

## Spécification et analyse des besoins

### Gestion de Dashboard Sourcing force

#### RESUME

Ce document est une note technique. Il est Restreint - Divulgateur ou transfert possibles après approbation de Sofia Technologies.

#### LISTE DE DISTRIBUTION

Nom	Rôle	Organisation
Mouna Trabelsi	Team leader	Sofiatech
Anwer lamouchi	Ingénieur logiciel	Sofiatech
Fahmi elkamel	Stagiaire ingénieur logiciel	Sofiatech

#### HISTORIQUE DES MODIFICATIONS

Version	Modifications	Nom		Date
1	Création	Auteur	Fahmi elkamel	06/03/2020
		Vérificateur	Anwer lamouchi	09/03/2020
		Approbateur	Mouna Trabelsi	

## SOMMAIRE

1. OBJECTIF .....	3
2. BESOIN FONCTIONNEL : .....	3
3. IDENTIFICATION DES ACTEURS : .....	4
4. DIAGRAMME DE CAS D'UTILISATION .....	6
5. SCÉNARIO DE CREATION D'UN WIDGET : .....	6

## 1. Objectif :

La gestion de Dashboard est un outil d'évaluation de l'organisation d'une entreprise ou d'une institution constituée de plusieurs indicateurs.

Notre objectif est de créer un système de gestion Dashboard sous forme de Widgets génériques, dynamiques et personnalisables il a pour fonction de permettre la visualisation, le suivi et l'exploitation facile de données pertinentes sous forme de chiffres, données et de graphiques.

Notre travail consiste essentiellement à développer trois modules principaux :

- ✓ Un module pour la gestion de Dashboard
- ✓ Un module qui permet de générer des widgets génériques et dynamiques
- ✓ Un module qui permet de configurer et personnaliser les widgets

## 2. Besoin fonctionnel :

- ❖ Créer un Dashboard sous forme de Widgets.
- ❖ Créer un générateur des widgets qui permettent de générer des widgets (standard ou spécifique) des différents types. Plusieurs types sont possibles pour présenter les informations : liste, graphique et les indicateurs avec ou sans liste. Les données indicatrices, liste et les chiffres de chart graphique que contient Dashboard doivent répondre à certains critères préalablement sélectionnés lors de la création du widget :
  - **Widget spécifique** : les widgets spécifiques représentent des fonctionnalités spécifiques grâce à des critères spécifiques pour chaque client.
  - **Widget standard** : les widgets standard doivent répondre à des fonctionnalités standard par des critères standard disponibles par tous les utilisateurs.
  - **Les types du widget :**
    1. **Type Graphique** : Les graphiques pouvant figurer dans le Dashboard sont nombreux, dont voici quelques-uns : des Courbes (Line chart), des Diagramme circulaire, les histogrammes et etc.

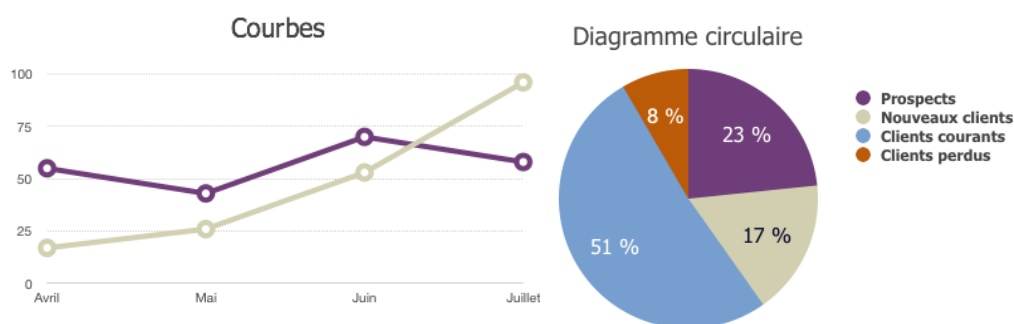


Figure 1 : Type Graphique

2. **Type liste** : Les données seront récupérées dynamiquement dans une liste selon des critères donnés lors de la création du widget.
3. **Type indicateur** : un widget qui contient un indicateur selon des critères donnés lors de la création du widget. Voilà un exemple de widget indicateur

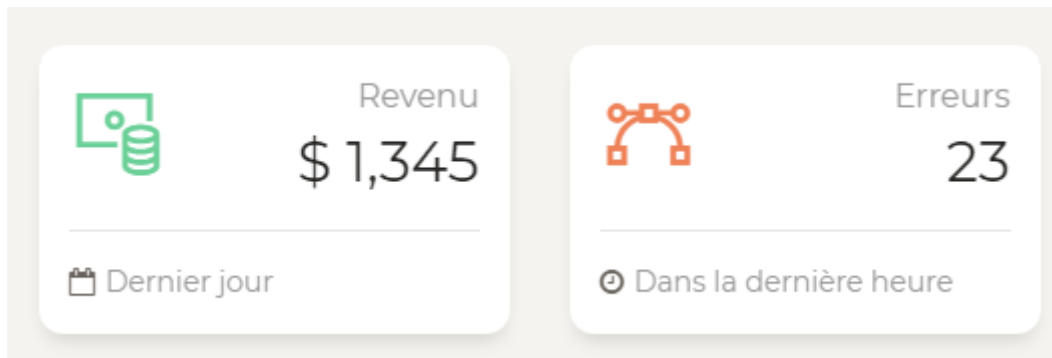


Figure 2 : Widget indicateur

4. **Type indicateur avec ou sans liste** : un widget qui contient un indicateur avec une liste des données selon des critères donnés lors de la création du widget.

