

# esi better

2CPI

## RAPPORT DU PROJET

SOUS LA SUPERVISION DE:

\*Zakaria Gheid

**MEMBRES:**

\*Hamza Bendahmane

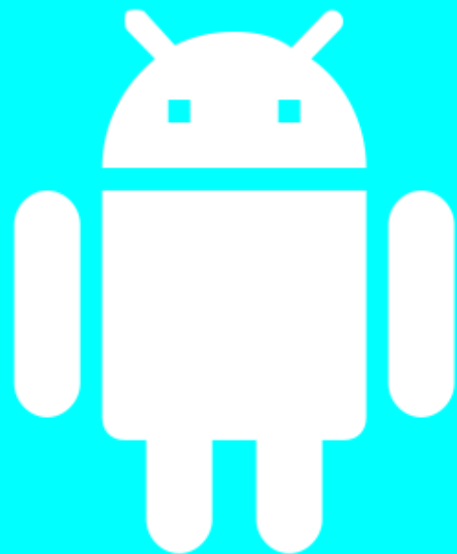
\*Djihane Bouziane

\*Boussaada Walid

\*Manel Boufeldja

\*Nour El Imene Larbaoui

\*Anouar Manaa



## Table des matières :

Introduction :	5
Les objectifs :	5
Le domaine d'utilisation :	5
Le principe du fonctionnement :	5
Les grandes fonctionnalités :	5
1-partage des tutoriels et des cours :	5
2-Echange des idées et des articles :	6
Conception :	6
1-Article :	6
Ajouter article :	7
Commenter l'article :	8
Meilleurs articles :	9
2-Cours :	10
Tutoriels :	10
Résumés :	16
BDD(NosQL firebase database) :	16
A quoi sert une base NosQL ?	16
Firestore :	17
Authentification :	17
Authentification avec google :	18
Méthodes de connexion :	19
Vérification des emails :	20
Liste des utilisateurs :	20
Cloud Firestore :	21
Cloud storage :	21
Realtime database :	22
Firebase Cloud Functions :	23

<i>Réalisation :.....</i>	<i>26</i>
<i>Les outils et langages utilisés :.....</i>	<i>26</i>
<i>Les permissions :.....</i>	<i>27</i>
<i>Planning du travail :.....</i>	<i>28</i>
<i>Sturcture du projet :.....</i>	<i>28</i>
<i>Aperçu de l'application :.....</i>	<i>29</i>
<i>Conclusion :.....</i>	<i>37</i>
<i>Perspectives :.....</i>	<i>38</i>

## Remerciements

*Avant tout, nous remercions Allah tout puissant, pour nous avoir donné la force et la persévérance de mener à bien ce projet.*

*Nous remercions notre encadreur : Mr GHEID ZAKARIA pour ses précieux conseils et pour nous avoir accompagné durant la réalisation, et les membres du jury pour l'honneur qu'ils nous accordent en acceptant d'évaluer notre travail.*

*Nous sommes également très reconnaissants envers les amis et collègues qui nous ont été un appui moral et intellectuel tout au long du déroulement de ce projet.*

*Finalement, nous remercions tous ceux et celles qui ont contribué de près ou de loin à l'aboutissement de cet humble travail.*

## **INTRODUCTION :**

C'est grâce à ce mini projet que nous avons eu l'opportunité de cumuler les connaissances avec celles de la pratique, dans le cadre de la 2<sup>ème</sup> Année du cycle préparatoire de l'ESI SBA ceci permet également de rentrer dans la vie universitaire quotidienne de l'étudiant et essayer de la faciliter.

Notre application ESI better est conçue pour rassembler des tâches importantes dont l'étudiant d'ESI a besoin dans ses études, comme l'échange des idées et des articles entre les étudiants, partage des cours et des tutoriels et tout ce qui utile, et d'autres fonctions intéressantes.

## **LES OBJECTIFS :**

- Partage des tutoriels, des cours et des résumés.
- Echange des idées et des articles entre les étudiants.
- Faciliter les études et gagne du temps.
- Obtenir des ressources d'éducation gratuit.
- Développer et enrichir les connaissances de l'étudiant.
- Mettre les étudiants à jour avec leurs idées et déterminer les meilleurs.

## **LE DOMAINE D'UTILISATION :**

Cette application est destinée seulement aux étudiants de 1<sup>ère</sup> et 2<sup>ème</sup> année du cycle préparatoire d'ESI SBA.

## **PRINCIPE DE FONCTIONNEMENT :**

ESI better est une application mobile Android, pour la fonctionner il suffit que l'étudiant créer un compte à l'aide de son adresse email et bénéficier les fonctionnalités.

## **LES GRANDES FONCTIONNALITES :**

### **1-Partage des tutoriels et des cours :**

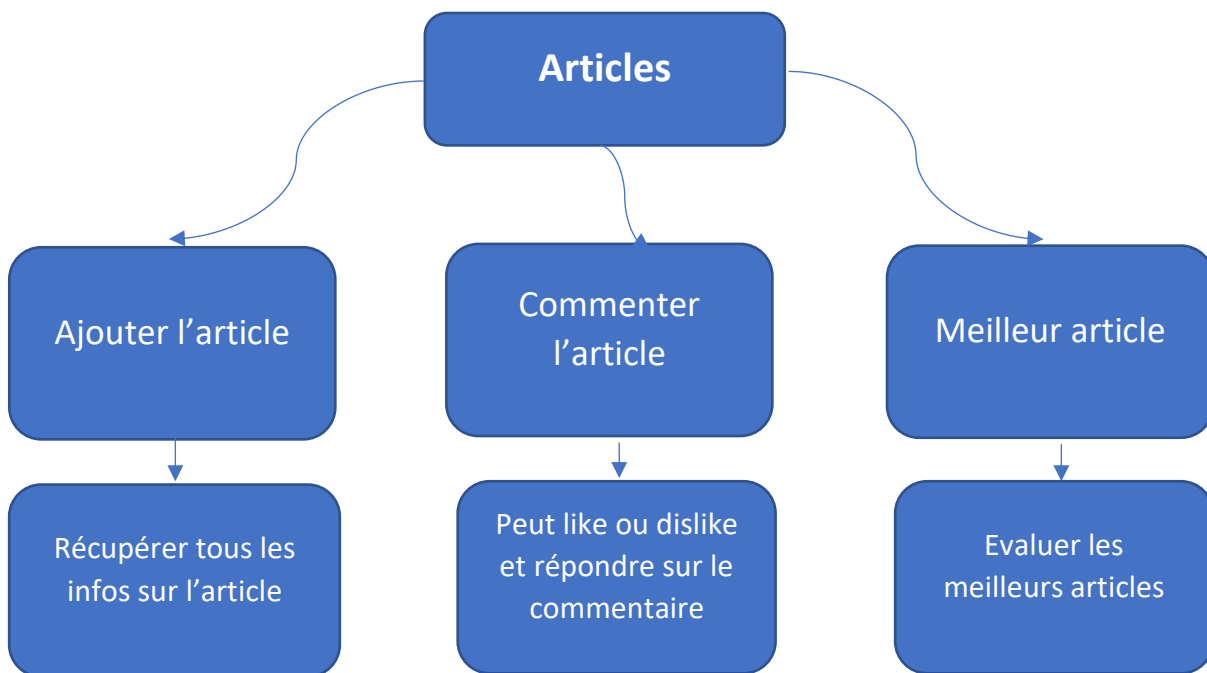
- Ajouter des tutoriels et des cours intéressants pour tous les modules.
- Ajouter des vidéos éducatifs.
- Visionnement et chargement des vidéos et des tutoriels.

