



Magdalena Larmet
Salesforce
Developer



MIGRER UNE APPLICATION **SALESFORCE**



Sommaire

- 1** Rappel du projet
- 2** Mettre à jour un Lead
- 3** Rechercher une opportunité
- 4** Captures d'écran
- 5** Démo et Conclusion



1

Rappels du **PROJET**

Exporter les divers rapports et apporter les corrections nécessaires.

Spécifications fonctionnelles

27-02-2023

Date démarrage

02-04-2023

Date de fin

A DEFINIR

Coût du projet

PHASES DU PROJET

1

MISE EN PLACE DE L'ENVIRONNEMENT

Installation de l'application

2

CREATION DU BOUTON

Fonction "Mettre à jour un Lead"

3

CREATION DU COMPOSANT

Fonction "Rechercher une opportunité"

4

DEMOS ET LIVRABLES

Création des spécifications fonctionnelles

BESOINS EXPRIMÉS

Bouton de mise à jour des leads

Permettre la mise à jour le statut du lead de non contacté à contacté.

Module de recherche d'opportunités

Permettre la recherche d'une opportunité par mot clé.

GRANDES LIGNES DU PROJET

« EG manufacture est une grande manufacture de tapisserie. Elle utilise actuellement une version classique de Salesforce et souhaite migrer son application vers Lightning avant la fin de l'année. »



2

Le bouton **Action Rapide**

Le bouton action rapide

Bouton Update Lead Status

Permettre la mise à jour d'un Lead

UpdateLeadStatus.cmp

```
<aura:component implements="force:lightningQuickAction,force:hasRecordId" controller="StatusController" >
    <aura:handler name="init" value="{!this}" action=" {!c.doInit}"/>
    <aura:handler event="aura:waiting" action=" {!c.showSpinner}"/>
    <aura:handler event="aura:doneWaiting" action=" {!c.hideSpinner}"/>
    <aura:attribute name="recordId" type="String" />

    <div class="slds-spinner_container">
        <div class="slds-spinner--brand slds-spinner slds-spinner--large slds-is-relative" role="alert">
            <span class="slds-assistive-text">Loading</span>
            <div class="slds-spinner__dot-a"></div>
            <div class="slds-spinner__dot-b"></div>
        </div>
    </div>

</aura:component>
```

Le bouton action rapide

Bouton Update Lead Status

Permettre la mise à jour d'un Lead



UpdateLeadStatusController.js

```
|{  
  doInit : function(component, event, helper) {  
    var action = component.get("c.updateLead");  
    action.setParams({  
      "leadId":component.get("v.recordId") // variable du front  
    });  
    action.setCallback(this,function(response){  
      var state = response.getState();  
      if(state=='SUCCESS'){  
        //lancer un toast sucess  
        helper.showToast("Succès !","Lead mis à jour avec succès !","success");  
      } else{  
        //toast error  
        helper.showToast("Oups !","Il y a eu une erreur, veuillez réessayer !","error");  
      }  
      // $A.get("e.force:closeQuickAction").fire();  
    });  
    $A.enqueueAction(action);  
  }  
}
```

Le bouton action rapide

Bouton Update Lead Status

Permettre la mise à jour d'un Lead



UpdateLeadStatus.helper

```
({
    showToast : function(titleValue, messageValue, typeValue) {
        var toastEvent = $A.get("e.force:showToast");
        toastEvent.setParams({
            "title": titleValue,
            "message": messageValue,
            "type": typeValue,
        });
        toastEvent.fire();
    }
})
```

Le bouton action rapide

Création de l'action rapide

Permettre la mise à jour d'un Lead

The screenshot shows the Salesforce Setup interface with the 'Object Manager' selected. Under the 'Lead' object, the 'Buttons, Links, and Actions' section is highlighted. A new quick action named 'Update Status 2' is being edited. The 'Action Detail' pane shows the configuration for this action:

Label	Update Status 2
Standard Label Type	Name
	Update_Status
Description	Lightning Component
	UpdateLeadStatus
Height	250px
Created By	MAGDALENA LARMET, 09/03/2023 17:57
Modified By	MAGDALENA LARMET, 09/03/2023 17:57

On the right side of the screen, there is a sidebar with various setup options: Lead, Action Type, Lightning Component, and Icon.