- ❖ Chaque widget doit répondre à un besoin bien défini qui est fixé par des critères sélectionnés à sa création.
- ❖ Chaque widget doit lier à un ou plusieurs rôles sélectionnés lors de la création du widget
- ❖ Les widgets seront livrés à l'utilisateur via les rôles.
- ❖ Chaque widget est identifié par un texte et une description, on peut mettre une ou plusieurs langues pour chaque widget.
- ❖ Chaque widget doit être personnalisable par l'administrateur ou par l'utilisateur :
  - Il peut changer leur disposition par glissement (drag and drop)
  - Il peut Changer le style de texte et la couleur
  - Il peut changer la couleur de background
  - Il peut définir un style pré défini
  - Sauvegarder les modifications pour chaque utilisateur
- ❖ L'utilisateur a la possibilité d'activer ou désactiver des widgets.
- ❖ Chaque utilisateur a la possibilité de créer un compte.
- ❖ L'administrateur a la possibilité d'activer ou désactiver un compte utilisateur.
- ❖ Gestion de transaction : l'administrateur a la possibilité de gérer les transactions commande

### 3. Identification des acteurs : Les acteurs qui interagissent avec le système sont le rôle utilisateur et le rôle super administrateur :

#### ► Partie utilisateur :

- ❖ Créer compte
- ❖ Authentification
- ❖ Consulter widget disponible
- ❖ Activer Widgets

- ❖ Désactiver Widgets
- ❖ Déplacer Widget
- ❖ Configurer widget :
  - Changer disposition
  - Changer background
  - Changer style Texte :
    - ✓ Changer Couleur
    - ✓ Changer Front
    - ✓ Changer Format
    - ✓ Changer Taille
    - ✓ Changer Alignement
    - ✓ Sauvegarder modifications
- ❖ Créer des styles : Chaque utilisateur peut créer son style de widget et l'applique sur ses widgets.

► **Super Administrateur :** Peuvent aussi gérer toutes les fonctionnalités de gestion d'utilisateur

- ❖ Générer widgets : voir l'exemple ci-dessous d'un scénario de création widget. (Page 6)
- ❖ Gérer widget :
  - Consulter
  - Dupliquer
  - Modifier
  - Supprimer
- ❖ Gérer Langue :
  - Ajouter
  - Modifier
  - Supprimer
- ❖ Gérer transaction
- ❖ Gérer user

## 4. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est une représentation graphique qui décrit les fonctionnalités d'un système. Il permet d'identifier l'interaction entre le système et les acteurs. La figure ci-dessous représente le diagramme de cas d'utilisation globale