## 2-Echange des idées et des articles :

- Ajouter des idées et des articles.
- Ajouter des commentaires sur les idées partagées.
- like et dislike et évaluer les meilleurs idées.

## CONCEPTION :

**1 /Article :** Cette partie contient tout ce qui concerne les articles, qui est composée de trois principales fonctionnalités :



## 1 -1/[Ajouter Article](#) :

La classe Article\_item.java permet de récupérer tous les informations sur l'utilisateur et son article publié :

### Class\_Article\_item.java :

```
m Article_item()  
m Article_item(String, String, String, String, String, Long, Long)  
m getBody():String  
m getDate():String  
m getDislikes():Long  
m getImage():String  
m getLikes():Long  
m getPostId():String  
m getTitle():String  
m getUid():String  
m setBody(String):void  
m setDate(String):void  
m setDislikes(Long):void  
m setImage(String):void  
m setLikes(Long):void  
m setPostId(String):void  
m setTitle(String):void  
m setUid(String):void
```

Et la classe Add\_Article.java permet d'ajouter l'article et sauvegarder les informations récupérées.

### Class\_Add\_Article.java :

```

public void add(String title, String body, Uri image, final AlertDialog dialog
    , String keyDocument, String likes, String dislikes) {
    DocumentReference ref = FirebaseFirestore.getInstance()
        .collection( collectionPath: "posts").document(keyDocument);
    final HashMap<String, Object> map = new HashMap<>();

    map.put("title", title);
    map.put("body", body);
    map.put("uid", uid);
    map.put("date", date);
    map.put("image", image.toString());
    map.put("likes", Long.parseLong(likes));
    map.put("dislikes", Long.parseLong(dislikes));
    map.put("PostId", keyDocument);
    ref.set(map).addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()) {
                Toast.makeText(getApplicationContext(), "Post Uploaded Successfully",
                    Toast.LENGTH_SHORT).show();
                dialog.dismiss();
                onBackPressed();
            } else
                Toast.makeText(getApplicationContext(), "failed Posting , Try again"
                    , Toast.LENGTH_SHORT).show();
        }
    });
}

```

## 1-2/[Commenter l'article](#) :

La méthode Add\_Comment de la classe Article\_activity.java aider à ajouter des commentaires sur un article :

### Add\_Comment ():



```

public void add_comment(View view) {
    final TextInputEditText addtext = findViewById(R.id.add_commento);
    if (!TextUtils.isEmpty(addtext.getText().toString())) {
        FirebaseFirestore dataBase = FirebaseFirestore.getInstance();
        HashMap<String, Object> commMap = new HashMap<>();
        String tex = addtext.getText().toString().trim();
        Calendar cc = Calendar.getInstance();
        String month = String.valueOf(cc.get(Calendar.MONTH) + 1);
        String date = cc.get(Calendar.YEAR) + "/" + month + "/" + cc.get(Calendar.DAY_OF_MONTH);
        commMap.put("uid", uid);
        commMap.put("likes", 0);
        commMap.put("date", date);
        commMap.put("comment", tex);
        DocumentReference ref = dataBase.collection( collectionPath: "comments").
            document( documentPath: "posts");
        // generate a very specific random reference ///---->TIME(ms)+Date ;
        String ss = cc.getTimeInMillis() + cc.get(Calendar.DAY_OF_MONTH) + month + cc.get(Calendar.YEAR);
        commMap.put("commentId", ss);
        ref.collection(postId).document(ss).set(commMap).addOnCompleteListener(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful()) {
                    Toast.makeText(getApplicationContext(), "success", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(getApplicationContext(), "failed", Toast.LENGTH_SHORT).show();
                }
            }
        });
        addtext.setText("");
        adapter.notifyDataSetChanged();
    }
}

```

### 1-3/[Meilleurs articles :](#)

Les articles sont classifiés par rapport le nombre des likes obtenus , et la classe bestOf.java réalise cette fonctionnalité :

#### Class\_bestOf.java :

```

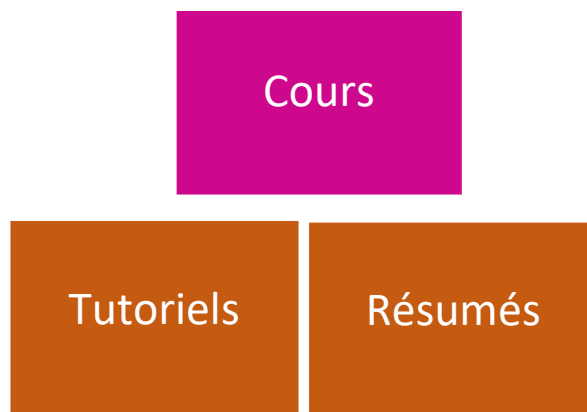
public bestOf() {
    // Required empty public constructor
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.ideas_fragment_best_of, container, attachToRoot: false);
}

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    loadLocal();
    query = FirebaseFirestore.getInstance()
        .collection( collectionPath: "posts")
        .orderBy( field: "likes", Query.Direction.DESENDING).orderBy( field: "dislikes", Query.Direction.ASCENDING);
    options =
        new FirestoreRecyclerOptions.Builder<Article_item>()
            .setQuery(query, Article_item.class).build();
    setupRecyclerAdapter();
}

```

## 2/Cours :



### 2-1/Tutoriels :

1. Nous créons la classe Add\_tutorial qui est constitués de tout les nécessaires traitements pour ajouter un tutoriel :

#### Class Add\_Tutorial.java :

Code	Commentaire
<pre>public class Add_Tutorial extends AppCompatActivity {     public final String uid = FirebaseAuth.getInstance().getCurrentUser().getUid();     public final CollectionReference reference = FirebaseFirestore.getInstance()         .collection("posts");     public Uri image_art, file_art;     final Calendar cc = Calendar.getInstance();     final String month = String.valueOf(cc.get(Calendar.MONTH) + 1);     final String date = cc.get(Calendar.YEAR) + "/" + month + "/" +         cc.get(Calendar.DAY_OF_MONTH);     String titleArt;     TextView post_article;     String dislikes = "0";     Spinner yearSpin, ModuleSpin;     String Module;     int Type, Year;     StorageReference images_url;     ImageView show_picked_image;     String BodyArt;     boolean haveChoosedTheType, haveChoosedTheModule;     String likes = "0";     EditText TitleBody;</pre>	Déclaration des variables pour récupérer tous les informations sur le tutoriel.

