Data Grid UI Specification

Version 0.1

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Change | Date |
| 0.1 | Pierre Derval | Initial draft created | 05.05.2018 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1.0 Introduction 5](#_Toc503449802)

[2.0 List types 7](#_Toc503449803)

[3.0 Common Interactions 7](#_Toc503449804)

[3.1 Selection 7](#_Toc503449805)

[3.2 Multi-Selection 7](#_Toc503449806)

[3.3 Read-only 8](#_Toc503449807)

[3.4 Locked 8](#_Toc503449808)

[4 Table View 9](#_Toc503449809)

[4.1 Visual Representation 9](#_Toc503449810)

[4.2 Basic Specifications 11](#_Toc503449811)

[4.2 No data in a Table View 11](#_Toc503449812)

[4.3 Selection 12](#_Toc503449813)

[4.3.1 Selectable 12](#_Toc503449814)

[4.3.2 Multi-Selectable 12](#_Toc503449815)

[4.3.3 Selecting a row 13](#_Toc503449816)

[4.4 SlidePanel behaviour 13](#_Toc503449817)

[4.5 Resizing a column 13](#_Toc503449818)

[4.6 Sorting a column 14](#_Toc503449819)

[4.7 Replacing a column with a column which is currently not visible 14](#_Toc503449820)

[4.8 Scrolling the list 15](#_Toc503449821)

[4.9 Responsiveness 15](#_Toc503449822)

[4.10 Error cases 15](#_Toc503449823)

[5 Icon list 16](#_Toc503449824)

[5.1 Responsiveness 16](#_Toc503449825)

[5.2 Error cases 16](#_Toc503449826)

[6. Tile list 17](#_Toc503449827)

[6.1 Error cases 17](#_Toc503449828)

[7 Display text summary 18](#_Toc503449829)

# 1.0 Introduction

A data grid is used to organize data, so it becomes easy for business users to browse and quickly get an overview of each piece of information that it contains.

The Data grid always show a LIST of business object. Those business object can be a person, an organization, an email.

In uxi, we generally refer Business Object to ‘Entity’.

Simply put, a Data grid is the representation of a list of entities.

A data grid can be used in multiple context. As a main element for a page, inside a dialog, inside a slide panel component, in a tab.

The data grid is part of the ‘strategic’ components as he is a main driver for the UX strategy.

**NOTE:**

**Can I use a Data Grid for a Search Result?**

Yes, you should

**But you say the Data Grid can only be used for one Entity? In a search, I can have multiple entities such as Person and Organization?**

No, in that case you have an entity called SearchResult. So, the guidelines will still be valid, it is a List of entity called SearchResult.

*Disclaimer:*

*All screenshots in this document are examples only. Their purpose is to demonstrate the functionality and interaction specified in the document. Implementation and test of all visual designs must be done according to the* UI guideline.

# 2.0 List types

4 types of lists exist:

* Table View
* Tile View
* Detail View
* Icon View

Each view must support Selection and MultiSelection.

# 3.0 Common Interactions

Each View for the Data Grid should achieve the same ‘goals’.

For now, Select and Multi-Select are the only shared interactions.

The look and feel of those interactions can variate depending on the view but the UX goal will always remain, whatever the common interaction is.

As such, each view needs to be able to:

* Select a single Entity
* Select multiple Entities
* Mark an Entity as Readonly (no actions are possible, no edition possible)
* Mark an Entity as Locked (cannot be selected)

## 3.1 Selection

Selection is an interaction where the user wants to select only one Entity inside the DataGrid.

When using Selection, we should not show any option to ‘Select All’ Entities.

The component needs to enforce the single selection by disabling all the other selector element once one Entity has been selected. The component should offer a way to remove the selection by re-clicking on the selected Entity. That would toggle the selection.

## 3.2 Multi-Selection

Multi-selection is an interaction where the user wants to select multiple Entities inside the DataGrid.

When using Multi-selection, we should always show an option to ‘Select All’ entities in a single click. The Select All should add a toggle to select/unselect all the Entities at once.

The component can enforce some validation by disabling the cell if required. Ex: can only select 3 Entities in the List of Entity.

## 3.3 Read-only

Read-only is used to draw attention on an Entity that it cannot be modified and/or that no action can be performed. For example, if you enable ‘inline-editing’ for the Data-Grid, it would not be available in a Read-only. A read-only entity can be selected.

