

Compte Rendu : My Pearl 10/02/2016

Lors de cette séance, j'ai g rer les requ tes pour l'appli gr ce   une classe. La fonctionnalit  d'upload des photos   partir du smartphone est donc maintenant rajout , ainsi, une fois la photo upload sur le serveur nous pouvons l'afficher dans le « carrousel » de l'application afin qu'il Like ou Passe les collections.

Le design final a  t  d fini avec Maeva de l' quipe L'ESSCA, il y a donc eue modification de la page d'accueil et de la page des logins (cot  design, en back c'est jacques qui g re).

Nous avons aussi choisie la solution pour l'h bergement qui sera « Hostinger », je vais aussi demand s l'ouverture d'un port   l' cole afin que l'on ait une VM pour l'application.

Lors du prochain sprint j'aurai pour but de terminer la fonction LIKE ET PASSE avec les photos qui apparaitront dans le carrousel principal.

Les +	Les -
<ul style="list-style-type: none">• Objectifs respect�s• H�bergement enfin choisie	<ul style="list-style-type: none">• Probl�me lors de l'affichage sur le carrousel principal, l'image uploader n'apparait pas � l'�cran.• Probl�me lors du push pour le code : Code mis dans le compte rendu.....

Code == MainActivity

```
package com.example.andji.testimagemaperle;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.AsyncTask;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
```

Andritiana RAZAFINJATO
BACHELOR 3

```
import android.widget.Toast;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.HashMap;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    public static final String UPLOAD_URL =
"http://simplifiedcoding.16mb.com/ImageUpload/upload.php";
    public static final String UPLOAD_KEY = "image";
    public static final String TAG = "MY MESSAGE";

    private int PICK_IMAGE_REQUEST = 1;

    private Button buttonChoose;
    private Button buttonUpload;
    private Button buttonView;
    private ImageView imageView;

    private Bitmap bitmap;

    private Uri filePath;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        buttonChoose = (Button) findViewById(R.id.buttonChoose);
        buttonUpload = (Button) findViewById(R.id.buttonUpload);
        buttonView = (Button) findViewById(R.id.buttonViewImage);

        imageView = (ImageView) findViewById(R.id.imageView);

        buttonChoose.setOnClickListener(this);
        buttonUpload.setOnClickListener(this);
    }

    private void showFileChooser() {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Selectionnez
la photo"), PICK_IMAGE_REQUEST);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK &&
data != null && data.getData() != null) {

            filePath = data.getData();
            try {
                bitmap =
MediaStore.Images.Media.getBitmap(getContentResolver(), filePath);
                imageView.setImageBitmap(bitmap);
            }
        }
    }
}
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public String getStringImage(Bitmap bmp){
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        bmp.compress(Bitmap.CompressFormat.JPEG, 100, baos);
        byte[] imageBytes = baos.toByteArray();
        String encodedImage = Base64.encodeToString(imageBytes,
Base64.DEFAULT);
        return encodedImage;
    }

    private void uploadImage(){
        class UploadImage extends AsyncTask<Bitmap,Void,String>{

            ProgressDialog loading;
            RequestHandler rh = new RequestHandler();

            @Override
            protected void onPreExecute() {
                super.onPreExecute();
                loading = ProgressDialog.show(MainActivity.this, "Upload en
cours", "Patientez...",true,true);
            }

            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                loading.dismiss();

                Toast.makeText(getApplicationContext(),s,Toast.LENGTH_LONG).show();
            }

            @Override
            protected String doInBackground(Bitmap... params) {
                Bitmap bitmap = params[0];
                String uploadImage = getStringImage(bitmap);

                HashMap<String,String> data = new HashMap<>();
                data.put(UPLOAD_KEY, uploadImage);

                String result = rh.sendPostRequest(UPLOAD_URL,data);

                return result;
            }
        }

        UploadImage ui = new UploadImage();
        ui.execute(bitmap);
    }

    @Override
    public void onClick(View v) {
        if (v == buttonChoose) {
            showFileChooser();
        }
        if(v == buttonUpload){
            uploadImage();
        }
    }
}
```

```
    }  
    }  
}
```

Code == RequestHandler (classe)

```
package com.example.andji.testimagemaperle;  
  
    import java.io.BufferedReader;  
    import java.io.BufferedWriter;  
    import java.io.InputStreamReader;  
    import java.io.OutputStream;  
    import java.io.OutputStreamWriter;  
    import java.io.UnsupportedEncodingException;  
    import java.net.HttpURLConnection;  
    import java.net.URL;  
    import java.net.URLEncoder;  
    import java.util.HashMap;  
    import java.util.Map;  
  
    import javax.net.ssl.HttpsURLConnection;  
  
/**  
 * Created by Andji on 10/02/2016.  
 */  
public class RequestHandler {  
  
    public String sendGetRequest(String uri) {  
        try {  
            URL url = new URL(uri);  
            HttpURLConnection con = (HttpURLConnection)  
url.openConnection();  
            BufferedReader bufferedReader = new BufferedReader(new  
InputStreamReader(con.getInputStream()));  
  
            String result;  
  
            StringBuilder sb = new StringBuilder();  
  
            while((result = bufferedReader.readLine())!=null) {  
                sb.append(result);  
            }  
  
            return sb.toString();  
        } catch (Exception e) {  
            return null;  
        }  
    }  
  
    public String sendPostRequest(String requestURL,  
                                HashMap<String, String> postDataParams) {  
  
        URL url;  
        String response = "";
```

```
try {
    url = new URL(requestURL);

    HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
    conn.setReadTimeout(15000);
    conn.setConnectTimeout(15000);
    conn.setRequestMethod("POST");
    conn.setDoInput(true);
    conn.setDoOutput(true);

    OutputStream os = conn.getOutputStream();
    BufferedWriter writer = new BufferedWriter(
        new OutputStreamWriter(os, "UTF-8"));
    writer.write(getPostDataString(postDataParams));

    writer.flush();
    writer.close();
    os.close();
    int responseCode = conn.getResponseCode();

    if (responseCode == HttpURLConnection.HTTP_OK) {
        BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        response = br.readLine();
    } else {
        response = "Erreur d'enregistrement";
    }
} catch (Exception e) {
    e.printStackTrace();
}

return response;
}

private String getPostDataString(HashMap<String, String> params) throws
UnsupportedEncodingException {
    StringBuilder result = new StringBuilder();
    boolean first = true;
    for (Map.Entry<String, String> entry : params.entrySet()) {
        if (first)
            first = false;
        else
            result.append("&");

        result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
    }

    return result.toString();
}
}
```