

+*Rappel :*+

* Rendez un document pdf, contenant l'intégralité des schémas, diagrammes, descriptions.

* Une évaluation blanche n'est qu'indicative : elle ne comptera pas dans la moyenne. Le soin apporté aux corrections n'est pas le même que pour l'évaluation finale et les notes blanches ne sont qu'un aperçu de votre travail à un instant t.

h1. DOCUMENTATION : *0/20*

* Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert. [sur 2 points]

Voir fichier PDF.

*_Pas encore

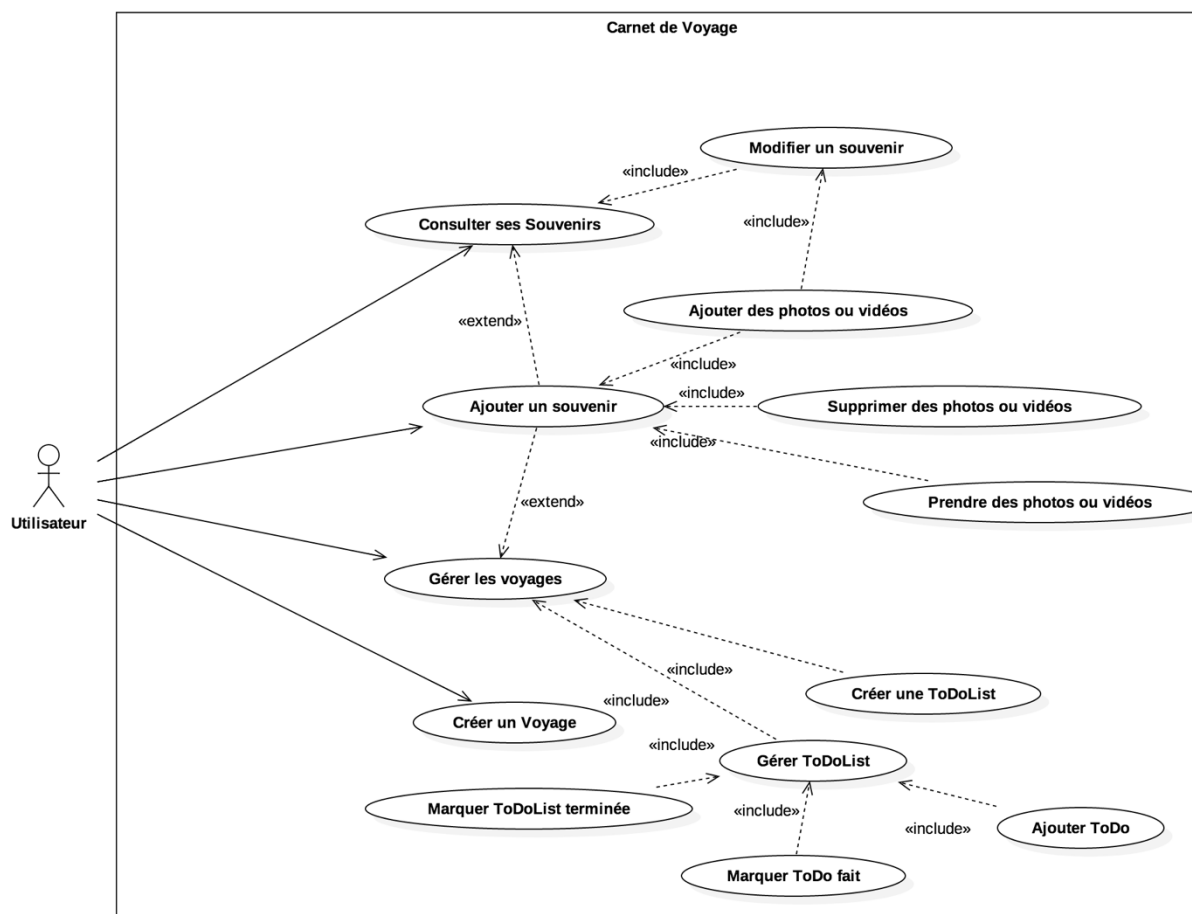
=> 0/2_*

* Je sais faire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application. [sur 5 points]

Voir le fichier PDF.