3

Le composant De Recherche

Le composant Recherche

Rechercher une opportunité sur un compte
Permettre la recherche d'une opportunité

OppListController.apx

```
public with sharing class OppController {  
    @AuraEnabled  
    public static List<Opportunity> getOpps(Id recordId){  
        return [SELECT Id, name, Amount, CloseDate, StageName FROM Opportunity WHERE AccountId =:recordId LIMIT 50];  
    }  
    @AuraEnabled  
    public static List<Opportunity> getOppByName(Id recordId, String searchKey){  
        String formatedSearchKey = '%' +searchKey+ '%';  
        return [SELECT Id, name, Amount, CloseDate, StageName FROM Opportunity WHERE AccountId =:recordId and Name LIKE :formatedSearchKey LIMIT 50];  
    }  
}
```



Le composant Recherche

Rechercher une opportunité sur un compte
Permettre la recherche d'une opportunité

OppList.cmp

```
<kaura:component implements="flexipage.availableForRecordHome,force:hasRecordId,force:appHostable" controller="OppController" access="global">
    <aura:attribute name="Opportunities" type="Opportunity" />
    <aura:attribute name="recordId" type="Id"/>
    <aura:attribute name="Account" type="Account"/>
    <aura:attribute name="Columns" type="List"/>
    <aura:attribute name="searchKey" type="String" />
    <aura:handler name="init" value="{!this}" action="{!c.doInit}" />
    <aura:handler event="aura:waiting" action="{!c.showSpinner}"/>
    <force:recordData aura:id="accountRecord" recordId="{!v.recordId}" targetFields="{!v.Account}" layoutType="FULL"/>

    <lightning:card iconName="standard:opportunity" title="{'Opportunity List for ' + v.Account.Name }">
        <lightning:input class="search" type="text" name="searchKey" label="Enter" aura:id="searchKey" placeholder="Search" value="{!v.searchKey}" />
        <lightning:button label="Search Opp" onclick="{!c.search}"/>
        <lightning:dataTable data="{!v.Opportunities}" columns="{!v.Columns}" keyField="Id" hideCheckboxColumn="true"/>
    </lightning:card>

    <aura:if isTrue="{!v.toggleSpinner}">
        <div class="slds-spinner_container">
            <div class="slds-spinner--brand slds-spinner--large slds-is-relative role="alert">
                <span class="slds-assistive-text">Loading</span>
                <div class="slds-spinner__dot-a"></div>
                <div class="slds-spinner__dot-b"></div>
            </div>
            <div class="slds-spinner__dot-b"></div>
        </div>
    </aura:if>
</aura:component>
```



Le composant Recherche

Rechercher une opportunité sur un compte
Permettre la recherche d'une opportunité



OppListController.js

```
|{  
  
doInit : function(component, event, helper) {  
    component.set("v.Columns", [  
        {label: "Name", fieldName: "Name", type: "Text"},  
        {label: "Amount", fieldName: "Amount", type: "Number"},  
        {label: "Close Date", fieldName: "CloseDate", type: "Date"}  
    ]);  
    var action = component.get("c.getOpps");  
    action.setParams({  
        recordId: component.get("v.recordId")  
    });  
    action.setCallback(this, function(a) {  
        component.set("v.Opportunities", a.getReturnValue());  
    });  
    $A.enqueueAction(action);  
},  
  
search: function(component){  
    var action = component.get("c.getOppByName");  
    action.setParams({  
        recordId: component.get("v.recordId"),  
        searchKey : component.get("v.searchKey")  
    });  
    action.setCallback(this, function(a) {  
        component.set("v.Opportunities", a.getReturnValue());  
    });  
    $A.enqueueAction(action);  
},  
  
showSpinner : function(component,event,helper){  
    // display spinner when aura:waiting (server waiting)  
    component.set("v.toggleSpinner", true);  
}  
})
```

Le composant Recherche

Rechercher une opportunité sur un compte
Permettre la recherche d'une opportunité

OppList.helper

```
|({  
    getOppList: function(component) {  
        var action = component.get("c.getOpps");  
        action.setCallback(this, function(a) {  
            component.set("v.opportunities", a.getReturnValue());  
        });  
        //$.A.enqueueAction(action);  
    }  
})
```



Le composant Recherche

Modification de la page Compte
Permettre la recherche d'une opportunité

The screenshot shows the Salesforce Lightning App Builder interface for modifying an Account Record Page. The top navigation bar includes 'Lightning App Builder', 'Pages', and 'Account Record Page'. The left sidebar lists various components under 'Components' and 'Fields', such as Salesforce Anywhere Messages, Tabs, Topics, Trending Topics, Twitter, Visualforce, and several custom components. The main content area displays an account record for 'Burlington Textiles Corp of America' and an opportunity list for the same account. The opportunity list includes columns for Name, Amount, and Close Date, showing one entry for 'Burlington Textiles Weaving Plant Generator' with an amount of 235000 and a close date of 2020-08-21. The right sidebar contains configuration fields for the page, including 'Label' (Account Record Page), 'Developer Name' (Account_Record_Page), 'Page Type' (Record Page), 'Object' (Account), 'Template' (Three Regions), and 'Description'. A 'Save' button is visible at the bottom right.