## 3.4 Locked

Locked is used to draw attention on an Entity that it cannot be modified and selected. The Entity is ‘locked’, this means you can see the information, but you are not able to perform any kind of action under that Entity. A locked Entity cannot be selected.

# 4 Table View

Table view is used to show many data inside a Data Grid.

The table view is generally used when you need to compare multiple Entities and differentiate which one you want based on extra properties.

For example, let’s say you have multiple persons with the same name, the extra properties how in the Row will help the user to differentiate the rows between each other. It will also allow the user to either select the appropriate Entity or view more details about it.

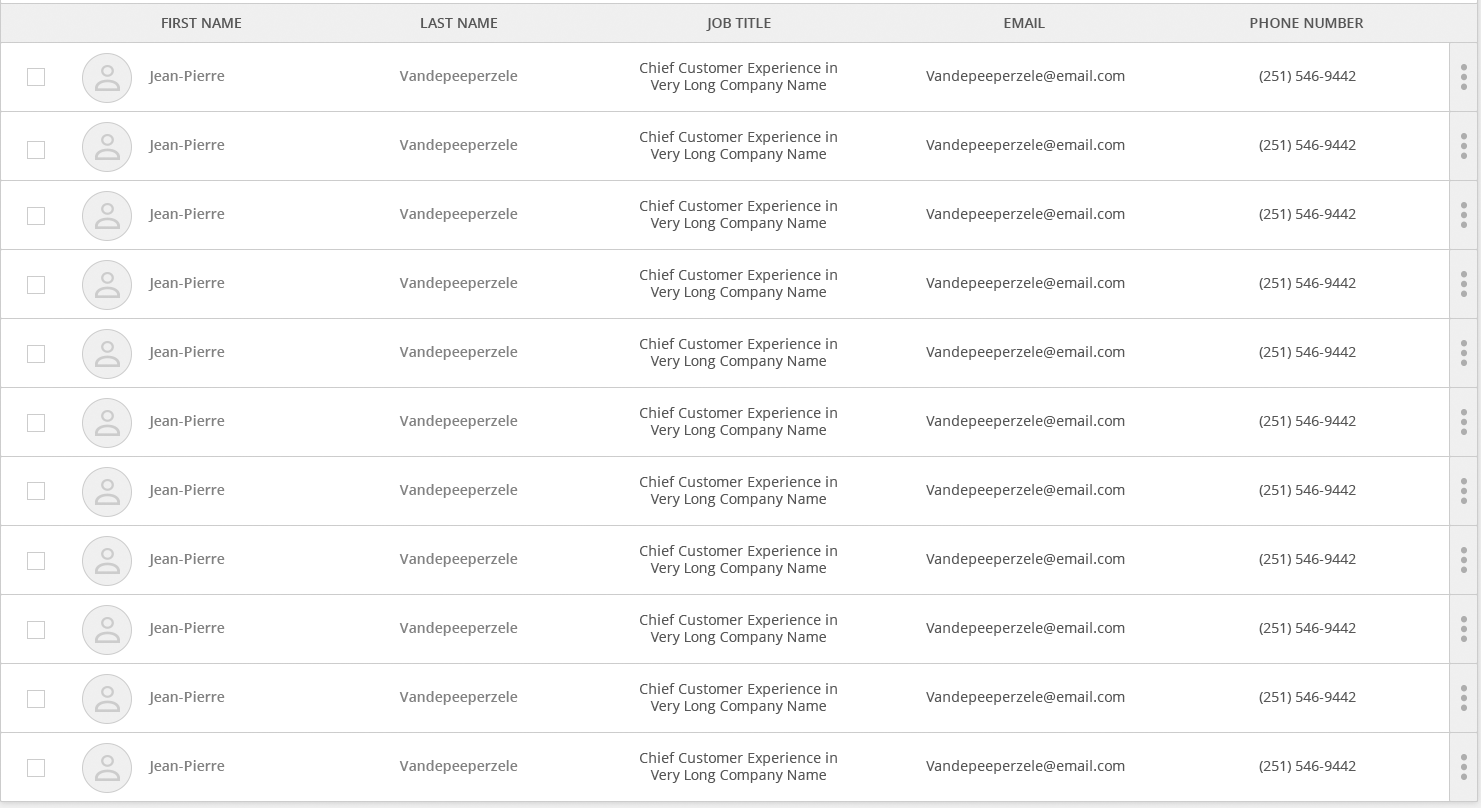
The table view is also need when you need sorting capabilities based on some Property. For example, you have a list of ‘Product’ and want to sort them by Price. Just by clicking on the ‘Price’ header of the Table, you could sort them (see sorting specification).

