



Application Android Java SQLite

13.04.2024

Réalisé par:

Medjber Soundous

Ghalem Salim

Chouiref Sidali

Introduction:

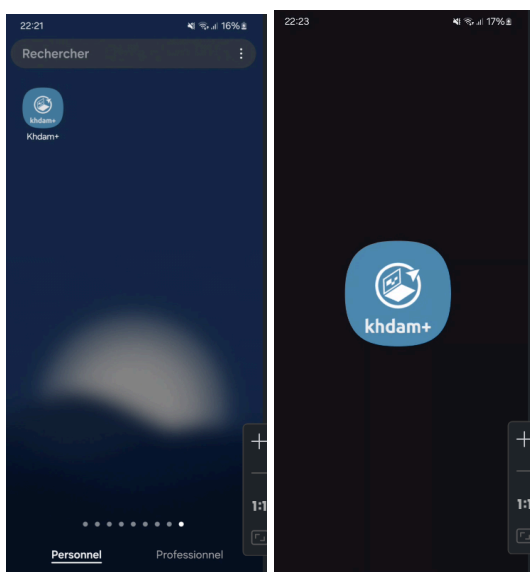
Notre solution pour le projet, une application mobile qui simplifie la gestion des employés en fournissant toutes les informations nécessaires sur chaque employé. Notre application permettra de gérer les détails essentiels des employés tels que leur nom, prénom, email, titre du poste, numéro de téléphone, adresse de résidence, et même leur photo, ainsi que la possibilité de changer le thème et la langue selon les préférences de l'utilisateur.

Notre application offre toutes les fonctionnalités nécessaires telles que l'ajout d'un nouvel employé, la modification des informations d'un employé, la suppression d'un employé, ainsi que la recherche rapide pour trouver un employé spécifique en le recherchant par nom et prénom. De plus, la possibilité de personnaliser l'affichage de la liste des employés selon vos préférences.

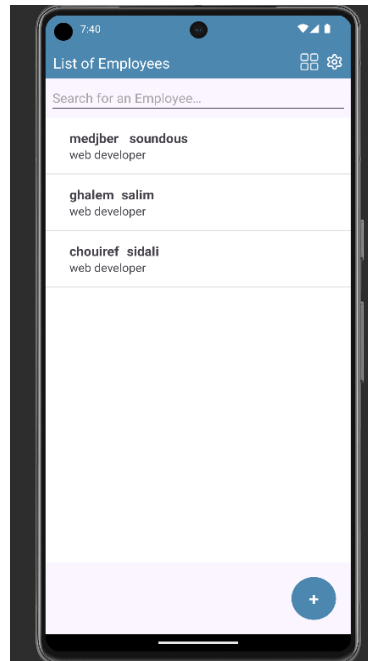
Notre application vous permettra de contacter facilement les employés par email, SMS ou appel directement.

Partie UI/UX:

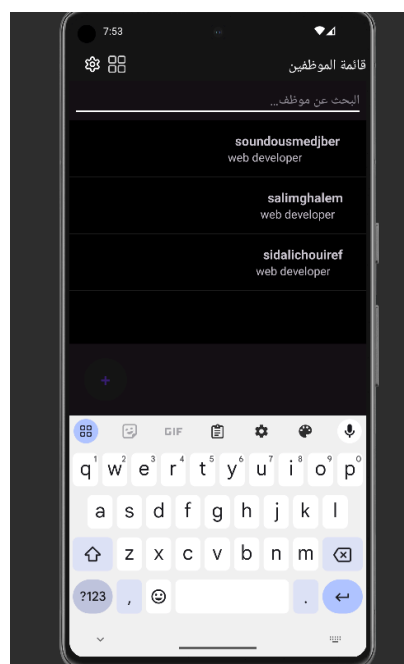
Icon et Naming de notre application:



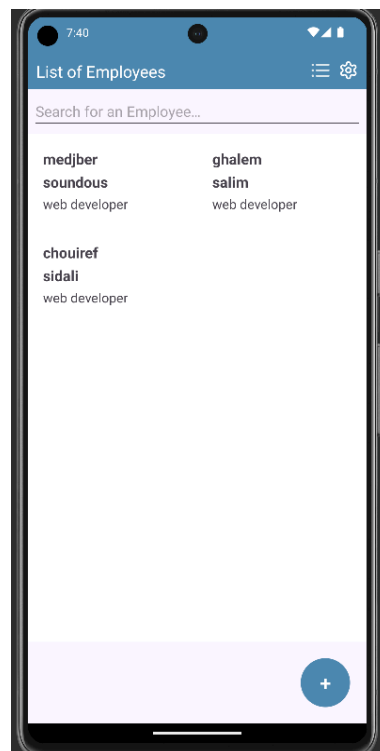
Activité Principale de notre application:



un exemple de page affichant du contenu en arabe:

