

2021



# PERSONALDIARY

INFO0306

## Sommaire

Sommaire .....	1
Introduction.....	2
Modélisation .....	3
Réalisation.....	5
Liste des fonctionnalités de l'application .....	9
Conclusion .....	10
Webographie.....	10

## Introduction

En m'appuyant sur le sujet, j'ai choisi de développer une application dont le sujet est la gestion de journal intime. Cette application nommée « PersonalDiary » permet donc de gérer des notes de rappel d'une photo de l'utilisateur, ces dernières sont affichées dans une liste de rappel.

Mon application est donc présentée sous la forme suivante, au lancement de l'application nous arrivons sur une page d'accueil ou est indiqué le nom, note a rentré ainsi que la photo à choisir qu'on enregistre par la suite dans une base de données sql lite.

L'activité principale de mon application, présente est de pouvoir ajouter des notes et des rappels a une photo pour garder des souvenir par exemple ou raconter l'histoire de la photo, nous retrouvons la liste de nos rappels dans un second fragment de notre activité. En cliquant sur le bouton « add image » on peut choisir une image de notre répertoire ou prendre une photo.

Dans un premier cas, si nous cliquons sur le bouton « save ». On nous demande d'enregistrer le mot de passe pour pouvoir ajouter le rappel dans la liste des rappels si on souhaite ajouter une image on clique sur add image.

Dans la seconde activité nous retrouvons la liste des rappels de notre application avec seulement la date le nom et la note du rappel sans l'image.

Si on clique sur un rappel celle-ci s'ouvre dans un autre fragment de notre activité, Dans ce fragment on peut consulter l'image le nom et la note .

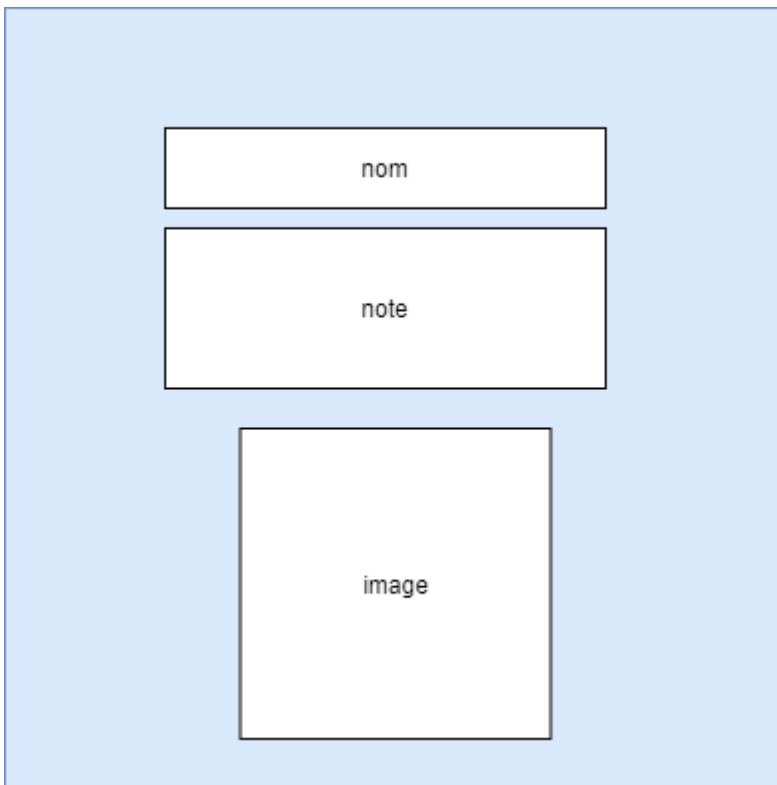
Page d'accueil premier fragment pour ajouter une note :

The diagram shows a user interface for adding a note. It features a light green header bar with two buttons: "ajouter note" and "consulter note". Below the header is a white container with a green rounded rectangle in the center. Inside this green rectangle are two input fields: "nom" and "information". Below the green rectangle are two more buttons: "image" and "enregistrer".

Voici le second fragment qui permet d'avoir la liste des rappels :