Les captures d'écran

4

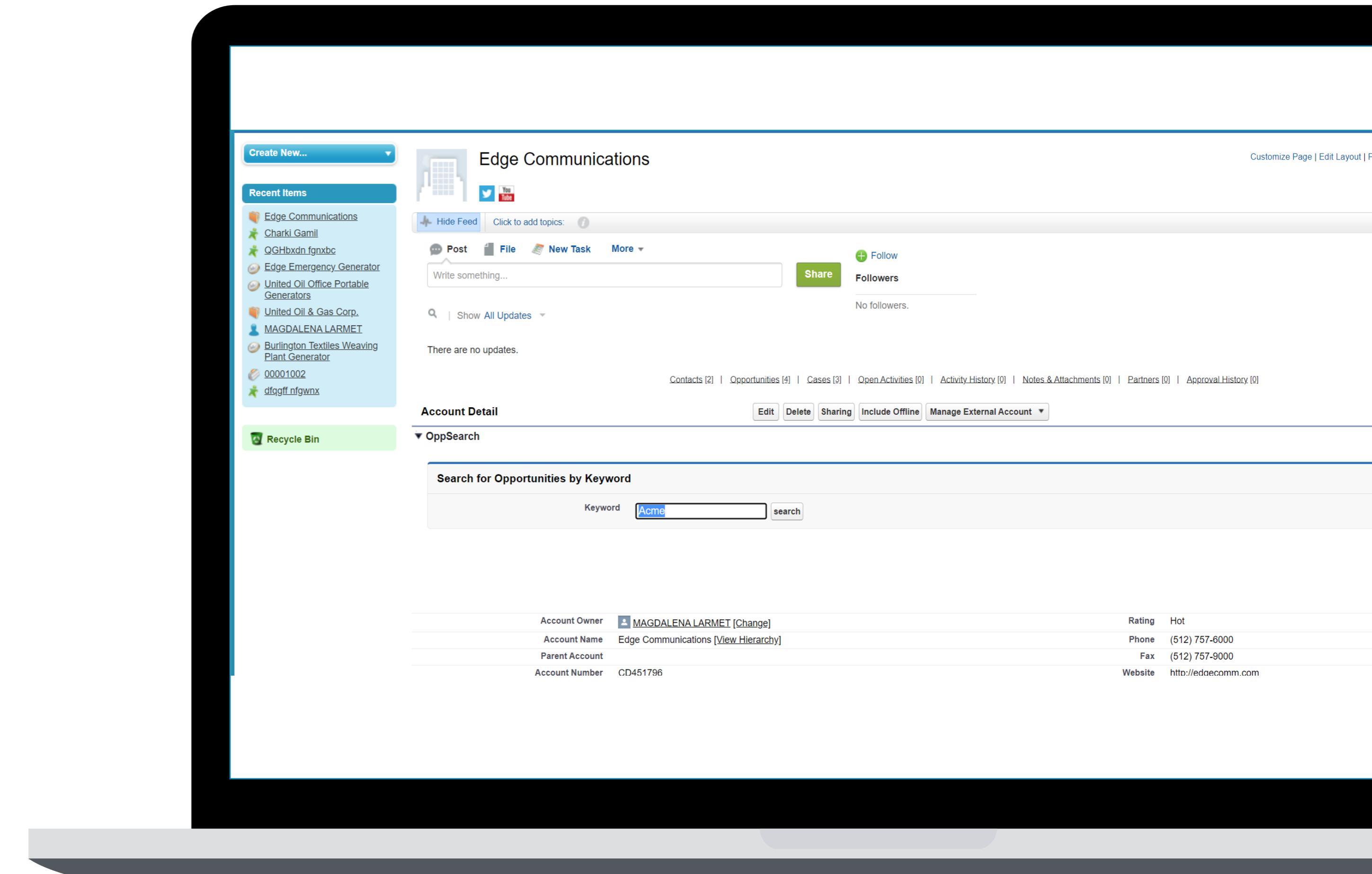


Composant Recherche Opportunité

Rechercher une opportunité

Version actuelle

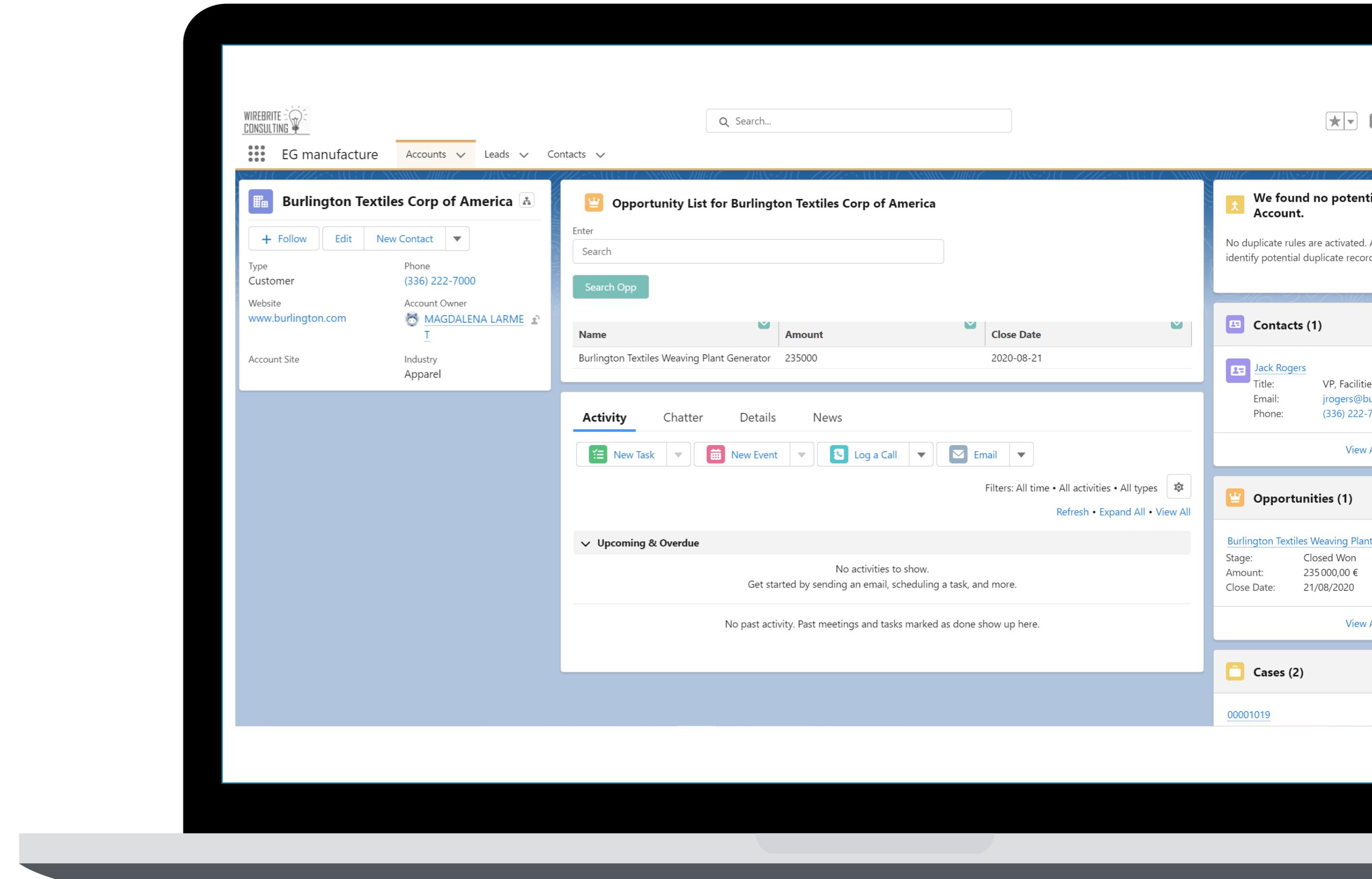
Un design **ancien**, pas très
engageant avec une **interface**
pas aux normes **W3C** en terme
d'ergonomie.