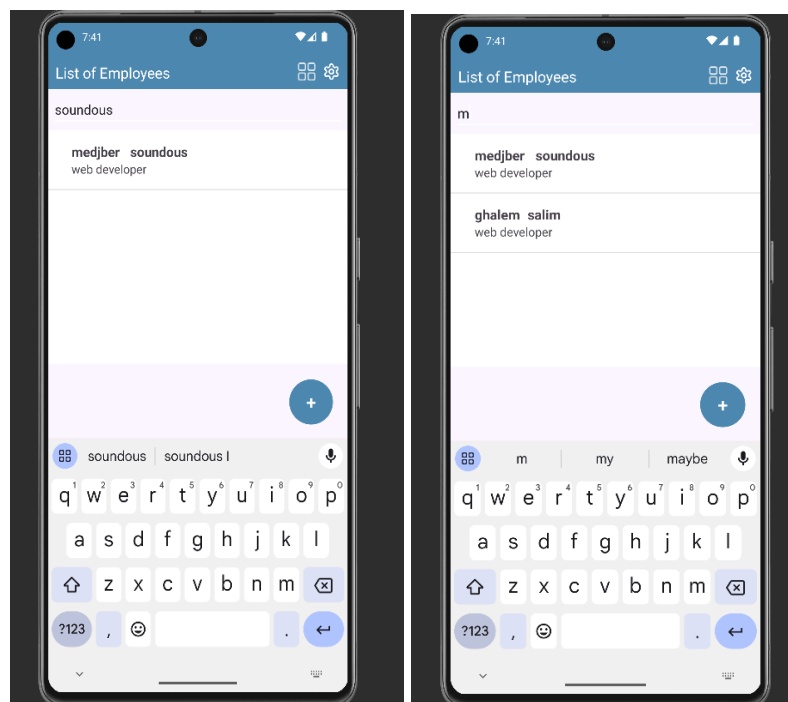


L'affichage de notre application lorsque l'on clique sur l'icône de grid pour mettre le contenu sous forme de grid:



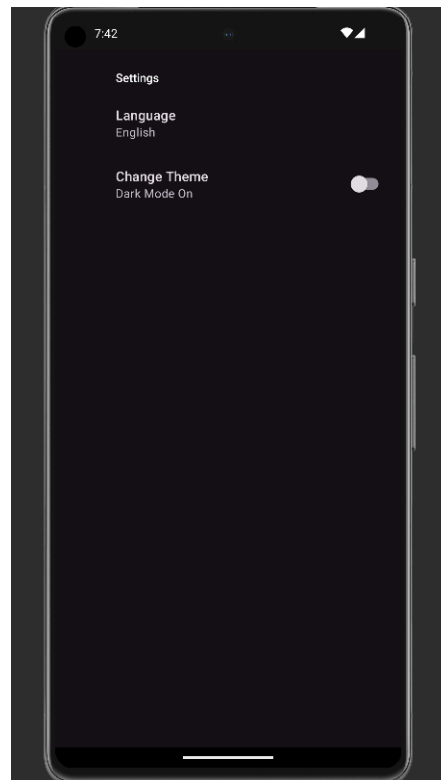
-l'icône de liste qui nous ramènera à l'affichage sous forme de liste

Edittext pour chercher un employé par nom ou par prenom :

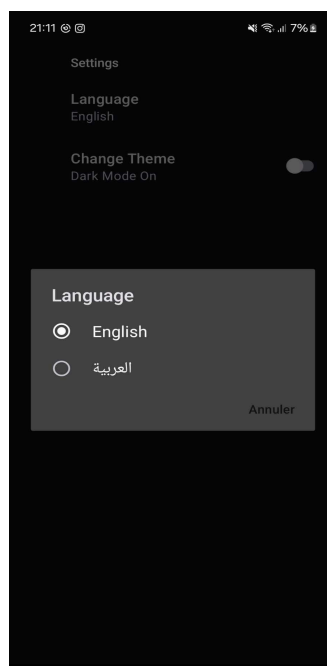


Activité Paramètres de notre application:

Lorsque l'on clique sur l'icône des paramètres, la page des paramètres apparaît pour changer la langue ainsi que le thème.

