - Order my search by:** Unchecked.
- Advanced settings:** A bottom section with a "<> Advanced settings" header and three buttons: "Advanced", "Save as...", and "Show data".

Filtering options

☒ Duration option

Lasting

CallStartTime

2016-February-01

Between two dates

2016-March-31

☐ In depth option

Having

☒ Operation filter

Operation from

Element name

Operation

Your value

COUNT

CallAssigneeSIP

IS NOT NULL

Just figures are allowed, take care.  
 If you use the between clause, please separe values with ','.

☐ Order my search by

</> Advanced settings

Advanced
Save as...
Show data

```

SELECT DISTINCT hourofday_table.hourID AS Hour, COUNT_CallAssigneeSIP, AVG_callDurationInService
FROM expressdesk.dbo.hourofday_table
LEFT OUTER JOIN
(
SELECT COUNT(CallAssigneeSIP) AS COUNT_CallAssigneeSIP, AVG(callDurationInService) AS AVG_callDurationInService, DATEPART(Hour, CallStartTime) AS timeid
FROM expressdesk.dbo.call
WHERE CallStartTime BETWEEN '2016-February-01' AND '2016-March-31'
GROUP BY DATEPART(Hour, CallStartTime)
HAVING COUNT(CallAssigneeSIP) IS NOT NULL
)
myquery ON hourofday_table.hourID = myquery.timeid
ORDER BY hourofday_table.hourID;
        
```

Table field

Table field

Hour	Count CallersSIP	AVG CALLDURATIONINQUEUE	AVG CALLDURATIONINSERVICE	AVG CALLDURATION	MIN CALLENDTIMEINSERVICE
0	25	44	220	252	8/10/2015 12:38:31 AM
1	4			71	4/17/2016 1:17:31 AM
2					
3					

Chart type:
 

Dual-Axis Line and Column

Configure Chart...

Display/Refresh Chart

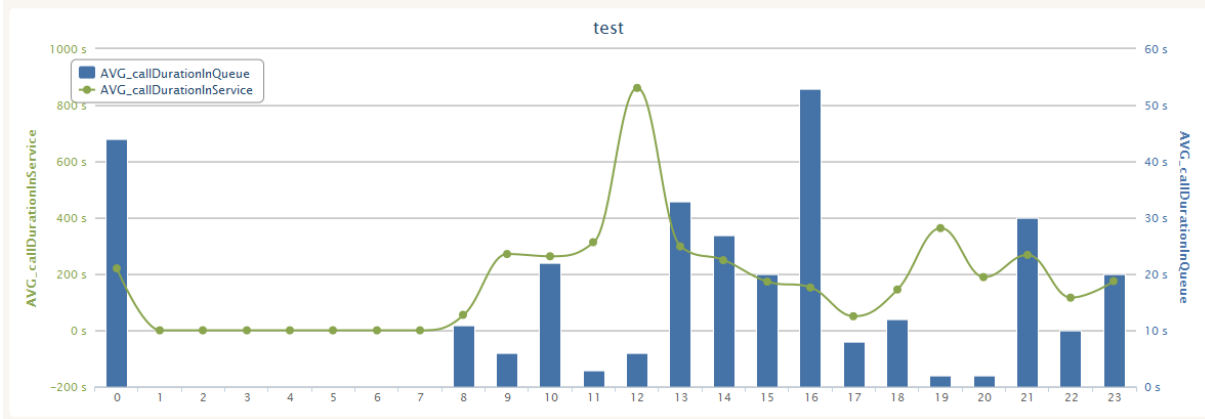
Open Chart...

Export to Excel format...

Save Chart...

Clear all values

Note Please configure the chart
×



## Code - Exemple de fonctions

La difficulté réside surtout dans le fait de pouvoir transmettre les informations correctement entre le client et le serveur.

De plus, comme le projet est relativement long, il faut également que tous les éléments restent cohérents entre eux.