Composant Recherche Opportunité

Rechercher une opportunité
Version Lightning

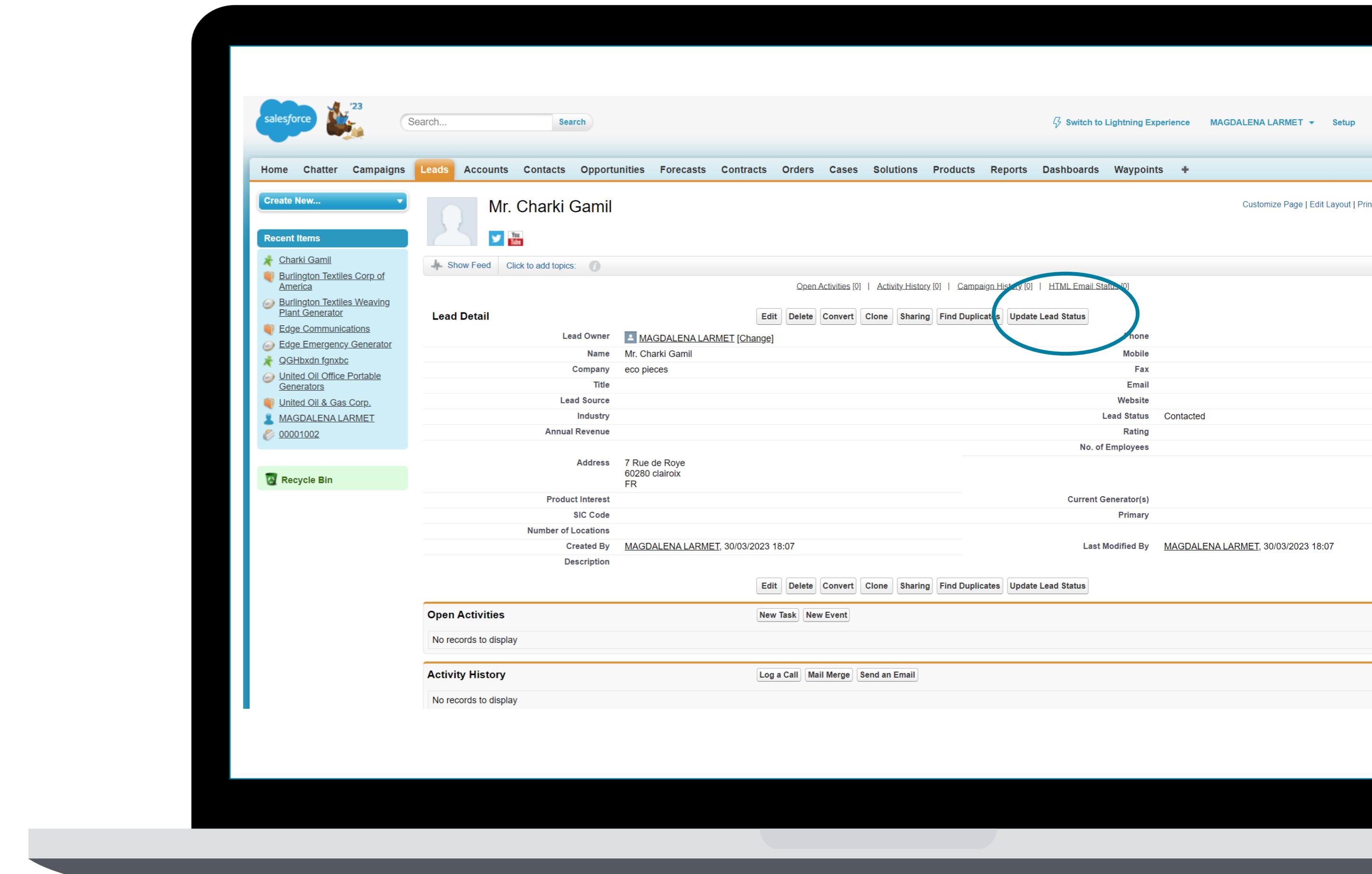
Un design **moderne, organisé**
sous forme de **blocs**, pas très
engageant avec les
informations importantes
mises en **disponibles** et le
composant de recherche
d'opportunité facilement
accessible.