*_Pas encore

=> 0/5_*

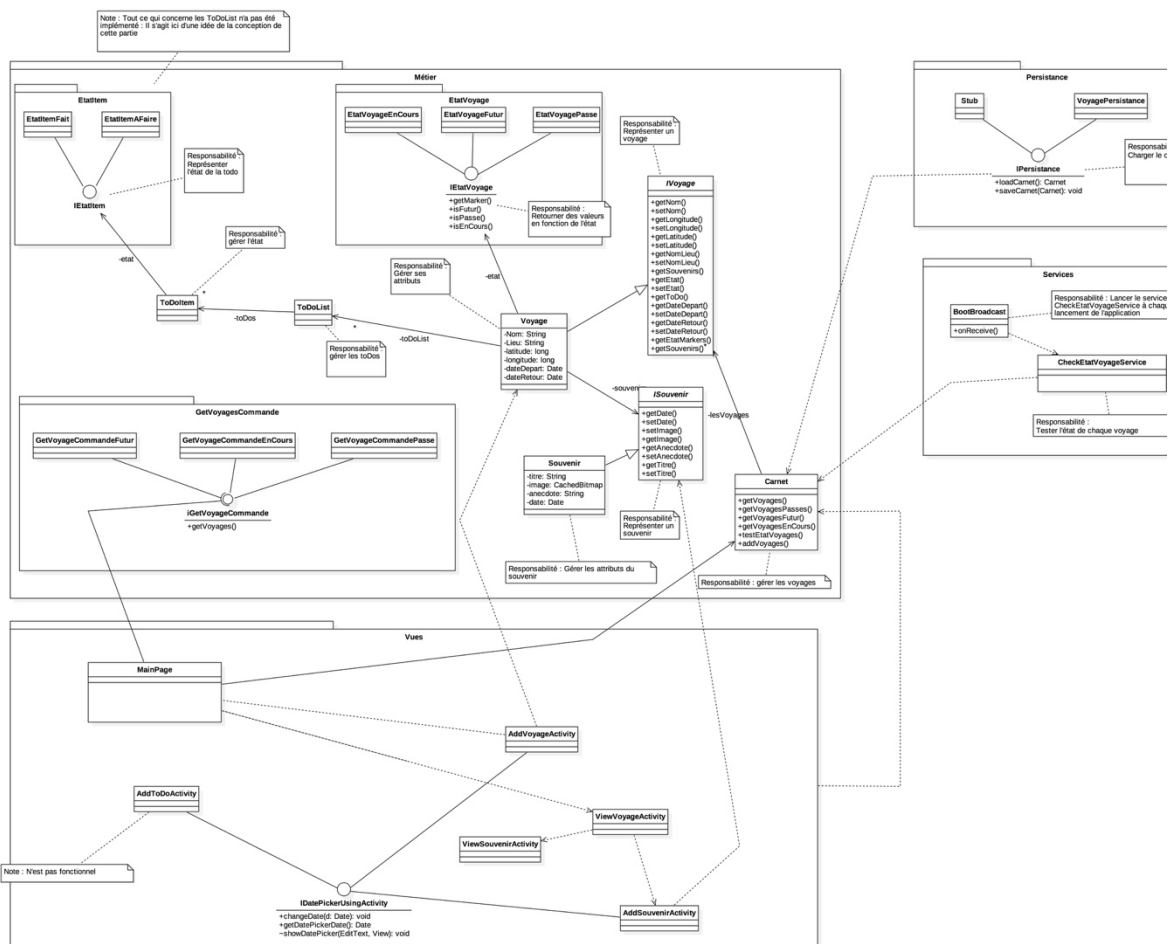


* Je sais concevoir un diagramme UML de qualité représentant mon application.
[sur 7 points]

Voir le fichier PDF.

*_Pas encore

=> 0/7_*



*Je sais décrire un diagramme UML en mettant en valeur et en justifiant les éléments essentiels. [sur 6 points]

Voir le fichier pdf.

*_Pas encore

=> 0/6_*

Voir le diagramme de classe

h1. Programmation : *0/20*

* Je sais utiliser les Intent pour faire communiquer deux activités. [sur 1 point]

```
Intent intent = new Intent(view.getContext(),ViewVoyageActivity.class);
intent.putExtra("index", getCarne().getVoyages().indexOf(v));
```

```
Intent intent = getIntent();
indexVoyage = intent.getIntExtra("index",-1);
```

*_Pas encore

=> 0/1_*

* Je sais développer en utilisant le SDK le plus bas possible. [sur 1 point]

+PREUVE :+

```
android {
    compileSdkVersion 26
    defaultConfig {
        multiDexEnabled true
        applicationId "com.example.alvillafra.travelion"
        minSdkVersion 15
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner
"android.support.test.runner.AndroidJUnitRunner"
    }
}
```

=> 0/1_*

* Je sais distinguer mes ressources en utilisant les qualifier. [sur 1 point]

