

Contrôle des valeurs saisies

Description du thème

Propriétés	Description
Intitulé long	Observation puis modification d'une procédure de l'application du contexte Gestion des formations.
Formation concernée	Gestion des Systèmes d'Information en classe de terminale GSI
Matière	Gestion des Systèmes d'Information
Présentation	Cet exercice s'appuie sur le contexte de niveau II Gestion des formations à la SNCF. À partir de l'observation d'une application, détecter les insuffisances et modifier le code pour éviter les erreurs de saisie
Notions	C. 2. L'adaptation de l'application 2.1. Prise en charge du dialogue homme-machine 2.3. Programmation des traitements
Transversalité	
Pré-requis	Instructions conditionnelles, répétitives, événement
Outils	Logiciel de développement d'application Access
Mots-clés	IHM VBA Test application utilisateur formulaire composant jeu d'enregistrements
Durée	2 heures
Auteur(es)	Christian Draux, Éric Deschaintre, Jean Philippe Pujol
Version	v 1.0
Date de publication	janvier 2007

L'objectif est le suivant : Suite à des erreurs signalées par plusieurs agents stagiaires, la responsable des formations, M^{me} Fiers observe qu'actuellement la saisie d'une nouvelle action de formation n'est pas toujours fiable, il souhaite que l'application soit mise à jour pour qu'elle vérifie la vraisemblance des informations saisies par les utilisateurs.

À partir de l'observation des programmes de l'application formation.mdb, on va :

- Observer les insuffisances dans le contrôle de validité des informations saisies
- Proposer des solutions pour garantir la concordance des informations saisies
- Compléter ou créer une procédure de façon à prendre en charge ces nouveaux besoins.

Il est souhaitable de réaliser cet exonet après la réalisation :

- du côté cours cours « Approche de la programmation du contexte 2 »
<http://www.reseaucerta.org/cotecours/cotecours.php?num=401>
- de l'exonet « Gestion des frais de déplacement »
<http://www.reseaucerta.org/cotecours/cotecours.php?num=402>

Observation de l'application

Le cas d'utilisation sur lequel s'appuie l'application est le suivant :

Cas d'utilisation	Créer une session
Acteur	Responsable des formations
Événement déclencheur	Début année ou besoin ponctuel
Parties prenantes et intérêts	Mettre à jour le planning des sessions afin de permettre l'inscription des agents
Portée	Planning des sessions
Pré-conditions	Composantes de la session valorisées et vérifiées ; responsable identifié
Scénario nominal	<ol style="list-style-type: none">1. Le système présente un formulaire2. Le responsable choisit l'activité concernée3. le système propose les actions correspondantes4. Le responsable choisit l'action concernée5. le système propose ses caractéristiques et les sessions concernées6. Le responsable renseigne le formulaire en saisissant la date, le nombre maximal de participants, le coût prévu et en choisissant le lieu. Puis il le soumet au système7. le système enregistre les données
Cas particuliers	7a. Certains champs sont mal renseignés : aller en 6.

L'interface correspondante se présente ainsi :

Numero	Date	Nb part.	Lieu	Coût héberg. prévu
156	26/10/2006	2	Saintes	112
171	12/07/2006	11	Angoulême	1200
*				

A la date du 12/10/2006 on cherche à ajouter une session de formation avec les valeurs suivantes :

Champ	Valeur
Activité :	Gestion
Action de l'activité :	Contrôleur de gestion
Date :	12/07/2006
Nombre de participants :	-2
Ville :	Angoulême
Coût d'hébergement prévu :	1200

1. Retrouver les valeurs aberrantes que l'utilisateur cherche à saisir.
2. Saisir les valeurs dans l'application.
3. Quels sont les contrôles effectués par l'application au moment de la saisie sur chacun des champs ?
4. Quels sont les champs qui acceptent cependant des valeurs erronées ?
5. Donner des exemples de valeurs incorrectes que l'on peut saisir.