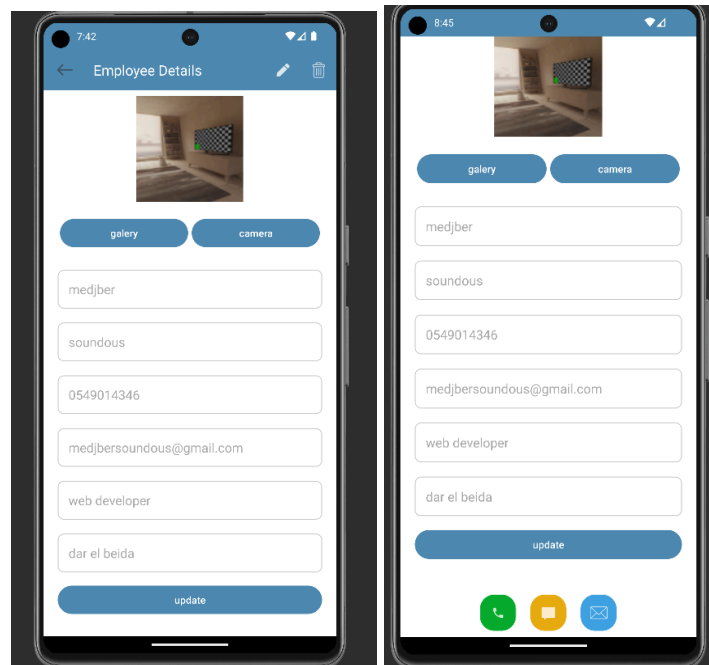


pour switcher entre anglais et arabe:



Activité Détails de notre application:

Lorsque l'on clique sur un employé, cela nous dirigera vers cette activité(avec ScrollView utilisé pour afficher tout)



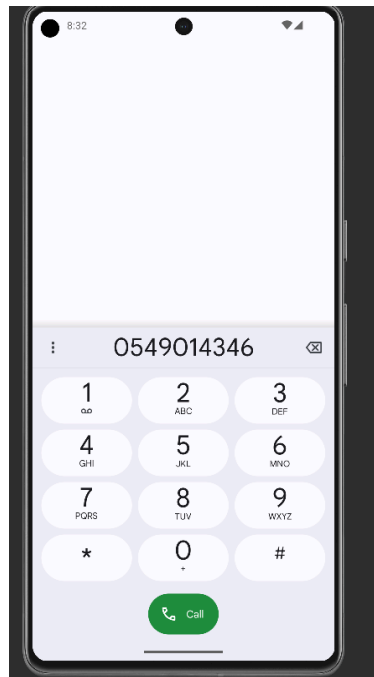
-les deux icônes pour supprimer et modifier les informations d'un employé.

-et le bouton update pour confirmer les modifications.

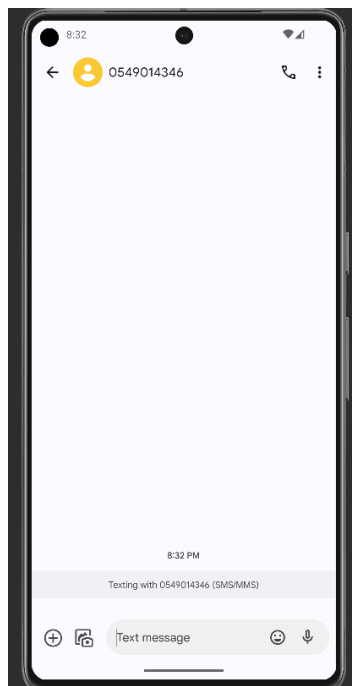
les 3 boutons pour contacter l'employé:

Chaque bouton nous dirige vers l'application de contact.