The table view is also used when the User wants to define a list of properties that he wants to see for a list of Entities. As the table is more ‘generic’ than the other view, it is an ideal view when you want your users to define what they want to see for a specific Entity in a list.

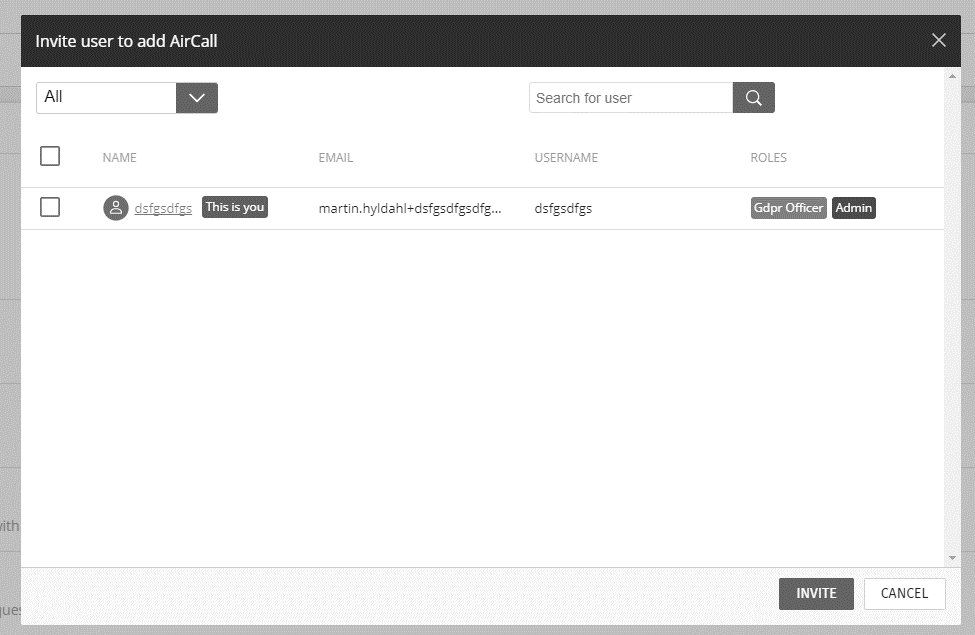
## 4.1 Visual Representation

*Figure 1: Example of a table view on a Data Grid. “Name” column contains hyperlinks to each entity’s task page.*

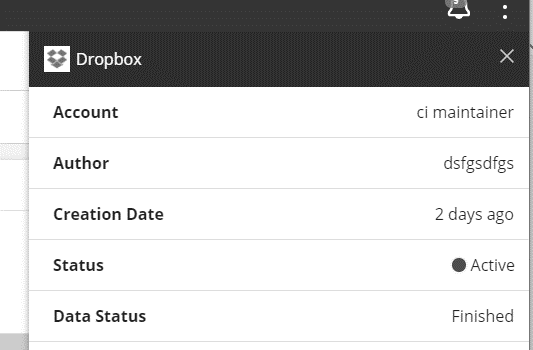
*Selection Activated*



*Figure 2: Example of a detail list in a dialog. “Name” contains hyperlinks – Multi-select activated.*

**

*Figure 3: Example of a detail list in a smartpanel. The “Name” column contains hyperlinks.*

**

## 4.2 Basic Specifications

* A Table view consists of a number of rows and columns.
* Each column has a title and optionally a sorting button.
* A column title can be empty.
* A tooltip shows the entire column title when hovering the mouse over the column title area.
* If a column title is longer than the column width, the title is truncated at the end and an ellipsis is added.
* A column title is a Text and/or an AvatarWithName

A column in a detail list can contain one of the following components:

* Text string
* AvatarWithName component
* Image (see image documentation for more information)
* A Link
* Supported Visualization (see visualization guidelines)
* A column MUST always show only 1 Entity (no multiple entity information in a single column)
* Alignment in cell can variate from left, center, right depending on the scenario.
* To emphases some content inside a column, it must use ‘bold and italic’.
* Badge can be used inside a cell to draw extra attention.

## 4.2 No data in a Table View

When there is no data in a row, we should always show an Alert to pick the attention of the user that the list is empty.

By default, the error message should be: **No data available.**

Data can be changed with the type of the entity. For example, for an Entity ‘Person’, it would be ‘No person available’.

*Figure 4: Example of a detail list with no data available.*

//TODO:

## 4.3 Selection

A table can be Selectable and Multi-Selectable.