Bouton Mise à jour Lead

Mettre à jour un Lead
Version actuelle

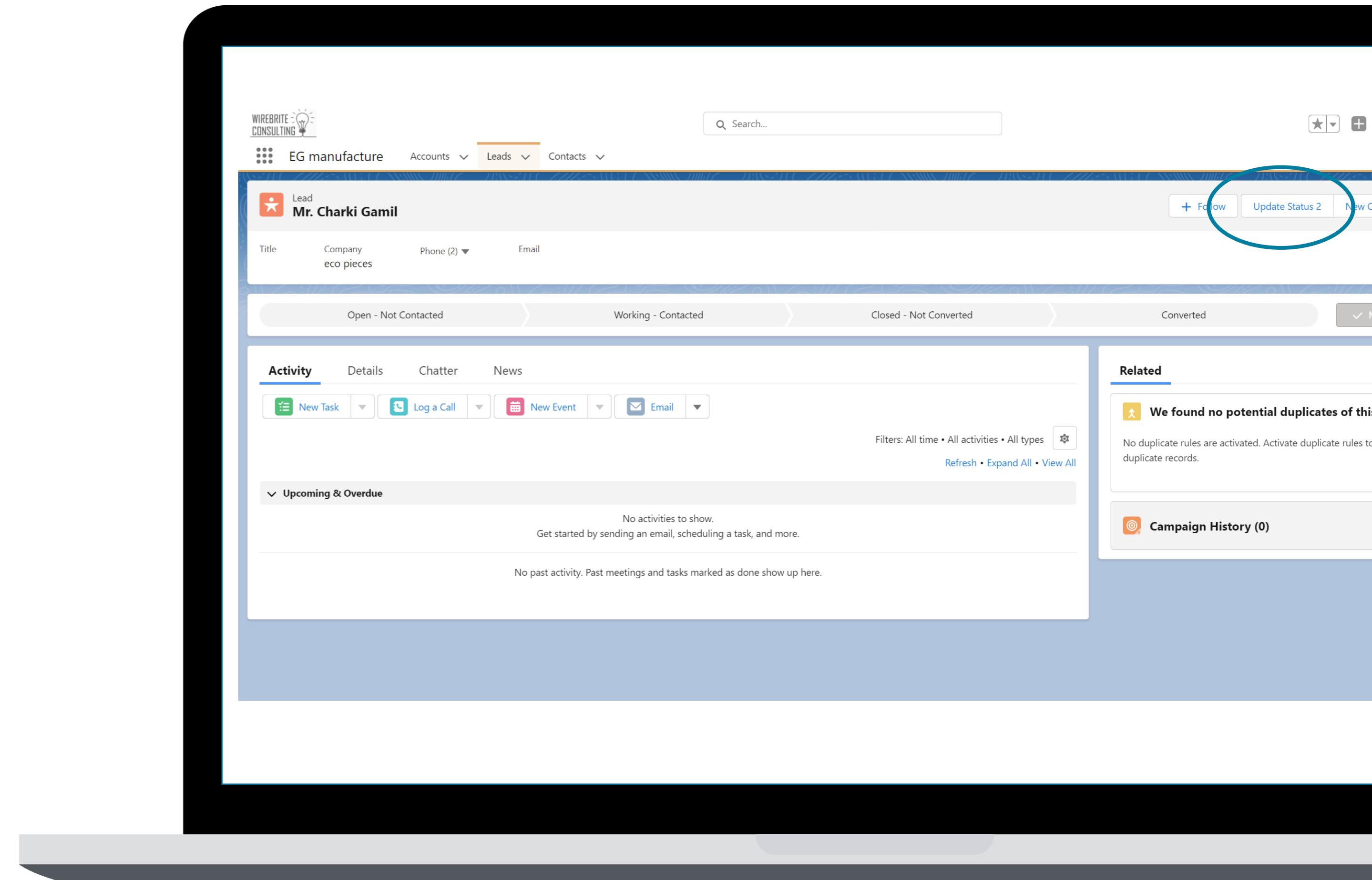
Un design **ancien**, assez **terne** avec un **bouton** d'action rapide accessible en haut de page. Une **interface** organisée sous forme de **liste** avec les **composants** les uns sous les **autres**.