<pre>String <b>typeof</b> = "add"; <b>boolean</b> Done = <b>false</b>; CircleImageView <b>photo</b>; ArrayAdapter&lt;CharSequence&gt; <b>adapterModule</b>;  @Override <b>protected void</b> onCreate(Bundle savedInstanceState) {     <b>super</b>.onCreate(savedInstanceState);     setContentView(R.layout.<b>courses_add_tutorial</b>) ;     <b>typeof</b> = getIntent().getExtras().getString("type");     <b>post_article</b> = findViewById(R.id.<b>post_Tutorial</b>);     <b>photo</b> = findViewById(R.id.<b>picture</b>);     <b>show_picked_image</b> = findViewById(R.id.<b>imageView6</b>);     <b>TitleBody</b> = findViewById(R.id.<b>Title</b>);     <b>images_url</b> = FirebaseStorage.getInstance().getReference("Images/" + <b>uid</b>);     <b>ModuleSpin</b> = findViewById(R.id.<b>ModuleSpin</b>);     <b>yearSpin</b> = findViewById(R.id.<b>yearSpin</b>);</pre>	<p>Initialisation des variables.</p>	
<pre>findViewById(R.id.<b>Add</b>).setOnClickListener(<b>new</b> View.OnClickListener() { @Override     <b>public void</b> onClick(View v) {         pick_Image(<b>null</b>);}  }); findViewById(R.id.<b>imageView2</b>).setOnClickListener(<b>new</b> View.OnClickListener() {     @Override     <b>public void</b> onClick(View v) {         pick_Video(<b>null</b>);     } });</pre>	<p>Choisir une image ou /et sélectionner un fichier.</p>	
<pre><b>yearSpin</b>.setEnabled(<b>true</b>); <b>yearSpin</b>.setSelection(0); <b>final</b> ArrayAdapter&lt;CharSequence&gt; adapterYear = ArrayAdapter.createFromResource(<b>this</b>,     R.array.<b>year</b>, android.R.layout.<b>simple_spinner_item</b>);  adapterYear.setDropDownViewResource(android.R.layout.<b>simple_spinner_dropdown_item</b>); <b>yearSpin</b>.setAdapter(adapterYear); <b>yearSpin</b>.setOnItemSelectedListener(<b>new</b> AdapterView.OnItemSelectedListener() {     @Override     <b>public void</b> onItemSelected(AdapterView&lt;?&gt; adapterView, View view, <b>int</b> i, <b>long</b> l) {         <b>Year</b> = i;         <b>haveChooosedTheType</b> = <b>!(Type==0)</b>;         <b>haveChooosedTheModule</b> = <b>!(Year==0)</b>;         <b>if</b> ( <b>haveChooosedTheModule</b> )             <b>ModuleSpin</b>.setEnabled(<b>true</b>);         <b>else</b> {             <b>ModuleSpin</b>.setEnabled(<b>false</b>);             <b>Done</b> = <b>false</b>;         }         <b>if</b> (<b>Year==1</b>)         {             <b>adapterModule</b> = ArrayAdapter.createFromResource(Add_Tutorial.<b>this</b>,                 R.array.<b>first_modules_titles</b>, android.R.layout.<b>simple_spinner_item</b>);  <b>adapterModule</b>.setDropDownViewResource(android.R.layout.<b>simple_spinner_dropdown_item</b>); <b>ModuleSpin</b>.setAdapter(<b>adapterModule</b>);  <b>ModuleSpin</b>.setOnItemSelectedListener(<b>new</b> AdapterView.OnItemSelectedListener() {     @Override     <b>public void</b> onItemSelected(AdapterView&lt;?&gt; adapterView, View view, <b>int</b> i, <b>long</b> l) {         <b>Module</b> = <b>ModuleSpin</b>.getSelectedItem().toString();         <b>Done</b> = <b>true</b>;     }     @Override     <b>public void</b> onNothingSelected(AdapterView&lt;?&gt; adapterView) {</pre>	<p>Traitement pour déterminer l'année correspondante de tutoriel publié.</p> <p>1CPI</p>	

<pre> });  } else if(Year==2){     adapterModule = ArrayAdapter.createFromResource(Add_Tutorial.this,         R.array.second_modules_titles, android.R.layout.simple_spinner_item);      adapterModule.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);     ModuleSpin.setAdapter(adapterModule);      ModuleSpin.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()     {         @Override         public void onItemSelected(AdapterView&lt;?&gt; adapterView, View view, int i, long l) {             Module = ModuleSpin.getSelectedItem().toString();             Done = true;         }          @Override         public void onNothingSelected(AdapterView&lt;?&gt; adapterView) {          }     }); } </pre>	<p><b>2CPI</b></p>
<pre> 0 post_article.setOnClickListener(new View.OnClickListener() {     @Override     public void onClick(View v) {         Bundle bundle = getIntent().getExtras();          String title = TitleBody.getText().toString().trim();         boolean imagesSET = image_art != null;         boolean filesSet = file_art!=null;          if (TextUtils.isEmpty(title))             Toast.makeText(Add_Tutorial.this, R.string.title_shou_em, Toast.LENGTH_SHORT).show();         else if (!imagesSET)             Toast.makeText(Add_Tutorial.this, R.string.add_im, Toast.LENGTH_SHORT).show();         else if (!filesSet)             Toast.makeText(Add_Tutorial.this, R.string.add_file_pl, Toast.LENGTH_SHORT).show();         else {             titleArt = TitleBody.getText().toString().trim();             final String keyDocument = uid + cc.getTimeInMillis();             StorageReference reference = FirebaseStorage.getInstance().                 getReference("Images/" + uid + "/" + keyDocument);             final StorageReference photoArt = reference.child("thumbnail.png");             View vv = LayoutInflater.from(Add_Tutorial.this).inflate(R.layout.general_layout_image,                 null, false);             final AlertDialog dialog = new             AlertDialog.Builder(Add_Tutorial.this).setTitle(R.string.uploading_file)                 .setView(vv).create();             dialog.show();             if (typeof.equals("add")) {                 uploadFromUri(file_art,title);             }         }     } }); } </pre>	<p>Lorsque l'utilisateur cliquer sur le bouton de publication.</p> <p>Prendre des vérifications pour effectuer la publication :</p> <p>Le titre ne doit pas être vide.</p> <p>Ajouter une image.</p> <p>Ajouter un fichier .</p>
<pre> private void uploadFromUri(Uri fileUri, String pdfname) {     startService(new Intent(this, MyUploadService.class)         .putExtra(MyUploadService.EXTRA_FILE_URI, fileUri) </pre>	<p>Enregistrer le fichier URI</p> <p>Commencer MyUploadService pour télécharger le fichier,meme si</p>

<pre>         .putExtra(MyUploadService.EXTRA_IMAGE_URI, image_art)         .putExtra("year", String.valueOf(Year))         .putExtra("title", pdfname)          .putExtra("module", Module)         .putExtra("pdf", false)         .putExtra("date", date)         .setAction(MyUploadService.ACTION_UPLOAD));      Toast.makeText(this, R.string.progress_uploading, Toast.LENGTH_LONG).show();      Intent intent = new Intent(this, Profile_Activity.class);     startActivity(intent);     finish(); } </pre>	<p>cette activité est tué ou mise en arrière_plan.</p> <p>Selectionner le module.</p>
---	---

2. Afficher tous les tutoriels pour chaque année et pour chaque module :

### Class\_ShowAllTutorials.java :

```

public class ShowAllTutorials extends AppCompatActivity {
    public DatabaseReference reference = null;
    int what_yearIs;
    String moduleName;
    Query query;
    FirebaseRecyclerAdapter<File_item, ViewHolder> adapter;
    FirebaseRecyclerOptions<File_item> options;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_show_all_tutorials);
        Bundle b = getIntent().getExtras();
        if (b.isEmpty())
            onBackPressed();
        else {
            what_yearIs = b.getInt("year");
            moduleName = b.getString("name");
        }
        updateView();
        query = FirebaseDatabase.getInstance()
            .getReference().child("Summaries").child(String.valueOf(what_yearIs))
            .child(String.valueOf(moduleName));
        options = new FirebaseRecyclerOptions.Builder<File_item>()
            .setQuery(query, File_item.class)
            .build();
        HandleTutos();
    }
}

```

Quelle année et quel module.