Each row in the table will have a checkbox. The header of this new column will either be empty (Selectable) or contain the Select All Checkbox (Multi-Selectable).

### 4.3.1 Selectable

A user can check/uncheck individual rows by using the checkboxes in front of each row.

* When a row is checked, the list enters check mode (//todo to verify this) which means that clicking/touching anywhere (except on hyperlinks) on another row will check that row (or uncheck it if it is checked).
* Hyperlinks are active in check mode.
* Activating a hyperlink does not select or check the row.
* Selections of rows can be stored and persisted if needed.
* Sorting columns is possible in check mode.

*Figure 5: Example of a selected row (the blue row).*

* Selecting a row is done by clicking/touching an area (which is not a hyperlink) on a row.
* One and only one row is always selected.
* First row in a list is default selected.
* When user selects another row, the previously selected row is automatically unselected.

### 4.3.2 Multi-Selectable

All points mentioned in the Selectable section is still applicable for the multi-select.

* If all rows are checked manually, the select all checkbox in the header should be checked.
* If all rows are checked and the user unchecks a single row, the select all checkbox in the header should be unchecked.

The click target for the checkbox for checking/unchecking all rows is not just the checkbox but includes some space around it to accomodate a finger press on touch.

The checkbox for checking/unchecking all rows has a tooltip “Select all rows” shown on mouse hover when not all rows are checked.

* The tooltip is ‘Select all rows’ when all rows are checked.
* The tooltip is “Deselect all rows” when all rows are checked.
* The tern ‘rows’ can be changed with the Entity Type name. (Select all persons, Deselect all Items…).

### 4.3.3 Selecting a row

Selecting a row means putting it in focus. It is not the same as checking a row. Selecting allows a user to quickly click/touch through a list and see details of individual rows in a slide-panel. Selecting is available when a list is not using Selectable or Multi-Selectable.

### 4.4 SlidePanel behaviour

When you have a SlidePanel showing when you ‘select’ a row. The slidepanel should only show once the user has selected a row. Before that, the SlidePanel is hidden.

* When using a SlidePanel along with the DataGrid, the SlidePanel MUST always show more information about the selected row (aka Entity) and nothing else.
* Use can close a smartpanel by clicking the “close” icon (“x”) in the top right corner of the slide panel.
* When a user selects or checks a row again, the Slidepanel reappears and remains visible until user closes it again.

### 4.5 Resizing a column

user can resize column widths by dragging the vertical separator line between two columns.

*Figure 6: Example of a vertical separator line (inside the red circle).*

* Content in a cell does not wrap to more lines if the column width is made smaller – for cells with only textual content, the content will be truncated and an ellipsis added.
* A tooltip shows the entire textual content when hovering the mouse over the cell containing truncated content.

Note that a column has a minimum width – see the [Visual Design Guideline](http://intranet.dk.sitecore.net/Project-Rooms/UX/Visual-Design/Guidelines.aspx) for details.

### 4.6 Sorting a column

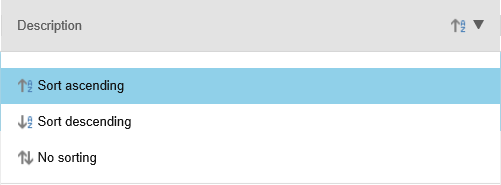
A sorting indicator will appear on a column if sorting is available for that specific column.

Sorting can be in one of three states:  no sorting applied (default),  descending sorting, ascending sorting. The states are shown in the following order and with the following display texts:

* “Sort ascending”
* “Sort descending”
* “No sorting”

Progress bar columns are sorted with the most progressed entities on top in case of descending sorting.

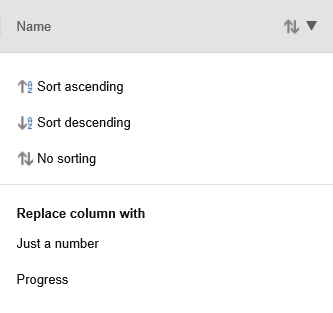
When user clicks/touches anywhere in a column title area, a drop-down menu is shown where user can choose among the three sorting states. If sorting is not enabled for a column, no sorting indicator is shown and nothing happens if user clicks/touches the the column title area unless the detail list contains columns that currently cannot be seen on the screen.



