

Exercice 9 Développement Mobile

Introduction

Le projet a été fait sous IntelliJ en utilisant une VM android pour les exécuteurs. Le téléphone utilisé pour faire fonctionner l'application est un Pixel 5 avec comme résolution $1080 \times 2340 pixels$.

Une video est fournie pour indiquer toutes les fonctionnalités de l'application

Quelques class et fonction importantes

Cette class permet d'afficher une boîte de dialog customisé.
J'y affiche un Objet TimePicker qui est un outil fournit par android studio pour sélectionner l'heure ou l'événement ce produit.

```
1 public class CustomDialog extends Dialog {
2     private EditText Description;
3     private EditText Titre;
4     private Button confirmer;
5     private Button annuler;
6     private TimePicker picker;
7     public CustomDialog(Activity activity) {
8         super(activity);
9         setContentView(R.layout.custom_layout_dialog);
10        picker=(TimePicker)findViewById(R.id.timePicker);
11        Description =(EditText)findViewById(R.id.DescriptionEdit);
12        Titre=(EditText)findViewById(R.id.Titre);
13        confirmer = (Button)findViewById(R.id.button);
14        annuler = (Button)findViewById(R.id.button2);
15        picker.setIs24HourView(true);
16    }
17    public Button getannulerbouton(){return annuler;}
18    public Button getconfirmerbouton(){return confirmer;}
19    public EditText getDescription() {return Description;}
20    public EditText getTitre() {return Titre;}
21    public TimePicker getPicker() {return picker;}
22 }
```

Cette fonction de ListEvenement permet de retourner tout les evenements d'une journée sélectionné.

```
1
2 public ArrayList<Evenement> dateEvenement(int year ,int month , int day){
3     ArrayList<Evenement> ret = new ArrayList<>();
4     for(Evenement e : listEvenement){
5         if(year==e.getYear() && month == e.getMonth() && day == e.getDay())
6             ret.add(e);
7     }
8     return ret;
9 }
```

Un adaptateur à été fait pour pouvoir afficher de façon personnalisé la liste des evenement dans la ListView.

```
1 public class EvenementAdapter extends BaseAdapter {
2     private Context context;
3     private List<Evenement> list;
4     private LayoutInflater inflater;
5     public EvenementAdapter(Context context , List<Evenement> list) {
6         this.context = context;
7         this.list = list;
8         this.inflater = LayoutInflater.from(context);
9     }
```

Cette parti du code est appelé dès que l'utilisateur selectionne une date de DatePicker. Elle va ensuite afficher dans le ListView la liste des evenements de la journée sélectionné.

```
1 this.datePicker.init(year , month , day , new DatePicker.OnDateChangedListener
    () {
2     @Override
3     public void onChanged(DatePicker datePicker , int year , int month ,
        int dayOfMonth) {
```