Bouton Mise à jour Lead

Mettre à jour un Lead
Version Lightning

Un design **moderne**, et assez
engageant avec un **composant**
pas aux normes **W3C** en terme
d'ergonomie et de code. Une
interface minimaliste pour
supprimer les **distractions**
visuelles.

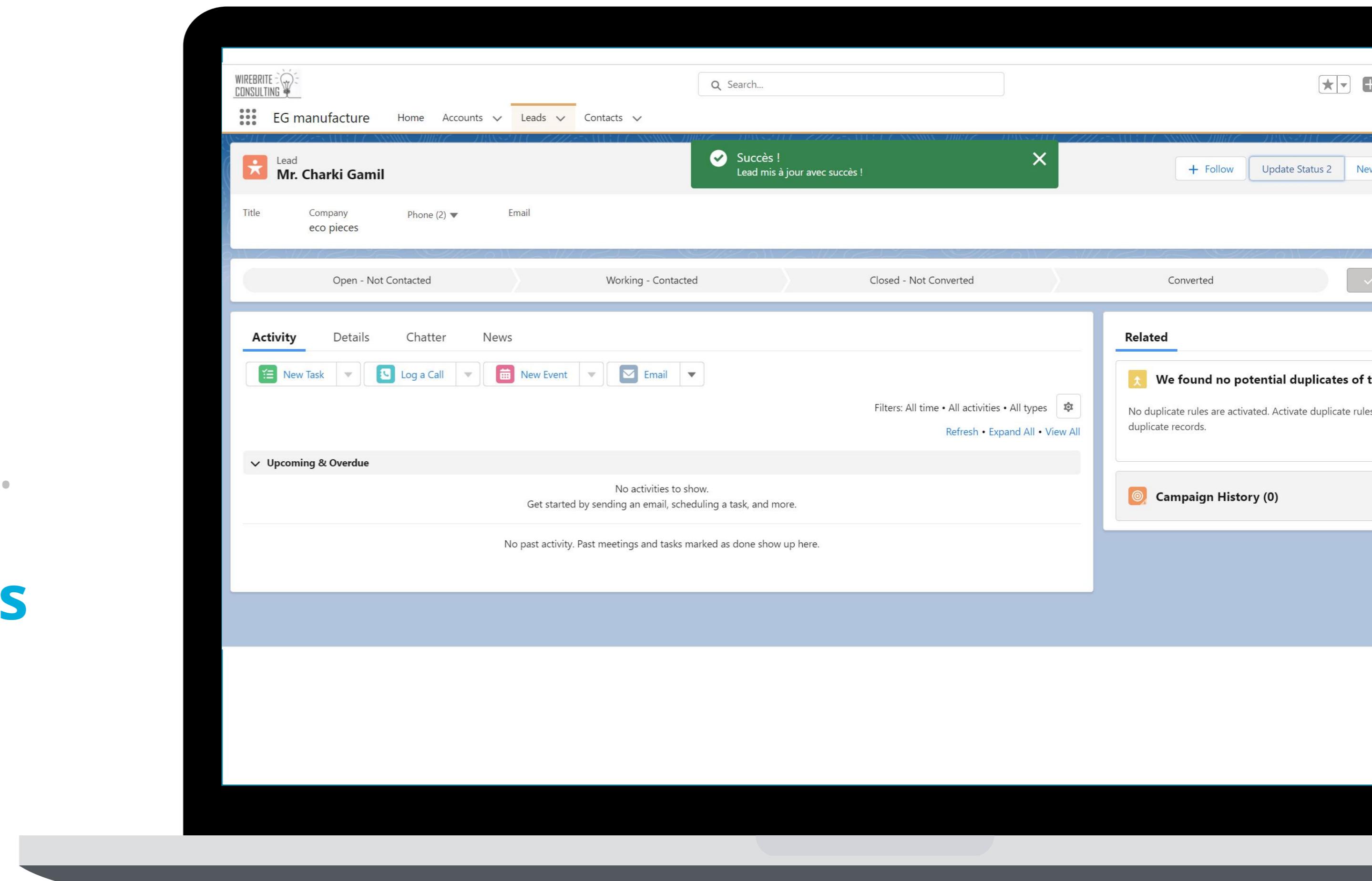


Page Dynamique

Actions rapide et Toast messages

Version Lightning