+PREUVE :+

```
<resources>
    <string name="app_name">Travelion</string>
    <string name="accueil">Accueil</string>
    <string name="v_a_venir">Voyages ? venir</string>
    <string name="from_date">Date de ce souvenir</string>
    <string name="from_dateaffich">Date</string>
    <string name="addSouvenir">Ajouter un souvenir</string>
    <string name="v_en_cours">Voyages en cours</string>
    <string name="v_passe">Voyages pass?s</string>
    <string name="lieu">Lieu de Voyage</string>
    <string name="date_depart">Date de d?part</string>
    <string name="date_retour">Date de retour</string>
    <string name="a_faire">A faire</string>
    <string name="annuler">Annuler</string>
    <string name="valider">Valider</string>
    <string name="ajouter">Ajouter</string>
    <string name="nom_voyage">Nom du Voyage</string>
    <string name="mes_souvenirs">Mes Souvenirs</string>
    <string name="date_souvenir">Date du souvenir</string>
    <string name="title_activity_view_to_do">ViewToDoActivity</string>
</resources>
```

=> 0/1_*

* Je sais modifier le manifeste de l'application en fonction de mes besoins.
[sur 1 point]

<?xml version="1.0" encoding="utf-8"?>

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.alvillafra.travelion">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission
android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="com.example.towntour.permission.MAPS_RECEIVE"
/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-feature android:name="android.hardware.camera" />
```

```

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
        android:name=".vues.MainPage"
        android:windowSoftInputMode="stateHidden|adjustPan">
        <intent-filter>
            <category android:name="android.intent.category.LAUNCHER" />
            <action android:name="android.intent.action.MAIN" />
        </intent-filter>
    </activity>
    <activity android:name=".vues.AjouterSouvenirActivity"></activity>
    <activity android:name=".vues.ViewSouvenirActivity"></activity>
    <activity android:name=".vues.AddVoyagePage" />
    <activity android:name=".vues.ViewVoyageActivity"></activity>
    <activity android:name=".vues.AddToDoActivity"></activity>
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="AIzaSyB0jY1Bz0x_CYIlzyAaZFxANJgnC3Xh59E" />
    <activity
        android:name=".vues.widgets.ViewToDoActivity"
        android:label="@string/title_activity_view_to_do"
        android:theme="@style/AppTheme.NoActionBar"></activity>
</application>
</manifest>

```

*_Pas encore
=> 0/1_*

* Je sais faire des vues xml en utilisant layouts et composants adéquats. [sur 1 point]

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_margin="5pt" android:focusable="true"
        android:focusableInTouchMode="true"
    >
        android:background="@android:color/holo_blue_dark">
        <Button
            android:gravity="center"
            android:id="@+id/menu"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:background="@android:color/holo_red_dark"
            android:text="Ajouter une image"

```

```

        ></Button>
<ImageView
    android:id="@+id/imageS"
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:layout_gravity="center"
/>
<EditText
    android:id="@+id/titre"
    android:gravity="center_horizontal"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
    android:hint="Titre" />
<EditText
    android:id="@+id/souvenir"
    android:layout_width="fill_parent"
    android:layout_height="150dp"
    android:background="@drawable/rounded_border_edittext"
    android:inputType="textPersonName"
    android:hint="Entrez votre souvenir"
/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/date_souvenir"/>
<EditText
    android:id="@+id/etxt_fromdate"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
/>
<Button
    android:id="@+id/buttonsave"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Enregistrer"
/>
<android.support.constraint.Guideline
    android:id="@+id/guideline"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
/>
<android.support.constraint.Guideline
    android:id="@+id/guideline2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
/>
<Button
    android:id="@+id/buttonReturn"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Retour"
/>
</LinearLayout>
</ScrollView>

```

*_Pas encore

=> 0/1_*

* Je sais coder proprement mes activités, en m'assurant qu'elles ne font que relayer les événements. [sur 1 point]

+PREUVE :+

Ce sont nos classes qui gèrent les événements. Nos activités relayent les événements pas ces classes.

=> 0/1_*

* Je sais coder une application en ayant un véritable métier. [sur 2 points]

Notre projet est composé de classes et interfaces constituant les éléments présents dans les activités.