Récupérer les données à partir de la base de données.

\*Les classes secondaires utilisées :

```
private void updateView() {  
    if (what_yearIs == 1) {  
        ((TextView)  
findViewById(R.id.title_sub11)).setText(R.string.st_year);  
        ((TextView)  
findViewById(R.id.title_mod)).setText(moduleName);  
        ((ImageView)  
findViewById(R.id.icon_year1)).setImageDrawable(getDrawable(R.  
drawable.ic_one));  
    } else {  
        ((ImageView)  
findViewById(R.id.icon_year1)).setImageDrawable(getDrawable(R.  
drawable.ic_222));  
        ((TextView)  
findViewById(R.id.title_sub11)).setText(R.string.st_year);  
        ((TextView)  
findViewById(R.id.title_mod)).setText(moduleName) ;}}}
```

Modifier la vue pour le module sélectionné.

```
private void SearchFor(String field) {  
  
com.google.firebase.database.Query query1 =  
    query.startAt("title", field)  
  
        .  
        endAt("title", field + "\uf8ff");  
FirebaseRecyclerOptions<File_item> options1 = new  
    FirebaseRecyclerOptions.Builder<File_item>()  
        .setQuery(query1, File_item.class)  
        .build();  
  
    adapter.updateOptions(options1);  
}
```

Recherche dans la liste des tutoriels publiés pour chaque module.

```
public class ViewHolder extends RecyclerView.ViewHolder {  
    private MaterialCardView root;  
    private TextView txtTitle;  
    private ImageView imagePoster;  
    public ViewHolder(final View itemView) {  
        super(itemView);  
        txtTitle = itemView.findViewById(R.id.textGrid);  
        imagePoster = itemView.findViewById(R.id.image_game);  
        root =itemView.findViewById(R.id.card_n); }  
}
```

Occuper l'instanciation de la vue d'un seul index du recyclerview

```

private void HandleTutos() {

    com.google.firebase.database.Query query = FirebaseDatabase.getInstance()
        .getReference().child("Tutorials").child(String.valueOf(what_yearIs)).limitToFirst(20).order
        ByChild("module").equalTo(moduleName);

    FirebaseRecyclerOptions<File_item> options = new
        FirebaseRecyclerOptions.Builder<File_item>()
            .setQuery(query, File_item.class)
            .build();

    adapter = new FirebaseRecyclerAdapter<File_item, ViewHolder>(options) {
        @NonNull
        @Override
        public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
            viewType) {
            View view =
                LayoutInflater.from(parent.getContext()).inflate(R.layout.courses_file_item, parent,
                    false);
            return new ViewHolder(view);
        }

        @Override
        protected void onBindViewHolder(@NonNull ViewHolder holder, int position,
            @NonNull File_item model) {
            //loader.hide();
            holder.txtTitle.setText(model.getTitle());
            holder.root.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Bundle b = new Bundle();
                    File_item item = adapter.getItem(position);
                    b.putString("date", item.getDate());
                    b.putString("title", item.getTitle());
                    b.putString("ref", adapter.getRef(position).toString());
                    b.putString("uid", item.getUid());
                    b.putString("link", item.getPostId());
                    b.putString("PostId", item.getPostId());
                    b.putString("image", item.getThumbnail());
                    Intent i = new Intent(getApplicationContext(), MediaPlayer.class);
                    i.putExtras(b);
                    startActivity(i);
                }
            });
            holder.setImageFile(model.getThumbnail());
        }
    };

    RecyclerView recyclerview = findViewById(R.id.grid_modules);
    recyclerview.setLayoutManager( new GridLayoutManager>ShowAllTutorials.this,2,
        GridLayoutManager.VERTICAL,false));
    recyclerview.setItemViewCacheSize(50);
    recyclerview.setAdapter(adapter); }

```

Récupérer tous les tutoriels publiés pour le module sélectionné à partir de la base de données.

Adapter d'affichage la liste des tutoriels.

Un LayoutInflater pour définir un fichier sélectionné parmi la liste.

Récupérer le tutoriel à partir d'adapter pour lire la vidéo.

Ajouter une image pour le tutoriel.

Un recyclerview pour afficher la liste des tutoriels.

## 2-2/Résumés :

On effectuons presque les mêmes traitements pour les résumés, on utilisant les classes Add\_Summaries.java et ShowAllSummaries.java elles appliquent les mêmes opérations que Add\_Tutoriel.java et ShowAllTutorials.java respectivement.

## • **BDD (NOSQL Firebase Database) :**

Les bases de données NoSQL sont de plus en plus utilisées mais savez-vous vraiment ce qu'elles renferment ? Prenons les exemples de Datastax (Cassandra) et MongoDB qui figurent parmi les solutions NoSQL les plus utilisées du marché.

## **A quoi sert une base NoSQL ?**

Les bases relationnelles ont été « inventées » par Edgar F. Codd en 1970. A l'époque, ce fut une approche révolutionnaire (modèle relationnel des données, algèbre de relations, sélections, projections, jointures, etc...). Oracle fut la première société à commercialiser un produit conforme à ce concept, dès 1977, avec le succès que l'on connaît. Ces dernières années, plusieurs avis nuancés ont été émis concernant le modèle relationnel :

- \* Difficultés pour sauver et récupérer des grappes d'objets avec des programmes écrits en Java ou C#... (on parle de « mismatch d'impédance »)
- \* Difficultés pour satisfaire les besoins des applications Web à grande échelle (nouveaux besoins métiers, nouvelles contraintes techniques)

Voici les 3 aspects qui ont motivé la nouvelle vague de bases NoSQL (Not Only SQL...) :

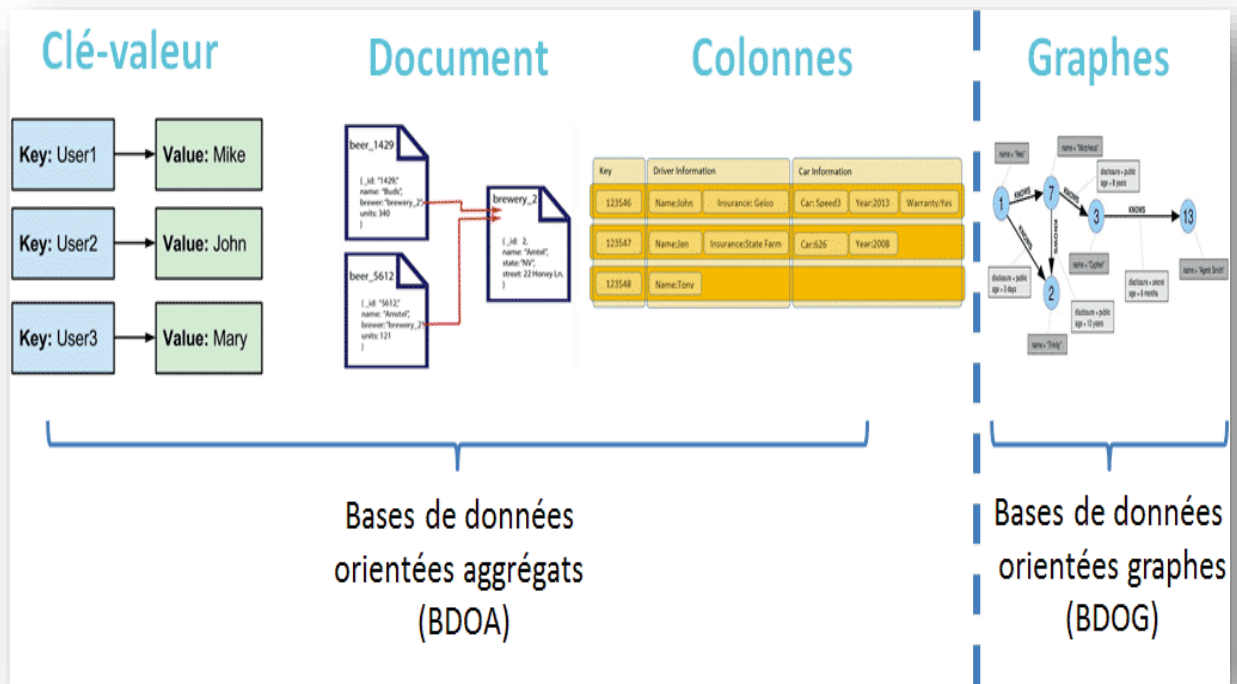
- \* Le VOLUME des données créées double tous les 2 ans. IDC estime qu'en 2020 le volume atteindra 44 Zettabytes (1 ZB = 1 milliards de terabytes)
- \* La VARIÉTÉ des types de données créées (Smartphones)
- \* La VÉLOCITÉ avec laquelle les données changent est également très importante (Internet of Things)