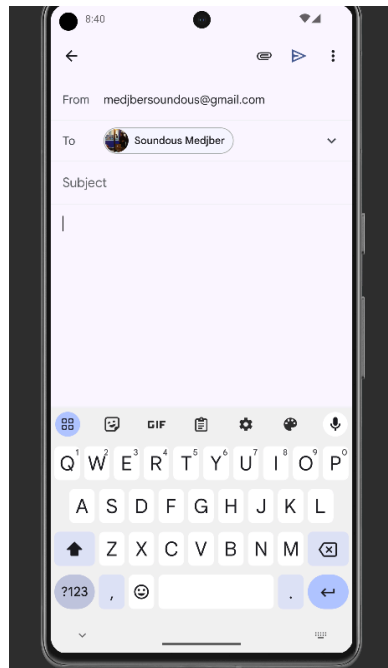
par appel téléphonique :



par sms:

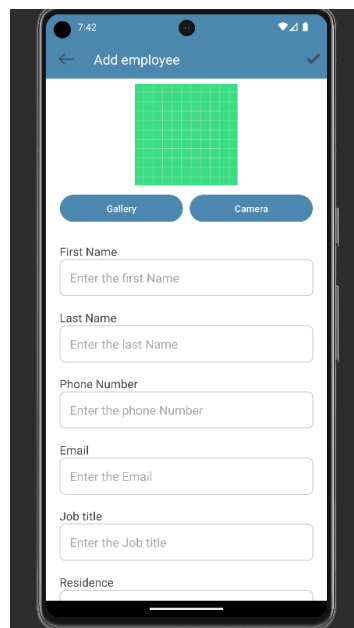


par mail:



Activité Ajout employé de notre application:

Accessible grâce au bouton + de l'activité principale



-Lorsque l'on clique sur l'icône de validation, le nouvel employé sera ajouté à la base de données et affiché dans la liste

Explication du code (Partie Java):

Fonctions utilisés dans L'activité settings:

- **SettingsActivity** : Cette activité gère la page des paramètres où l'utilisateur peut modifier la langue et le thème de l'application.

updateLocale: Cette fonction met à jour la langue de l'application en fonction de la préférence sélectionnée par l'utilisateur. Elle prend en charge la mise à jour de la configuration de la langue et la réinitialisation de l'activité pour appliquer les modifications.

onCreatePreferences: c'est pour charger les préférences à partir du fichier XML root préférences et d'initialiser les éléments de l'interface utilisateur en fonction de ces préférences.

Fonctions utilisés dans L'activité Principale et Détails:

- **MainActivity** : main activité contient les fonctions pour afficher la liste des employés et gérer les interactions principales de l'application.

displayEmployees: Cette fonction récupère les données des employés à partir de la base de données et les affiche dans la vue appropriée, soit sous forme de liste ListView ou bien GridLayout .

filterEmployeeList: Cette fonction permet de filtrer la liste des employés en fonction du texte saisi par l'utilisateur dans le "edittext" soit par nom ou bien par prénom.

updateGridLayout: Cette fonction met à jour l'affichage des employés dans GridLayout quand on ajoute un nouvel employé.

showEmployeeDetails: Cette fonction ouvre l'activité EmployeeDetails pour afficher les détails d'un employé spécifique lorsque l'utilisateur clique sur un employé.

toggleListViewGridLayout: pour changer l'affichage soit grid soit list.

GoToSettings: pour ouvrir l'activité Settings lorsque l'utilisateur souhaite accéder aux paramètres de l'application pour changer le thème ou la langue.

[onActivityResult](#): Cette fonction est appelée lorsque l'activité AddEmployeeActivity se termine avec un résultat, lorsqu'un nouvel employé est ajouté, elle actualise l'affichage de la liste des employés.

[applyLanguage](#): Cette fonction permet d'appliquer la langue sélectionnée par l'utilisateur, elle récupère la langue sélectionnée dans les paramètres de l'application et met à jour la configuration de la langue en conséquence.

[onConfigurationChanged\(Configuration newConfig\)](#) : Cette méthode est appelée lorsque la configuration de l'appareil change, lorsque la langue ou bien le thème est modifiée

- **EmployeeListAdapter** : Cet adaptateur gère l'affichage des employés sous forme de liste.
- **EmployeeGridAdapter** : Cet adaptateur gère l'affichage des employés sous forme de grid.
- **EmployeeDetailsActivity** : est une activité qui affiche les détails d'un employé sélectionné et permet à l'utilisateur d'effectuer des actions telles que la modification des informations de l'employé, la suppression de l'employé, l'envoi d'un email, l'appel ou l'envoi de SMS.

[toggleEditMode](#): Cette fonction permet de basculer entre le mode édition et le mode lecture seule pour les Edittext affichant les détails de l'employé.

[setEditTextReadonly](#): Cette fonction rend les Edittext en mode lecture seule en désactivant la possibilité de modifier leur contenu.

[setEditTextEditable](#): Cette fonction rend les Edittext éditables en activant la possibilité de modifier leur contenu.