Observons la procédure associée à l'événement « Avant mise à jour » (*before update* en anglais) du contrôle 'nombreMaxParticipants' :

```
1. Private Sub nombreMaxParticipant_BeforeUpdate(Cancel As Integer)
2. Dim nbInscrits As Integer
3. If Not IsNull(Me.numero) Then
4.     Dim rsInscrits As DAO.Recordset
5.     Set rsInscrits = CurrentDb.OpenRecordset("select count(*) as nb
6.     from inscription where numeroSession=" & Me.numero)
7.     nbInscrits = rsInscrits("nb")
8.     rsInscrits.Close
9.     If Me.nombreMaxParticipant < nbInscrits Then
10.        MsgBox "Impossible, le nombre d'inscrits est " & nbInscrits
11.        Cancel = True
12.    End If
13. End Sub
```

6. Quel est le contrôle de validité déjà pris en charge par le code associé à l'événement « Avant mise à jour » ?
7. Quels sont les autres contrôles que doit prendre en charge l'application sur chacun des champs du formulaire ?

Évolution de l'application pour prendre en charge de nouveaux besoins

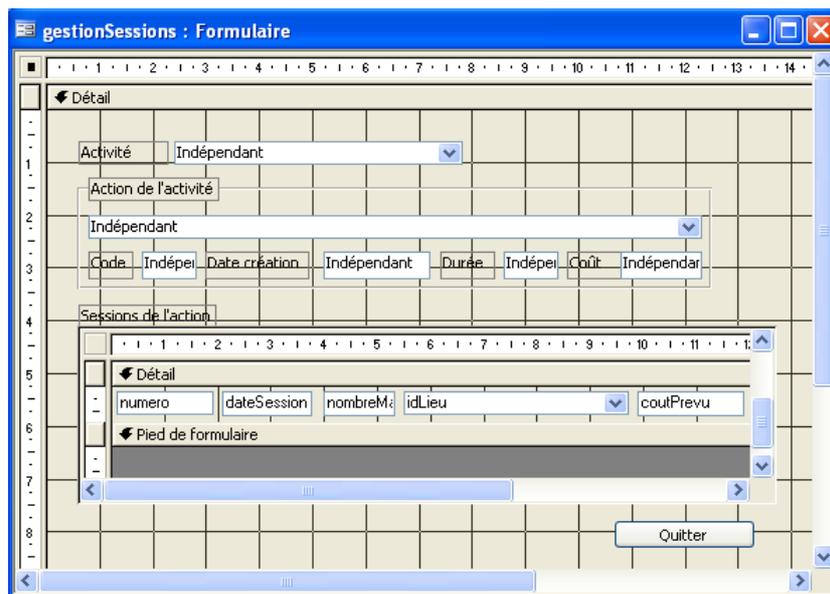
Le diagnostic sur les besoins d'évolution de l'application étant réalisé, il faut maintenant modifier le code de l'application pour les prendre en compte.

Validation de la date d'une session.

On décide de réaliser deux contrôles sur la date de session :

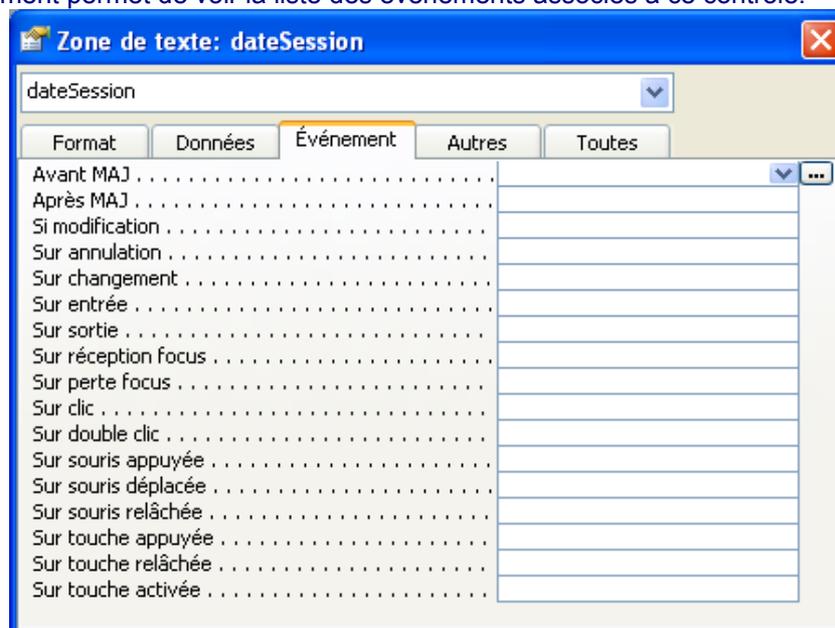
- la date saisie doit être supérieure à la date du jour.
- la date saisie doit être comprise dans l'année civile en cours.