En fait, le terme base NoSQL définit une nouvelle génération de produits qui ne suivent pas le modèle relationnel. Mais l'architecture de ces produits varie beaucoup entre eux.

## **Les types de bases de données NoSQL :**

Clé- valeur, document, colonnes et graphes sont les 4 types de bases de données NoSQL. Etudions d'un peu plus près chacun de ces types.





Notre type choisi est le type "Clé-valeur" ce qu'on appelle en Firebase « Firebase Realtime Database » et on a utilisée aussi Firestore Firebase qu'utilise les Documents .

## ● FIREBASE :

Firebase vous offre des fonctionnalités telles que les analyses, les bases de données ,la messagerie et les rapports d'incidents pour que vous puissiez vous déplacer rapidement et vous concentrer sur vos utilisateurs, nous la utilisant pour manipuler notre application :

### 1/Authentification :

Firebase Authentification fournit des services de backend, des SDK faciles à utiliser et des bibliothèques d'interface utilisateur prêtes à l'emploi pour authentifier les utilisateurs auprès de votre application. Il prend en charge l'authentification à l'aide de mots de passe, de numéros de téléphone, de fournisseurs d'identité fédérés populaires tels que Google, Facebook, Twitter etc...

## Firestore Authentication avec Google :

```
void firebaseAuthWithGoogle(final GoogleSignInAccount acct) {
    AuthCredential credential = GoogleAuthProvider.getCredential(acct.getIdToken(), null);
    firebaseAuth.signInWithCredential(credential)
        .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's information
                    FirebaseUser user = firebaseAuth.getCurrentUser();
                    if (!task.getResult().getAdditionalUserInfo().isNewUser())
                        goToProfile();
                    else {
                        boolean succ = false;
                        User_Account newUser = null;
                        try {
                            newUser = new User_Account(acct.getPhotoUrl(), acct.getDisplayName(),
                                status: "is Student", wilaya: "16", isMale: true, birthday: "01/01/1999");
                        } catch (URISyntaxException e) {
                            e.printStackTrace();
                        }
                        succ = newUser.SaveData(user, newUser);
                        if (succ) {
                            Snackbar.make(signin, "Done", Snackbar.LENGTH_LONG).show();
                            goToProfile();
                        }
                    }
                } else {
                    // If sign in fails, display a message to the user.
                    Snackbar.make(signin, "Authentication Failed.",
                        Snackbar.LENGTH_SHORT).show();
                }
            }
        });
}
```

**Méthodes de connexion :** il y a deux possibles méthodes pour se connecter :

### 1-Avec email et mot de passe :

```
private void signin() {
    TextInputEditText SignUpMail = findViewById(R.id.email_field1);
    TextInputEditText SignUpPass = findViewById(R.id.password_field1);
    String email = SignUpMail.getText().toString();
    final String pass = SignUpPass.getText().toString();
    if (TextUtils.isEmpty(email)) {
        SignUpMail.requestFocus();
        SignUpMail.setError(getString(R.string.empty_email));
    } else if (!isValidEmail(email)) {
        SignUpPass.requestFocus();
        SignUpPass.setError(getString(R.string.r_u_esist2));
    }
    if (TextUtils.isEmpty(pass)) {
        SignUpPass.requestFocus();
        SignUpPass.setError(getString(R.string.empty_password));
    } else if (!isValidPassword(pass)) {
        SignUpPass.requestFocus();
        SignUpPass.setError(getString(R.string.short_password));
    } else if (isValidEmail(email) && isValidPassword(pass)) {
        firebaseAuth.signInWithEmailAndPassword(email, pass)
            .addOnCompleteListener(login.this, new OnCompleteListener<AuthResult>() {
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (!task.isSuccessful()) {
                        Toast.makeText(login.this, R.string.error +
                            task.getException().getLocalizedMessage(),
                                Toast.LENGTH_LONG).show();
                    } else {
                        Snackbar.make(signin, getString(R.string.Login_successful),
                            Snackbar.LENGTH_LONG).show();
                        final FirebaseUser user = firebaseAuth.getCurrentUser();
                        if (!user.isEmailVerified()) {
                            verify_your_email(user);
                        } else {
                            goToProfile();
                        }
                    }
                }
            })
    }
}
```

### 2-Avec Google :

```
private void signin_google() {
    GoogleSignInOptions gso = new
    GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(R.string.default_web_client_id))
        .requestEmail()
        .build();
    GoogleSignInClient mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
    Intent signInIntent = mGoogleSignInClient.getSignInIntent();
    startActivityForResult(signInIntent, RC_SIGN_IN);
}
```

## Vérification des emails :

\*Nous utilisons l'API email vérification pour vérifier l'existence, la validité et la qualité de toute adresse email :

<https://emailverification.whoisxmlapi.com/api>

```
private void verify_your_email(final FirebaseUser user) {  
  
    if (user != null && user.getEmail().contains("@esi-sba.dz")) {  
        user.getId();  
        if (!user.isEmailVerified()) {  
            final AlertDialog dialog = new AlertDialog.Builder(login.this).create();  
            dialog.setTitle(getString(R.string.verify_your_email));  
            dialog.setMessage(getString(R.string.email_sent));  
  
            LayoutInflater inflater = LayoutInflater.from(login.this);  
            final View login_layout = inflater.inflate(R.layout.general_layout_verify_email,  
null);  
  
            final Button verify = login_layout.findViewById(R.id.verify);  
            dialog.setView(login_layout);  
            dialog.show();  
            verify.setOnClickListener(new View.OnClickListener() {  
                @Override  
                public void onClick(View v) {  
                    firebaseAuth.getCurrentUser().reload();  
                    if (!user.isEmailVerified()) {  
                        Snackbar.make(verify, getString(R.string.yremail_not_vrf),  
Snackbar.LENGTH_LONG).show();  
                    } else {  
                        dialog.dismiss();  
                        goToProfile();  
                    }  
                }  
            });  
        } else {  
            goToProfile();  
        }  
    }  
}
```