*Figure 6: Example of a drop-down menu for selecting sorting state*

### 4.7 Replacing a column with a column which is currently not visible

If the detail list contains more columns than what can be seen on the screen, it is possible for user to replace a visible column with one of those columns that currently cannot be seen. This is done by clicking/touching the column title area of a given column and then choosing from the shown drop-down menu which column to replace the selected column with.



*Figure 7: Example of a drop-down menu for selecting sorting state or which column to replace the selected column*

* If the selected column can be sorted, the list of currently not visible columns is shown after the sorting options and beneath the title “Replace column with” – otherwise, only the list of currently not visible columns is shown beneath the title.
* The column which is now visible inherits the column width of the column that it is replacing.
* The titles of the columns that are currently not visible are used as display texts in the drop-down menu.
* An arrow icon  is shown in the column title area as soon as there is a drop-down available (sorting is enabled and/or the column can be replaced).

### 4.8 Scrolling the list

Detail lists are using infinity scroll. Infinity scroll prevents the scroll bar from scrolling to the bottom of the page, causing the page to grow with additional content instead.

If a column contains images, user sometimes has to wait a short time before an image can be seen after having scrolled down. This can be the case when it takes some time to load the image. Until the image is loaded and displayed, an image placeholder icon is displayed (see the [Visual Design Guideline](http://intranet.dk.sitecore.net/Project-Rooms/UX/Visual-Design/Guidelines.aspx) for details if the image placeholder icon).

### 4.9 Responsiveness

When reducing the width of the browser window, the rightmost visible column is dropped off once it reaches its minimum size (refer to the [Visual Design Guidelines](http://intranet.dk.sitecore.net/Project-Rooms/UX/Visual-Design/Guidelines.aspx)) and user continues to reduce the width of the browser window – then follows the same procedure for the new rightmost visible column and so on. When the viewport reaches a certain minimum size (refer to the [Visual Design Guidelines](http://intranet.dk.sitecore.net/Project-Rooms/UX/Visual-Design/Guidelines.aspx)), the detail list gets transformed into a tile list.

### 4.10 Error cases

In case an image cannot be loaded e.g. due to the image being corrupt or the image simply cannot be found, a browser-specific error icon will be placed on top of the image placeholder icon.

## 5 Icon list

*Figure 8: Example of an icon list on a list page.*

An icon list works in many ways exactly like a detail list, however there are a few differences.

* An icon list does not contain columns – even though the icons are organized in rows and columns.
* Checkboxes for checking icons exist here too (if enabled by the developer) but they are placed differently on an icon list – each icon has a checkbox in the top left corner of the icon (see [Visual Design Guidelines](http://intranet.dk.sitecore.net/Project-Rooms/UX/Visual-Design/Guidelines.aspx) for details).
* A checkbox for checking all icons in the list is placed in the far left side of the icon list header area.
* The same 3 icon sizes are supported in icon lists as in detail lists.
* Icon labels are placed below each icon and they are left-aligned.
* An icon label does not wrap – if it is longer than the width of the icon, the label is truncated and an ellipsis is added.
* A tooltip shows the entire icon label when hovering the mouse over the truncated label.
* An icon label is a text string.

When user opens an icon list, image placeholder icons will be shown for all images until they are loaded. In many cases, images are loaded so fast that user will never see the image placeholder icons. user can scroll the list while images are being loaded.

### 5.1 Responsiveness

When reducing the width of the browser window, the content of the icon list is wrapped. Content will be gradually wrapped until a single column is visible.

### 5.2 Error cases

In case an image cannot be loaded e.g. due to the image being corrupt or simply cannot be found, a browser-specific error icon will be placed on top of the image placeholder icon.

## 6. Tile list

*Figure 9: Example of a tile list on a list page.*

The tile list works similarly to an icon list – there are only a few differences.

* Tiles do not have labels placed below their frame.
* The content and layout inside a tile can differ from application to application – however, the [Visual Design Guideline](http://intranet.dk.sitecore.net/Project-Rooms/UX/Visual-Design/Guidelines.aspx) should be followed.
* The width of all tiles is fixed but can vary from application to application.
* The height of a tile may vary from tile to tile depending on the content inside each tile.

### 6.1 Error cases

In case an image cannot be loaded e.g. due to the image being corrupt or simply cannot be found, a browser-specific error icon will be placed on top of the image placeholder icon.

## 7 Display text summary

“No data available” - see page 7  
“Select all rows” - see page 7  
“Deselect all rows” - see page 8  
“Sort ascending” - see page 9  
“Sort descending” - see page 9  
“No sorting” - see page 9  
“Replace column with” - see page 10