Un bouton d'**action rapide** sur chaque opportunité pour **mettre à jour** le **statut** du lead. **de la société.** Un toast **confirme** la mise à jour **une fois** l'action terminée.

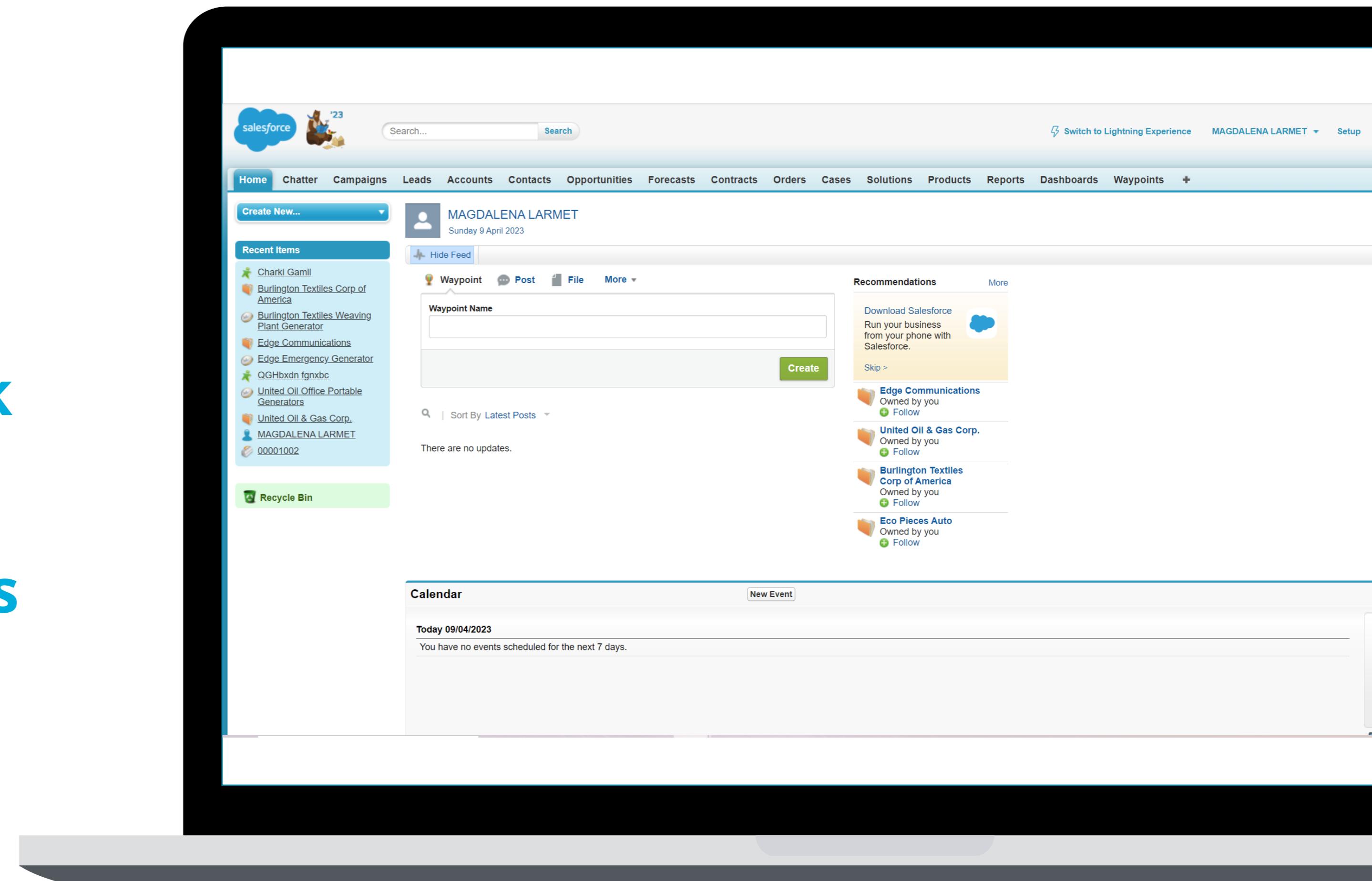


Page Accueil

La page d'accueil
Version actuelle

Une page d'**accueil** qui liste les dernières actions et présente un **calendrier**. Design très **back office** et pas très **engageant**.