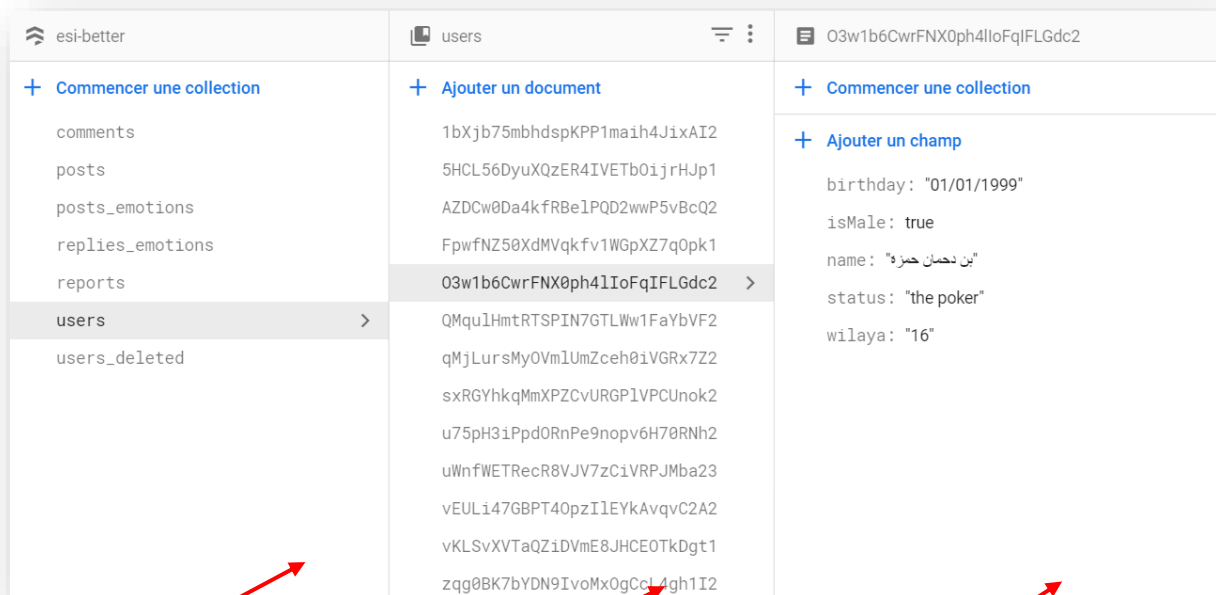
## liste des utilisateurs :

Identifiant	Fournisseurs	Date de création	Dernière connexion	ID utilisateur ↑
n.larbaoui@esi-sba.dz	✉	30 mars 2020	1 avr. 2020	5HCL56DyuXQzER4IVETbOijrHJp1
m.boufeldja@esi-sba.dz	✉	29 mars 2020	21 juil. 2020	qMjLursMyOVmUmZceh0iVGRx7Z2
h.bendahmane@esi-sba.dz	✉	19 avr. 2020	29 août 2020	sxRGYhkqMmXPZCvURGPIVPCUn...
d.bouziane@esi-sba.dz	✉	28 mars 2020	5 mai 2020	u75pH3iPpdORnPe9nopv6H70RnH2
m.boussaada@esi-sba.dz	✉	31 mars 2020	2 juin 2020	vEULi47GBPT40pzllEYkAvqvC2A2

## 2/ Cloud Firestore :

Cloud Firestore est une base de données NoSQL flexible et évolutive pour le développement mobile, Web et serveur de Firebase et de Google Cloud Platform.

En suivant le modèle de données NoSQL de Cloud Firestore, vous stockez les données dans des documents contenant des champs mappés à des valeurs. Ces documents sont stockés dans des collections, qui sont des conteneurs pour vos documents que vous pouvez utiliser pour organiser vos données et créer des requêtes. Les documents prennent en charge de nombreux types de données , des chaînes simples et des nombres aux objets complexes et imbriqués. Vous pouvez également créer des sous-collections dans des documents et créer des structures de données hiérarchiques qui évoluent à mesure que votre base de données se développe. Le modèle de données Cloud Firestore prend en charge la structure de données la mieux adaptée à votre application.



Collections






documents

champs

## 3/Cloud Storage :

Cloud Storage for Firebase est un service de stockage d'objets puissant, simple et économique conçu pour l'échelle de Google.

Nous utilisons Cloud storage car nous avons besoin de stocker et de diffuser du contenu généré par l'utilisateur, tel que les images et tutoriels et les résumés :

 Nom	Taille	Type	Dernière modification
 Images/	–	Dossier	–
 Summaries/	–	Dossier	–
 Tutorials/	–	Dossier	–
 posts/	–	Dossier	–

#### 4/Realtime Database :

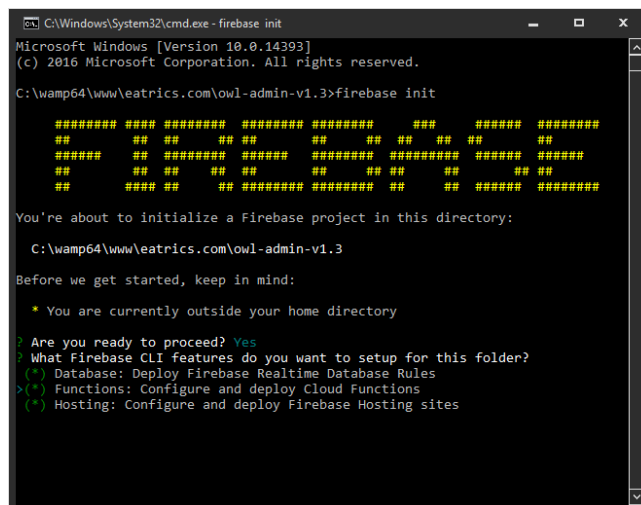
La base de données Firebase Realtime est une base de données hébergée dans le cloud. Les données sont stockées au format JSON et synchronisées en temps réel avec chaque client connecté. Lorsque vous créez des applications multiplateformes avec les SDK iOS, Android et JavaScript de Firebase, tous vos clients partagent une instance de base de données en temps réel et reçoivent automatiquement des mises à jour avec les données les plus récentes :

```
esi-better
├── Notifications
│   └── sxRGYhkqMmXPZCvURGPIVPCUnok2
├── notSeen
│   └── sxRGYhkqMmXPZCvURGPIVPCUnok2
│       └── likes_count: 0
├── replies
│   ├── sxRGYhkqMmXPZCvURGPIVPCUnok21597081737328
│   └── sxRGYhkqMmXPZCvURGPIVPCUnok21597082060516
└── users
    ├── 5HCL56DyuXQzER4IVETb0ijrHJp1: "eLkDDTrEDAg:APA91bEvm8JlSCDiHE6taeYMLGtCyTBhFdZ
    └── sxRGYhkqMmXPZCvURGPIVPCUnok2: "cWV7HTqdQ0K34csusbVyR8:APA91bE3dKA0dpKRTs5f1V5A
```

## 4/Firebase Cloud Functions :

Cloud Functions for Firebase est un framework sans serveur qui vous permet d'exécuter automatiquement du code backend en réponse aux événements déclenchés ;Triggers ; par les fonctionnalités Firebase et les requêtes HTTPS. Votre code JavaScript ou TypeScript est stocké dans le cloud de Google et s'exécute dans un environnement géré. Il n'est pas nécessaire de gérer et de mettre à l'échelle vos propres serveurs