*_Pas encore

=> 0/2_*

Voir la classe MyDatePickerDialog ou encore MainPage

* Je sais parfaitement séparer vue et modèle. [sur 1 point]

*_Pas encore

=> 0/1_*

Voir classe AjouterSouvenirActivity

* Je maîtrise le cycle de vie de mon application. [sur 1 point]

@Override

```
protected void onCreate(@Nullable Bundle savedInstanceState) {  
    Intent intent = getIntent();  
    indexVoyage = intent.getIntExtra("index", -1);  
    if(indexVoyage == -1){  
        intent.putExtra("error", "Impossible de récupérer le voyage  
selectionné");  
        setResult(RESULT_CANCELED, intent);  
    }  
    dateFormatter = new SimpleDateFormat("dd/MM/yy", Locale.FRANCE);  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.add_memory);  
    findViewById();  
}
```

@Override

```
protected void onPause(){
```

```
    super.onPause();  
    Log.i(tag, "onSave");  
}
```

@Override

```
protected void onStart(){
```

```
    souvenirEdit.setText(anecdote);  
    titreEdit.setText(titreS);  
    super.onStart();  
    Log.i(tag, "onStart");  
}
```

@Override

```
protected void onDestroy()
```

```
{  
    Log.i(tag, "onDestroy");  
    super.onDestroy();  
}
```

```
protected void onRestore()
```

```
{
```

```

        Log.i(tag, "onRestore");
    }
    @Override
    protected void onStop()
    {
        super.onStop();

        Log.i(tag, "onStop");
    }

```

*_Pas encore
=> 0/1_*

* Je sais utiliser le findViewById à bon escient. [sur 1 point]

```

public void findViewByIds()
{
    dateEditText = (EditText) findViewById(R.id.etxt_fromdate);
    dateEditText.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showDatePicker((EditText) findViewById(R.id.editTextDateRetour),v);
        }
    });
    dateEditText.setFocusable(false);
    dateEditText.setInputType(InputType.TYPE_NULL);
    dateEditText.invalidate();
    souvenirEdit = (EditText) findViewById(R.id.souvenir);
    titreEdit = (EditText) findViewById(R.id.titre);
    this.imageView = (ImageView) this.findViewById(R.id.images);
    Button menu=(Button)findViewById(R.id.menu);
    menu.setOnClickListener(new View.OnClickListener(){
        public void onClick(View v) {
            Log.i(tag, "boutonPhoto appuye");
            showMenu(v);
        }
    });
    Button retour = (Button) findViewById(R.id.buttonReturn);
    retour.setOnClickListener(new View.OnClickListener(){
        public void onClick(View v) {
            //setResult(0,intentBack);
            setResult(RESULT_OK);
            finish();
        }
    });
    Button save = (Button)findViewById(R.id.buttonsave);
    save.setOnClickListener(new View.OnClickListener(){
        public void onClick(View v) {
            Log.d(tag, "Saving memory");
            saveMemory(v);
        }
    });
}

```

*_Pas encore
=> 0/1_*

* Je sais gérer les permissions dynamiques de mon application. [sur 1 point]

Dans le manifest :

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.CAMERA"/>

<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COM-
MANDS" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="com.example.towntour.permission.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission an-
droid:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />

<uses-feature android:name="android.hardware.camera" />
```

*_Pas encore
=> 0/1_*

* Je sais gérer la persistance légère de mon application. [sur 1 point]

```
@Override
public void onSaveInstanceState(Bundle savedInstanceState) {
    savedInstanceState.putString("anecdoteKey", anecdote);
    savedInstanceState.putString("titre1Key", titreS);
    savedInstanceState.putString("dateKey", date.toString());
    savedInstanceState.putParcelable("imageKey", image);
    super.onSaveInstanceState(savedInstanceState);
    Log.i(tag, "Memory saved");
}
```

*_Pas encore
=> 0/1_*

* Je sais gérer la persistance profonde de mon application. [sur 1 point]

+PREUVE :+

```
public class CarnetPersistance implements IPersistance {
    public CarnetPersistance() {
    }
    private static final String filename = "carnet_save";
    /**
     * Retourne le carnet depuis un fichier
     * @param context Context de l'application
     * @return
     */
    @Override
    public Carnet loadCarnet(Context context) {
        FileInputStream inputStream;
        try{
            inputStream = context.openFileInput(filename);
            ObjectInputStream ois = new ObjectInputStream(inputStream);
            Carnet c = (Carnet) ois.readObject();
            if(c!=null){
                Log.i("pers", "C'est nickel");
                return c;
            }
            Log.i("pers", "Bon au moins y a pas d'exception");
            return new Carnet();
        }
        catch (Exception e){
        }
```



```

        e.printStackTrace();
        return new Carnet();
    }
}
/**
 * Sauvegarde le carnet dans un fichier
 * @param c le carnet
 * @param context context de l'application
 */
@Override
public void saveCarnet(Carnet c, Context context) {
    FileOutputStream outputStream;
    try {
        outputStream = context.openFileOutput(filename,
Context.MODE_PRIVATE);
        ObjectOutputStream oos = new ObjectOutputStream(outputStream);
        oos.writeObject(c);
        outputStream.close();
        Log.i("pers", "Pareil pas d'exception");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

=> 0/1_*

* Je sais afficher une collection de données. [sur 1 point]

Que mettre ?