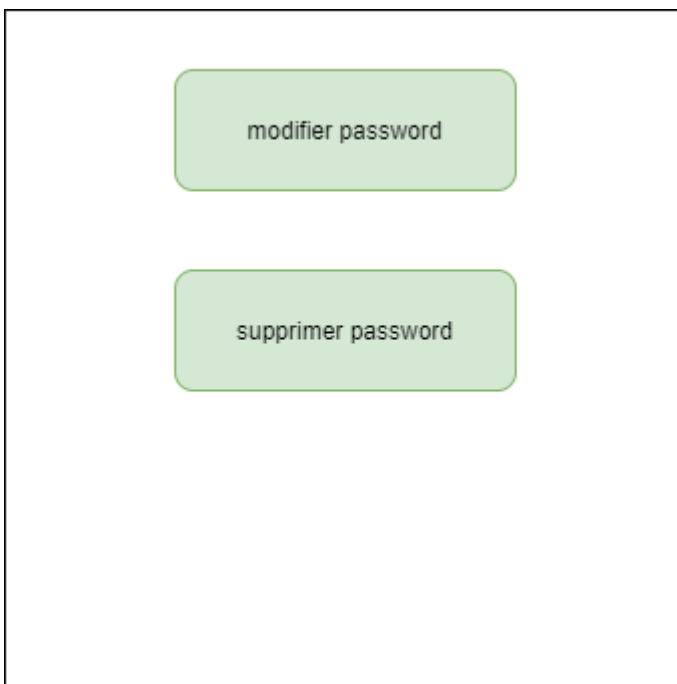
The diagram shows a user interface for displaying a list of reminders. It features a light green header bar with two buttons: "ajouter note" and "consulter note". Below the header is a white container. Inside the container is a table with three columns. The first column is labeled "DATE". The second column is labeled "nom". The third column is labeled "note".

Si l'on souhaite consulter un rappel on clique dessus et un autre fragment s'ouvre :



A diagram showing a light blue rectangular container. Inside the container, there are three white rectangular boxes stacked vertically. The top box is labeled "nom", the middle box is labeled "note", and the bottom box is labeled "image".

Ainsi qu'une page pour pouvoir modifier et supprimer le mot de passe.



A diagram showing a white rectangular container. Inside the container, there are two light green rounded rectangular buttons stacked vertically. The top button is labeled "modifier password" and the bottom button is labeled "supprimer password".

## Réalisation

L'application « PersonalDiary » possède ainsi 7 classes JAVA rangées dans le dossier « modele », ces 7 classes représentent la connection à la base de données par exemple avec le databaseconnector, mon main principal MainActivity, ViewNote, ViewNoteReal, ListActivity, DiarySettings, ainsi que AddNotes. Elles sont toutes construites de la même forme.

### Databaseconnector

Pour enregistrer toutes les données liées au rappel de l'utilisateur ainsi que le mot de passe obligatoire pour ajouter une photo, j'ai utilisé une base de donnée SQLite.

```
public void insertNote(String subject, String note, Bitmap image, String date)
{
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    byte[] img = null;
    if (image != null) {
        image.compress(Bitmap.CompressFormat.JPEG, quality: 100, baos);
        img = baos.toByteArray();
    }
    ContentValues newNote = new ContentValues();
    newNote.put("note", note);
    newNote.put("subject", subject);
    newNote.put("image", img);
    newNote.put("created_at", date);
    open();
    database.insert( table: "notes", nullColumnHack: null, newNote);
    close();
}
```

Par exemple à la création d'une note on entre dans la base de données le nom de la note le sujet ainsi que la photo.

Le Mot de passe par default que j'ai choisi pour ajouter une note est « pass ».

Pour sauvegarder l'état de mon application j'ai utilisé un bundle

Les bundles sont généralement utilisés pour transmettre des données entre diverses activités Android. Cela dépend du type de valeurs que vous souhaitez transmettre, mais les ensembles peuvent contenir tous les types de valeurs et les transmettre à la nouvelle activité.

## Fichier addnote

```
Format formatter;  
formatter = new SimpleDateFormat( pattern: "EEE", Locale.ENGLISH);  
  
currentDayString = formatter.format(now);  
  
formatter = new SimpleDateFormat( pattern: "dd-MMM",Locale.ENGLISH);  
  
currentDateString = formatter.format(now);  
  
formatter = new SimpleDateFormat( pattern: "hh:mm",Locale.ENGLISH);  
currentTimeString = formatter.format(now);  
datetimetxt.setText(" "+currentDayString + "\n\n"+ currentDateString + "\n\n" +currentTimeString);  
  
Bundle extras = getActivity().getIntent().getExtras();
```

A chaque création d'une note on lui associe la date de sa création, que l'on enregistrera aussi dans la table des notes pour la base de donnée.

On affiche la liste des notes dans le `listActivity`. On affiche seulement la date, le nom ainsi que la note dans la preview du rappel .

```
public void onCreate(Bundle savedInstanceState)  
{  
    super.onCreate(savedInstanceState);  
    contactListView = getListView(); // obtenir le ListView intégré  
    contactListView.setOnItemClickListener(viewContactListener);  
    // contactListView.setBackgroundResource(R.drawable.bg);  
    contactListView.setCacheColorHint(android.R.color.transparent);  
    nte= (TextView)findViewById(R.id.nteTextView);  
    String[] from = new String[] { "subject","note","created_at"};  
    int[] to = new int[] { R.id.subjTextView,R.id.nteTextView,R.id.timeTextView };  
    contactAdapter = new SimpleCursorAdapter(  
        context: listActivity.this, R.layout.contact_list_item, c: null, from, to);  
    setListAdapter(contactAdapter);  
}
```