On a choisi le langage Node js por la déploiements des Fonctionnements . et on a utilise Firebase CLI comme environnement.



```
C:\Windows\System32\cmd.exe - firebase init
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\wamp64\www\eatrics.com\owl-admin-v1.3>firebase init

#####
##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##  ##
#####

You're about to initialize a Firebase project in this directory:

  C:\wamp64\www\eatrics.com\owl-admin-v1.3

Before we get started, keep in mind:

  * You are currently outside your home directory

> Are you ready to proceed? Yes
> What Firebase CLI features do you want to setup for this folder?
  (*) Database: Deploy Firebase Realtime Database Rules
  (*) Functions: Configure and deploy Cloud Functions
  (*) Hosting: Configure and deploy Firebase Hosting sites
```

```
exports.sendLikeNotification = functions.firestore.document("posts/{post_id}").onWrite(async (change,context)=> {
```

```
    const post_id = context.params.post_id;
    let tokenId;
    let newPost ;
    const admin_id = change.after.data().uid;
    const tokenRef = admin.database().ref('users/'+admin_id);
    const likes_num = change.after.data().likes;
    const dislikes_num = change.after.data().dislikes;
    const body = change.after.data().body;
    const date = change.after.data().date;
    newPost = likes_num === 0 && dislikes_num === 0 ;
    if (newPost)
    return null ;

    const post_name = change.after.data().title;
        const image = change.after.data().image;

    // The array containing all the user's tokens.

    await tokenRef.once("value").then(function(snapshot) {
        return tokenId = snapshot.val();
    });
});
```



```

const payload = {
  data:{
    post_id:post_id,
    body:body,
    title: post_name,
    date: date,
    reply: "",
    uid: admin_id,
    image: image,
    name: "",
    number : "",
    comment: "",
    likes:likes_num.toString(),
    dislikes:dislikes_num.toString(),
    type:"like"
  }
};

console.log("liked");

const time = Date.now();

```

```

const userNotification = admin.database().ref('/Notifications/'+admin_id+'/'+post_id);

await userNotification.set({seen: false, post_id: post_id, time: time, title: post_name,
likes:likes_num, name: "", number : "", comment: "",reply: "", date: date, image: image, body:
body, dislikes:dislikes_num, type:"like" });

```

```

return admin.messaging().sendToDevice(tokenid, payload);

```

Node.js 8 has been deprecated.						
Function	Trigger	Region	Runtime	Memory	Timeout	
sendCommentNotific...	document.create comments/posts/{post_id}/{comment_id}	us-central1	Node.js 8 Deprecated	256 MB	60s	
sendLikeNotificati...	document.write posts/{post_id}	us-central1	Node.js 8 Deprecated	1 GB	200s	⋮
sendRepliesNotific...	ref.create replies/{post_id}/{comment_id}/{reply_id}	us-central1	Node.js 8 Deprecated	256 MB	60s	
updateNotSeen	ref.write Notifications/{userId}/{notificationId}	us-central1	Node.js 8 Deprecated	1 GB	200s	

## Réalisation :

## LES OUTILS ET LANGAGES UTILISES :



**Android Studio** est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS et Linux.



**L'Extensible Markup Language**, généralement appelé XMLnote 1, « langage de balisage extensible1 » en français, est un métalangage informatique de balisage générique qui est un sous-ensemble du Standard Generalized Markup Language (SGML). Sa syntaxe est dite « extensible » car elle permet de définir différents langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS, SVG... Elle est reconnaissable par son usage des chevrons (<, >) encadrant les noms des balises. L'objectif initial de XML est de faciliter l'échange automatisé de contenus complexes (arbres, texte enrichi, etc.) entre systèmes d'informations hétérogènes (interopérabilité). Avec ses outils et langages associés.



**Java** est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications, mais qui ont l'inconvénient d'être plus lourd à l'exécution (en mémoire et en temps processeur) à cause de sa machine virtuelle. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.



JSON (JavaScript Object Notation) est un format d'échange de données en texte lisible. Il est utilisé pour représenter des structures de données et des objets simples dans un code qui repose sur un navigateur Web



**Firebase** est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, JavaScript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Le et par James Templin, le service est racheté par Google en octobre 2014.



**Adobe XD** est une solution d'UX/UI design complète pour la conception de sites web, d'applications mobiles, etc. Créez, prototypez et partagez, le tout depuis Adobe XD.

Alliant rapidité, précision et qualité, Adobe XD permet aux designers de modifier et partager facilement des prototypes interactifs avec collaborateurs et réviseurs sur l'ensemble des appareils et plates-formes, dont Windows, Mac, iOS et Android.



**Node JS « JavaScript »** est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de Node.js.



**Photoshop** est un logiciel de retouche, de traitement et de dessin assisté par ordinateur. Édité par Adobe, il est principalement utilisé pour le traitement des photographies numériques, mais sert également à la création ex nihilo d'images.

## LES PERMISSIONS :

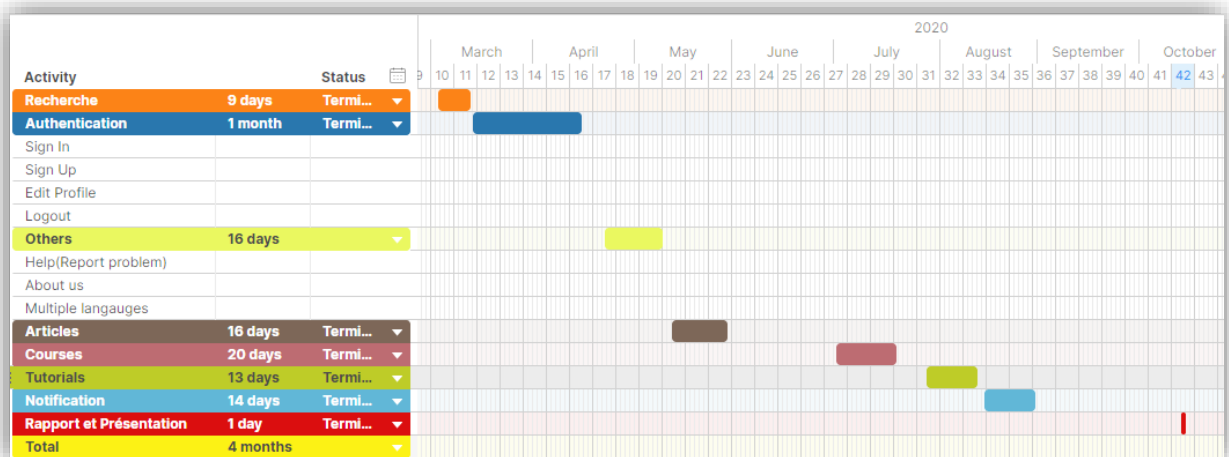
- \*Accès à Internet.
- \*Accès au Stockage.
- \*Accès pour envoyer des notifications.