*_Pas encore

=> 0/1_*

* Je sais coder mon propre adaptateur. [sur 2 points]

```

public class SouvenirsAdapter extends ArrayAdapter<ISouvenir> {
    public SouvenirsAdapter(@NonNull Context context, int resource, @NonNull
List<ISouvenir> objects) {
        super(context, resource, objects);
    }
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
        ISouvenir s = getItem(position);
        if(convertView == null){
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.travel_view_item,parent,
false);
        }
        TextView tvNom = convertView.findViewById(R.id.view_name_travel);
        TextView tvLieu = convertView.findViewById(R.id.view_place_travel);
        tvNom.setText(s.getTitre());
        DateFormat format = new SimpleDateFormat("dd/MM/yy");
        tvLieu.setText(format.format(s.getDate()));
        return convertView;
    }
}
}

```

*_Pas encore

=> 0/2_*

```
* Je maîtrise l'usage des fragments. [sur 2 points]
package com.example.alvillafra.travelion.vues.widgets;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.DialogFragment;
import android.app.Fragment;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.DatePicker;
import com.example.alvillafra.travelion.R;
import com.example.alvillafra.travelion.vues.AddVoyagePage;
import com.example.alvillafra.travelion.vues.IDatePickerUsingActivity;
import java.util.Calendar;
import java.util.Date;
/**
 * Created by alvillafra on 28/02/18.
 */
public class MyDatePickerDialog extends DialogFragment {
    private Button cancelButton;
    private Button validateButton;
    private DatePicker datePicker;
    private IDatePickerUsingActivity activity;
    public static MyDatePickerDialog newInstance(){
        Fragment f = new MyDatePickerDialog();
        return (MyDatePickerDialog) f;
    }
    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, @Nullable Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.date_picker_layout,container,false);
        cancelButton = (Button) v.findViewById(R.id.annuler);
        validateButton = (Button) v.findViewById(R.id.valider);
        datePicker = (DatePicker) v.findViewById(R.id.datePicker);
        Calendar c = Calendar.getInstance();
        activity = (IDatePickerUsingActivity) getActivity();
        Date d = activity.getDatePickerDate();
        if(d != null){
            c.setTime(d);
        }
        datePicker.updateDate(c.get(Calendar.YEAR),c.get(Calendar.MONTH),c.get(Calendar.
DAY_OF_MONTH));
        return v;
    }
    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        cancelButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dismiss();
            }
        });
    }
}
```

```

});
validateButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Date d = new Date(datePicker.getYear()-
1900,datePicker.getMonth(),datePicker.getDayOfMonth());
        Log.i("Year","Année : " + datePicker.getYear());
        activity = (IDatePickerUsingActivity) getActivity();
        try{
            activity.changeDate(d);
            dismiss();
        }
        catch(Exception e){
            AlertDialog.Builder builder = new
AlertDialog.Builder(activity);
            builder.setMessage(e.getMessage())
                .setTitle("Vérifiez vos dates");
            AlertDialog a = builder.create();
            a.show();
        }
    }
});
}
}

```

*_Pas encore
=> 0/2_*

* Je maîtrise l'utilisation de Git. [sur 1 point]
 +PREUVE :+
 *_Pas encore
 => 0/1_*

Voir les différentes branches créées en fonction de la fonctionnalité en cours de développement

h1. Application : *0/20*

* Mon application présente un intérêt à être publié sur le store. [sur 5 points]
 +PREUVE :+
 *_Pas encore
 => 0/5_*

Voir le contexte. Permet aux utilisateurs d'organiser leurs voyages et d'y enregistrer les moments clés

* Mon application fonctionne de manière à être utilisée par le public. [sur 5 points]
 Il manque toute la fonctionnalité ToDoList mais le reste fonctionne : reste à cacher les boutons liés à la todoList
 *_Pas encore
 => 0/5_*

* Mon application utilise des contraintes spécifiées lors du choix du projet. [sur 5 points]
 Caméra, Explorateur de fichier, Maps de l'API Google
 *_Pas encore
 => 0/5_*

* Mon application utilise les contraintes à bon escient. [sur 5 points]
 Ajouter des photos, et montrer sur une carte le lieu de voyage

*_Pas encore
=> 0/5_*