### JavaScript

```
function clearArrayAndRemoveLiContent(arrayName, tmpArrayName) {
    while (arrayName.length != 0) {
        var tmp = tmpArrayName.search("Array"); // will get the index start of array within the string "selectArray"
        var divName = tmpArrayName.slice(0, tmp); // result -> select
        var itemName = arrayName[arrayName.length - 1];
        $('#'+ itemName + '-dropped').hide("fade");
        $('#'+ itemName + '-dropped').remove(); // Remove the "li" created
        adjustSizeAndPushArray('-', divName + "-son", itemName); // decrement the array
        $('#'+ itemName).show("fade"); // show the draggable who's deleted
    }
    $('#'+ divName + '-son > label').show("fade");
}

// This function will also called for the configuration restoration
function droppableDrop(currentRecipient, currentDraggableHTML) {
    var currentDraggableID = $('#'+ currentDraggableHTML);
    currentDraggableID.hide("fade"); // move the dragged item
    $('#'+ currentRecipient).append('<li id="' + currentDraggableHTML + '-dropped"><table><tr><td>' + currentDraggableHTML + '</td><td><span>'
    $('#'+ currentRecipient + '> label').hide("fade");
    adjustSizeAndPushArray('+', currentRecipient, currentDraggableHTML); // we increment the size of the div who is dropped
    // because we can't use a select clause with this kind of query, it will throw an error
    if (currentRecipient === "select-son") {
        $('#chkB_timeUnit').prop('checked', false);
        $('#showTimeUnit').hide("fade");
    }
}

/*****/

// make the trash icon clickable
$('#'+ currentDraggableHTML + '-del').click(function (event) {
    $('#'+ currentDraggableHTML + '-dropped').remove(); // Remove the "li" created
    adjustSizeAndPushArray('-', currentRecipient, currentDraggableHTML); // decrement the array
    currentDraggableID.show("fade"); // show the draggable who's deleted
    // We see if the operation div is empty, if yes we show the label "drag me here"
    if ($('#'+ currentRecipient + ' li').length < 1) {
        $('#'+ currentRecipient + '> label').show("fade");
    }
});

/*****/
}
```

```

// We catch the control who have caused a postback, if it is related to the chart, we'll use theses ids
string ctrl1 = GetPostBackControlId();
if (ctrl1 == "ChartTypeDropDown" || ctrl1 == "GencartButton" || ctrl1 == "OpenDefStn" || ctrl1 == "saveDefStn" || ctrl1 == "SaveAsChartName")
{
    // We have to create a tmp query in the database in order to restore it
    if (!string.IsNullOrEmpty(hiddenPassingInformation.Value.ToString()))
    {
        DataTable tmpDT;
        if (ctrl1 != "SaveAsChartName")
        {
            txtB_queryDescr.Text = "tmpQueryCreatedInOrderToRestoreConfiguration";
            registerTheQuery();
            querybox.Text = buildTheQuery();
            tmpDT = DBInterface.Instance.GetIdTmpQuery();
            if (ChartTitle.Text == "" || ChartIdTextInput.Text == "")
            {
                refreshTableAndGraphConf(tmpDT.Rows[0].ItemArray[1].ToString());
            }
            // SaveAsChartName triggered the function, in this case the user pressed "save changes". We need to store a tmp query in order to draw the chart if everithing else is ok.
        }
        else
        {
            tmpDT = DBInterface.Instance.GetIdTmpQuery();
            if (tmpDT.Rows.Count < 1)
            {
                // A tmp name, normally nobody use this name for a description
                txtB_queryDescr.Text = "tmpQueryCreatedInOrderToRestoreConfiguration";
                registerTheQuery();
                querybox.Text = buildTheQuery();
                tmpDT = DBInterface.Instance.GetIdTmpQuery();
                if (DBInterface.Instance.SelectSavedChart("tmpQueryCreatedInOrderToDrawTheChart").Rows.Count < 1)
                {
                    // there is no existing tmp query, we need to create a new one
                    DBInterface.Instance.SaveChartAndQuery(SaveAsChartName.Text, ChartTitle.Text, ChartSubtitle.Text, ChartTypeDropDown.SelectedValue, querybox.Text, "0", CategoriesDropDown);
                }
                // The chart is drawn
                prepareDrawingChart(DBInterface.Instance.SelectSavedChart("tmpQueryCreatedInOrderToDrawTheChart"));
                // Tmp query who draw the chart is deleted
                DBInterface.Instance.DeleteTmpChart();
            }
        }
        displayDefaultScreen("");
        // We restore the query configuration (because a postback is triggered and a postback make the page to this original statement)
        restoreTheQuery(tmpDT.Rows[0].ItemArray[0].ToString());
        // We don't need anymore the first tmp query, it's deleted
        DBInterface.Instance.DeleteTmpQuery();
        txtB_queryDescr.Text = "";
    }
    else
    {
        // No more explanations
        throw new ArgumentException("An error happened");
    }
}

```

## Résumé

- Travail effectué en « freelance » durant 7 semaines.
- Mon patron demandait certains éléments à implémenter et je lui fournissais.
- Gestion du support vis-à-vis des applications développées.
- Utilisation de machines virtuelles, de VPN et de différents serveurs.
- Premier travail à responsabilité.
- Travail effectué principalement en Anglais + autoformation en ASP, JQuery.
- Travail de documentation au préalable, afin de voir ce que la concurrence pouvait proposer et ainsi pouvoir faire mieux.
- Utilisation de nouveaux langages et logiciels.
- Langages utilisés : C#, ASP, JS, JQuery, HTML, CSS.
- Principal client : IKEA. Communication avec le client.
- Mise en place d'une documentation pour chaque application développée.
- Premier travail en tant que salarié, beaucoup d'expérience emmagasinée.