Fonctionnalité tactile

Resté appuyé sur un rappel pour le delete par exemple a l'aide de la fonction deleteContact  
Dans Viewnote .

```
private void deleteContact(final long position) {

    AlertDialog.Builder builder =
        new AlertDialog.Builder(getActivity());

    builder.setTitle("Delete");
    builder.setMessage("Confirm delete");

    builder.setPositiveButton( text: "YES",
        (dialog, button) → {
            final DatabaseConnector databaseConnector =
                new DatabaseConnector(getActivity());

            databaseConnector.deleteContact(position);
            Intent intent = getActivity().getIntent();
            getActivity().finish();
            startActivity(intent);

            Toast.makeText(getActivity(), text: "Deleted", Toast.LENGTH_SHORT).show();

        }
    );
}
```



La fonction pour modifier le mot de passe lorsqu'on ajoute un rappel dans Diarysetting

```
public void set_password() {  
    // Création d'un alertDialog avec un bouton  
    AlertDialog.Builder alertDialog = new AlertDialog.Builder(context: DiarySettings.this);  
  
    alertDialog.setTitle("SET PASSWORD");  
  
    final EditText input = new EditText(context: DiarySettings.this);  
    final EditText input1 = new EditText(context: DiarySettings.this);  
  
    input.setSingleLine(true);  
    input1.setSingleLine(true);  
  
    input.setHint("Enter Password");  
    input1.setHint("Confirm Password");  
  
    LinearLayout ll = new LinearLayout(context: this);  
    ll.setOrientation(LinearLayout.VERTICAL);  
    ll.addView(input);  
    ll.addView(input1);  
}
```

Le sharedpreference a été utilisé pour pouvoir avoir le mot de passe  
Cette classe est utile pour enregistrer les préférences de son application. En utilisant cette méthode, on ne peut stocker que des associations clé, valeur.

L'application est directement codée en anglais cependant j'ai rajouté la langue française grâce au fichier `string.xml`.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Personal Diary</string>
    <string name="enter_pass">Entrer le mot de passe</string>
    <string name="submit">nous faire parvenir</string>
    <string name="close_app">fermer l\'application</string>
    <string name="toast_here_go">bienvenue</string>
    <string name="wrong_pass">Mauvais mot de passe!</string>
    <string name="action_settings">Paramètres</string>
    <string name="add_notes">AJOUTER UNE NOTE</string>
    <string name="view_notes">VOIR LES NOTES</string>
    <string name="subject">"matière "</string>
    <string name="note">REMARQUE</string>

```

### Liste des fonctionnalités de l'application

- ✓ Intégrer au moins 2 activités
- ✓ Utiliser les ressources pour le texte, les couleurs, les images etc.
- ✓ Internationaliser votre application
- ✓ S'adapter à l'orientation de l'écran
- ✓ Gérer la portabilité de votre écran (taille écran, résolution de l'écran, etc.)
- ✓ Intégrer un menu dans la barre des status
- ✓ Inclure le concept de « `SharedPreferences` »
- ✓ Utiliser un « `Bundle` » pour la sauvegarde d'état de votre activité
- ✓ Utiliser un ou plusieurs capteurs
- ✓ Exploiter l'écran tactile
- ✓ Inclure des fragments
- ✓ Utilise Appareil photo / micro
- ✓ Inclure une Base de données embarquée « `SQLite` »

## Conclusion

Pour conclure, dans l'ensemble les objectifs fixés quant à la réalisation de mon application ont été majoritairement respectés. Je n'ai cependant pas géré le cas de l'authentification utilisateur, ce qui déforme complètement la mise en forme de mon application.

Dans un second cas, l'objectif était de rajouter la fonction de la localisation associée à la photo pour pouvoir avoir un aperçu de l'endroit exact de la photo pour par exemple faire une map ou il y aurait noté tous les endroits où l'utilisateur a pris une photo. Ces données saisies se seraient ensuite retrouvées dans une base de données SQLite. La condition manquante était de faire apparaître les données de cette base de données dans une listView, condition que je n'ai pas réussi à réaliser.

Ainsi pour obtenir une application fonctionnelle et finalisée, j'ai préféré opté pour la solution présentée ci-dessus.

## Webographie

<https://gastack.fr/programming/4999991/what-is-a-bundle-in-an-android-application>  
<https://developer.android.com/guide/components/activities/intro-activities>  
<https://developer.android.com/reference/android/app/Activity>  
<https://devstory.net/10433/android-sqlite-database>  
<http://supertos.free.fr/supertos.php?page=1244>  
<https://developer.android.com/reference/android/content/SharedPreferences>