Basculer le formulaire en mode modification (bouton ) vous devez obtenir l'image suivante :



Sélectionner le contrôle `dateSession` et cliquer sur l'icône des propriétés du contrôle () ou clic-droit Propriétés.

L'onglet Événement permet de voir la liste des événements associés à ce contrôle.



1. *Rechercher dans l'aide du logiciel la définition des deux premiers événements (touche F1 lorsque le curseur est positionné sur la ligne d'un événement).*
2. *Parmi ces 2 premiers événements quel est celui qui semble le plus approprié pour déclencher le code qui permettra de tester la validité d'une date ?*

Date saisie inférieure à la date du jour

3. *Rechercher dans l'aide s'il existe une fonction qui fournit la date du jour (on parle aussi de « date système »)*

Pour accéder à l'interface d'édition du code associé à un événement, cliquer sur les points ( ) situés au bout de la ligne de l'événement.

L'interface montre immédiatement la procédure qui est exécutée lorsque cet événement se produit.

```
Private Sub dateSession_NomEvenement(Cancel As Integer)
End Sub
```

NomEvenement prend la valeur du nom de l'événement choisi. Exemple Click() pour l'événement « sur Clic ».

4. *Rédiger le message à afficher dans la boîte de dialogue pour avertir l'utilisateur de l'erreur.*
5. *En vous inspirant du code utilisé dans la procédure nombreMaxParticipant_BeforeUpdate(Cancel As Integer), rédiger le code qui permettra d'afficher le message d'erreur dans une boîte de dialogue.*



Validation du nombre de participants.

L'application empêche actuellement de saisir un nombre de participants inférieur au nombre d'inscrits à une session (voir plus haut la procédure nombreMaxParticipant_BeforeUpdate).

6. *Dans quel cas ce contrôle est il utile ?*

Validation du nombre minimal de participants

On souhaite maintenant empêcher la saisie d'une valeur 0 à la création d'une session.

7. *Modifier le code de l'application (Private Sub nombreMaxParticipant_BeforeUpdate(Cancel As Integer)) afin qu'elle prenne en charge cette nouvelle fonctionnalité.*

Validation du nombre maximal de participants

Si le nombre minimal est désormais contrôlé, ce n'est pas le cas du nombre maximal.

8. *Combien de participants une session peut-elle accueillir ?*

Pour répondre, vous rédigerez la requête d'interrogation de la table session qui détermine le plus grand nombre de participants d'une session puis vous l'exécuterez.

9. *Modifier le code de l'application pour tenir compte de cette nouvelle contrainte.*

Il pourrait arriver qu'exceptionnellement une session dépasse ce maximum. On veut donc laisser à l'utilisateur le choix de la décision à prendre.

10. *Que faut-il modifier pour **ne pas empêcher** la saisie d'une valeur maximale erronée ?*

Pour faire prendre conscience à l'utilisateur que son choix est important, il serait bon que la boîte de dialogue affiche les deux choix possibles et les boutons correspondants.

11. Rédiger le message à afficher.

12. Rechercher dans l'aide du logiciel les paramètres à employer pour afficher plusieurs boutons et récupérer la valeur du bouton choisit par l'utilisateur.

Le résultat à obtenir est le suivant :



Vérification d'un dépassement budgétaire.

La création d'une nouvelle session ne doit pas entraîner un dépassement du budget prévu pour une action de formation.

Exemple : la formation « Formation soudeur inox niveau 1 » (E0335) comporte déjà les sessions suivantes :

numero	coutReel
45	380,82
71	477,33
148	810
Total	1 668,15

Le budget alloué à cette action de formation pour toutes ses sessions est de : 2 110.

Il faut donc alerter l'utilisateur si le coût prévu pour cette nouvelle session est supérieur à 2110 - 1668,15.

13. Modifier l'application pour prendre en compte cette nouvelle contrainte.