[convertImageToByteArray](#): Cette fonction convertit l'image de profil de l'employé en tableau d'octets afin de le stocker dans la base de données.

[openCamera2](#) , [openGallery2](#): Ces fonctions permettent à l'utilisateur de prendre une nouvelle photo avec l'appareil photo ou de sélectionner une photo depuis la galerie.

[onActivityResult](#): Cette fonction est appelée lorsque l'utilisateur prend une photo ou sélectionne une image de la galerie. Elle récupère l'image sélectionnée et l'affiche dans l'élément ImageView correspondant.

[SendMail](#), [MakeCall](#), [MakeSMS](#): Ces fonctions permettent à l'utilisateur d'envoyer un e-mail, de passer un appel ou d'envoyer un SMS en utilisant les informations de contact fournies.

[deleteEmployee](#): Cette fonction permet à l'utilisateur de supprimer l'employé actuellement affiché.

[updateEmployee](#): Cette fonction permet à l'utilisateur de mettre à jour les informations de l'employé actuellement affiché dans la base de données après avoir apporté les modifications nécessaires.

- **Employee** : cette class encapsule les informations d'un employé et fournit des méthodes pour accéder à ces informations et pour représenter et manipuler les données des employés.
- **DatabaseHelper** : Cette classe contient les fonctions pour interagir avec la base de données SQLite, elle permet d'exécuter des opérations telles que l'ajout, la modification et la suppression d'employés dans la base de données.

[onUpgrade\(SQLiteDatabase db, int oldVersion, int newVersion\)](#) : Cette méthode est appelée lorsqu'une mise à jour de la base de données est nécessaire. Elle supprime la table existante si elle existe déjà et appelle onCreate() pour recréer la table avec la nouvelle structure.

[insertEmployee\(\)](#) : Cette méthode insère un nouvel employé dans la base de données avec les informations fournies.

[getAllEmployees\(\)](#) : Cette méthode récupère tous les employés de la base de données.

[updateEmployee\(\)](#) : Cette méthode met à jour les informations d'un employé existant dans la base de données.

[deleteEmployee\(\)](#) : Cette méthode supprime un employé de la base de données.

[getEmployeeById\(long employeeId\)](#) : Cette méthode récupère les informations d'un employé spécifique en fonction de son identifiant.

[getAllEmployeesFiltered\(String query\)](#) : Cette méthode récupère les employés de la base de données dont le nom ou le prénom correspond à la requête spécifiée.

[getEmployeeProfileImage\(long employeeId\)](#) : Cette méthode récupère l'image de profil d'un employé spécifique.

Fonctions utilisés dans L'activité Add Employee:

- **AddEmployeeActivity** : Cette activité gère l'ajout d'un nouvel employé à la base de données. Elle contient les fonctions nécessaires pour collecter les informations de l'utilisateur et les ajouter à la base de données.

[onCreate\(Bundle savedInstanceState\)](#) : Cette méthode est appelée lors de la création de l'activité. Elle initialise les éléments de l'interface utilisateur et configure les écouteurs d'événements pour les boutons de validation et de retour.

[convertImageToByteArray\(\)](#) : Cette méthode convertit une image sélectionnée à partir de la galerie ou prise avec l'appareil photo en un tableau de bytes (byte array) pour le stockage dans la base de données.

[openCamera\(View view\).openGallery\(View view\)](#) : Ces méthodes sont appelées respectivement lorsque l'utilisateur souhaite prendre une photo avec l'appareil photo ou sélectionner une image depuis la galerie. Elles lancent des intents appropriés pour ces actions.

[onActivityResult\(int requestCode, int resultCode, Intent data\)](#) : Cette méthode est appelée lorsque l'activité de la caméra ou de la galerie se termine. Elle récupère l'image sélectionnée et l'affiche dans l'interface utilisateur.

[addEmployee\(\)](#) : Cette méthode récupère les informations saisies par l'utilisateur (prénom, nom, email etc...), convertit l'image en un tableau de bytes, puis insère un nouvel employé dans la base de données en utilisant le DatabaseHelper.

[applyLanguage\(\)](#) : Cette méthode applique la langue sélectionnée par l'utilisateur à l'activité en cours en fonction de ses préférences, elle récupère la langue sélectionnée à partir des préférences partagées et met à jour la configuration de la ressource en conséquence.

[onConfigurationChanged\(Configuration newConfig\)](#) : Cette méthode est appelée lorsque la configuration de l'appareil change, lorsque la langue ou bien le thème est modifiée