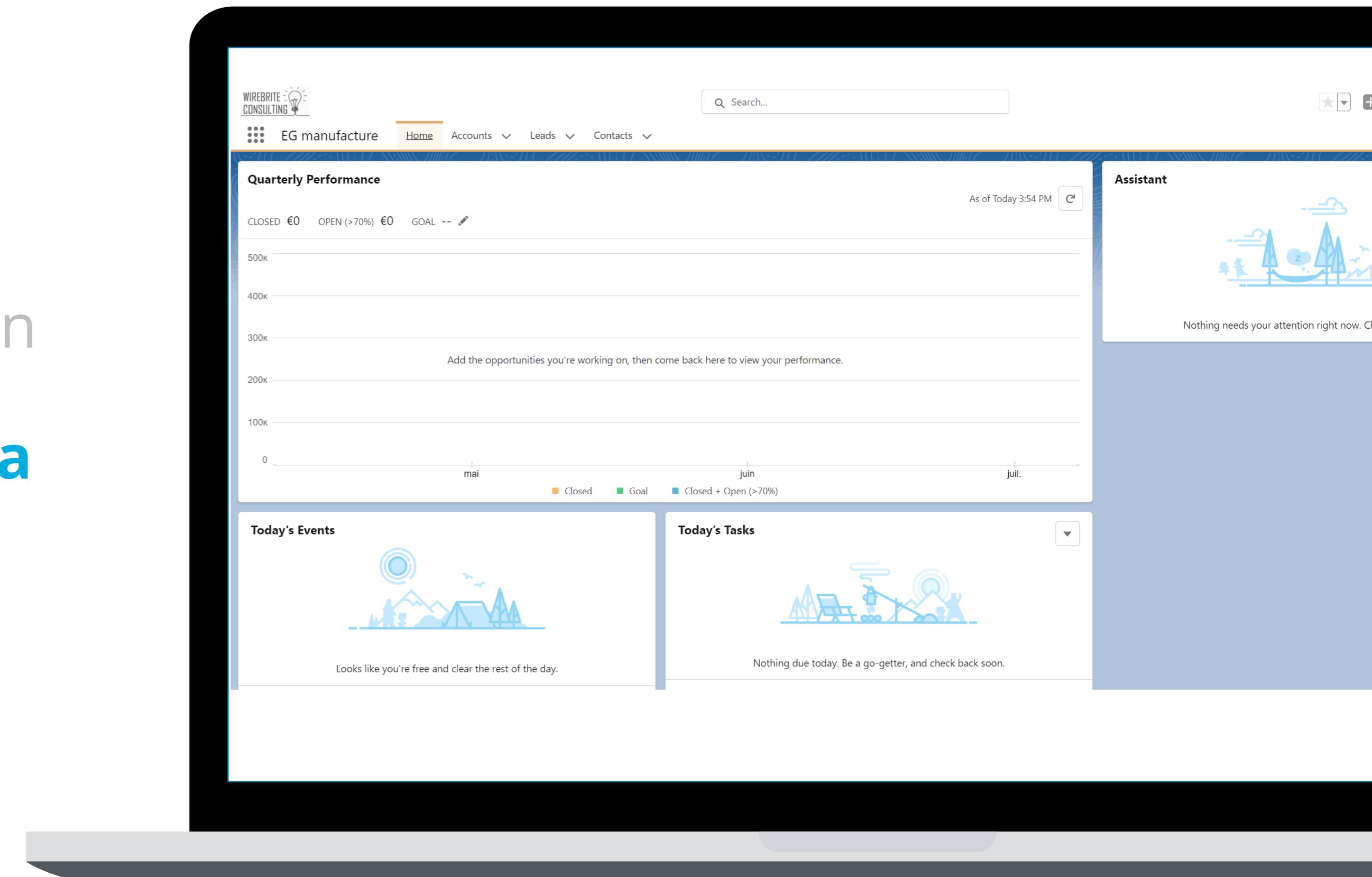
Design qui ne tient pas compte des **évolutions technologiques** en terme d'**ergonomie** et de **normes**.



Page Accueil

La page d'accueil
Version Lightning

Une page d'**accueil personnalisable** sur lesquels on peut afficher les **métriques** et **informations** importantes **de la société**. Un Design en **adéquation** avec les avancées **ergonomiques actuelles** et répondant aux **normes W3C**.





ms

Magdalena Larmet
Salesforce
Developer