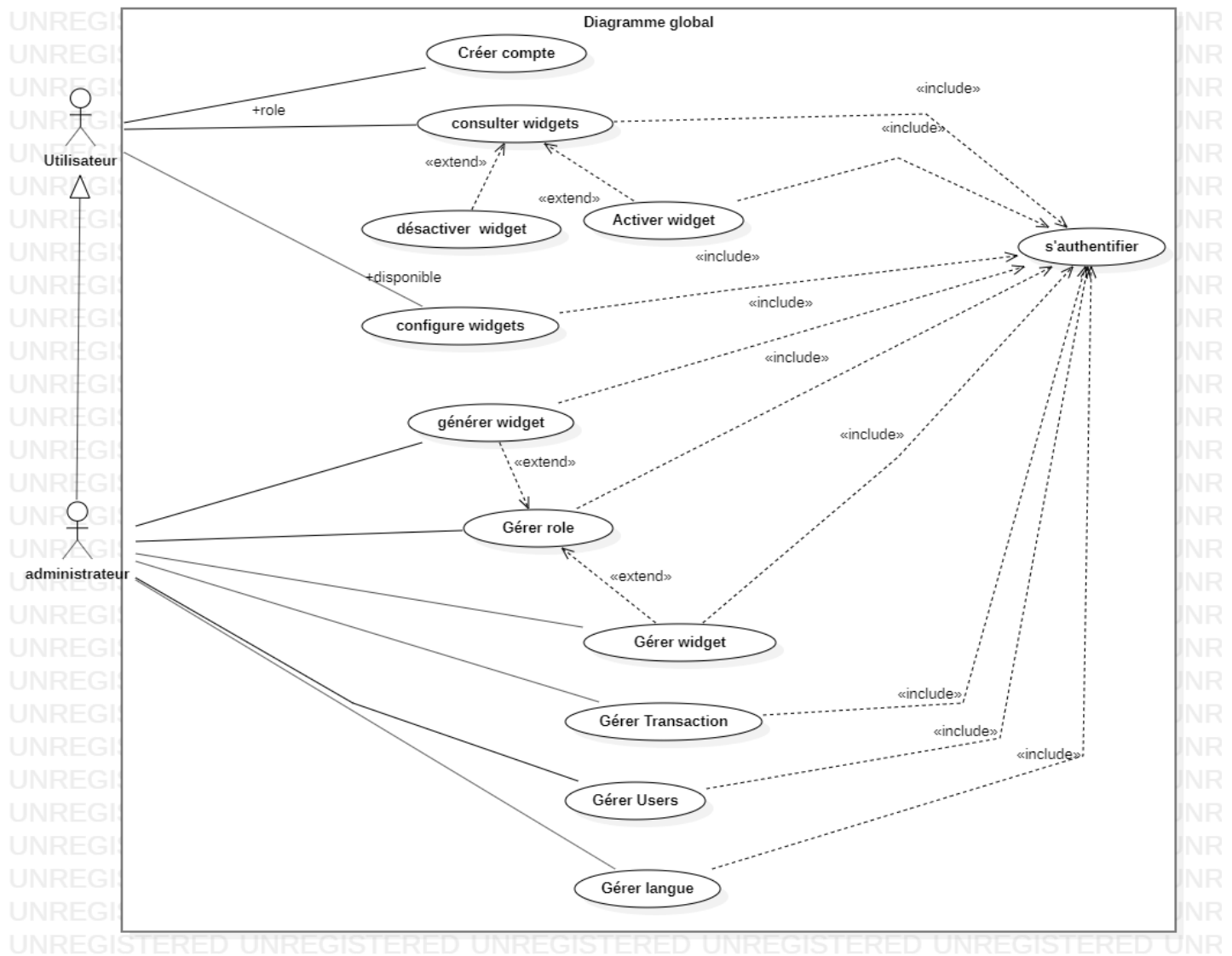


Figure 3 : Diagramme de cas d'utilisation global

## 5. Scénario de création d'un widget

. Le widget est créé sur 5 étapes :

- Saisir de données
- Créations des filtres
- Éditer l'apparence de widget
- Gérer les rôles.
- Teste widget

Voilà un exemple de créer un widget (liste) pour récupérer toutes les transactions qui l'ont créé après 27/02/2020

**Étape 1 :**

- Sélectionner ou saisir les données suivantes :
- Titre de widget
- Langues de titre
- Utilisation Standard/Custom
- Type d'affichage : liste, graphique, indicateur
- Type de données à utiliser : Transactions / Core data /Fournisseurs
- En clique sur bouton suivant

The interface shows the 'Saisir les données' (Enter data) step for creating a widget. It features five tabs: 'Saisir les données', 'créer filtre', 'Editer l'apparence', 'gérer les rôles', and 'Test'. The 'Saisir les données' tab is active. It contains three main sections: 'donnée' (data) with fields for 'Titre' (Title) and 'langue' (Language) set to 'français'; 'entité' (entity) set to 'transaction'; and 'widget' (widget) with 'Type' set to 'liste'. At the bottom, there are two radio buttons for 'standard' and 'custom' (selected), and a 'suivant' (next) button.

Figure 4 : interface saisir données

**Étape 2 : Créations des filtres**

- Il faut choisir filtre
- Il faut choisir les critères
- Il faut choisir les conditions
- Saisir la valeur.
- Cliquez sur le bouton Suivant

Saisir les données   créer filtre   Editer l'apparence   gérer les rôles   Test

filtre   date

propriete   Date\_creation   after   27/02/2020   ET   ou   +

id  
status  
fournisseur  
client  
Date\_creation

after  
before  
strictly\_after  
strictly\_before

suivant

Figure 5 : interface création filtres

- Remarque : vous pouvez ajouter une autre condition en appuyant sur le bouton plus.

### Étape 3 : Editer l'apparence de widget

Dans cette étape l'administrateur doit créer ou choisir un style pour le widget

Saisir les données   créer filtre   Editer l'apparence   gérer les rôles   Test

apparence

taille   3   front texte   format   style prédefeni

Arial  
Calibri (Body)

gras  
italique

background   titre

suivant

Figure 6 : interface configuration style widget



**Étape 4 : Gérer les rôles.**

The interface displays a top navigation bar with five icons: 'Saisir les données', 'créer filtre', 'Editer l'apparence', 'gérer les rôles' (highlighted in red), and 'Test'. Below the navigation bar is a form with a label 'rôle' and a dropdown menu. The dropdown menu is open, showing three options: 'directeur', 'administrateur', and 'techniciens'. A 'suivant' button is located at the bottom right of the form.

**Figure 5 : interface Gérer les rôles.**