## Structure du projet :

### Project Structure (Main Activities & Features)



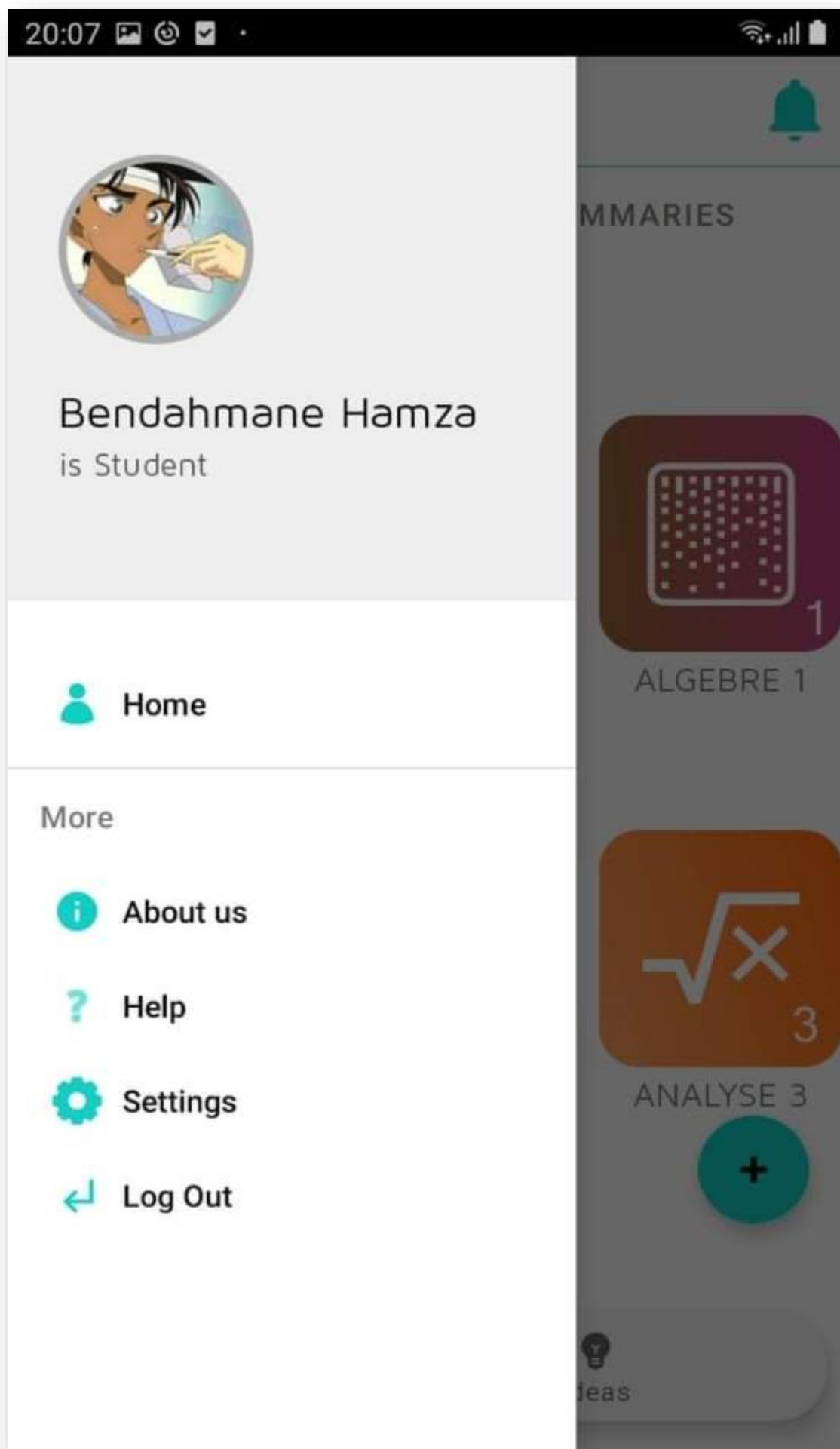
## Planning de travail :



## Aperçu sur notre application :

On vous présente notre application ESIbetter :

The logo for ESIbetter, featuring the word 'esi' in a teal, lowercase, sans-serif font, positioned above the word 'better' in a grey, lowercase, sans-serif font.



# Settings

## Profile



Edit Profile



Change Password



## More



**Notifications**



Language




log out



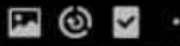
Looking for something ?

---

**2** st Year  SFD



20:07



المشاعير . الحبر الصلي ليس حبيب

البقاء في المنزل ليس سيئا لهذه الدرجة



منازلهم



hddnnememmrk

By

Bendahmane  
Hamza

2020/8/10



1

0

add article

esi  
betterLooking for something ?

NEW

BEST OF

0   1

hddnnemem

Bendahmane  
Hamza1   0

bsbsnne

Bendahmane  
Hamza

Courses



Ideas

2020/8/10

1

0

add article

Your comment

POST



Bendahmane Hamza  
hgufjdb



Reply

2020/8/12



Bendahmane Hamza

ربنلاوي



2020/8/13



Bendahmane Hamza

## Notifications

المشاهير : العجوز الصبي ليس صغيراً  
القاء في المنزل ليس صيداً لهذه الدرجة



**hddnnemememrk**

New Comment from : Bendahmane  
Hamza it said : bdbdbenenyour post  
has 4 comments .

Sat Aug 15 19:16:57 GMT+01:00 2020

## Conclusion :

Notre équipe a pu, durant le second semestre de cette année venir à bout du projet qui lui a été soumis.

Nous avons veillé à répondre le plus fidèlement possible au cahier des charges et aux attentes de nos clients à travers le produit final que notre application constitue. Nous avons par ailleurs, à travers ce rapport, expliquer et apporter toutes les informations et supports nécessaires à l'éclaircissement de l'approche que nous avons suivie.

Notre travail a suivi brièvement le cheminement suivant Il a d'abord été question de définir notre problématique, celle à laquelle notre projet doit répondre puis de définir nos objectifs, ceux que nous devons mener à bien à l'issue du projet. Il a fallu par la suite choisir nos outils de programmation, de gestion de projet et de communication pour pouvoir entamer la phase d'apprentissage puis de réalisation selon ce qui a été exposé précédemment.

Nous sommes sortis enrichis de cette expérience éducative puisque nous avons acquis des connaissances et compétences dans divers aspects.

D'une part, l'aspect relationnel puisque Le travail en équipe nous a permis de prendre conscience de l'importance du travail collectif par rapport au travail individuel, d'apprendre à toujours maintenir la communication entre les membres de l'équipe afin d'assurer une cohésion dans le travail, de respecter et de prendre en compte les idées et propositions de chacun et d'ouvrir le champ au débat pour permettre l'émergence de nouvelles idées

D'autre part, l'aspect technique. Nous avons effectivement appris à manipuler de nouveaux langages de programmation, pris connaissance d'une nouvelle notion à la base de l'informatique à savoir les bases de données et appris à résoudre un problème informatique en répondant à un cahier des charges précis

Nous sommes certaines qu'il nous reste encore beaucoup de choses à apprendre de nos enseignants et de nos recherches personnelles. La quête du savoir n'a pas de limites ni dans le temps, ni dans son volume.

## Perspectives :

Les fonctionnalités dont dispose notre application satisfont les consignes du cahier des charges. Ces fonctionnalités peuvent cependant être améliorées et apporter une valeur ajoutée à l'application comme il est également possible d'enrichir l'application par d'autres fonctionnalités selon les besoins des étudiants.

Nous planifions de poursuivre nos travaux sur l'application pour parvenir à mettre au point ces ajouts et ainsi donner plus d'envergure à notre projet.

En premier lieu, nous prévoyons d'ajouter les fonctionnalités :

### Chat :

Pour la communication entre les étudiants.

### Search Team :

Trouver un groupe pour la révision collective.

Et d'améliorer les fonctionnalités d'ajout des cours et des tutoriels pour les autres niveaux de CS.