



Komeet

2021

Application recherche de salle de réunion

Crée par : Antoine RIGOT - Joris THIVARD

Komeet

Table des matières

I. Contexte	03
II. Mission	04
III. Environnement de développement	05
IV. Architecture de la solution applicative	06
V. Périmètre fonctionnel	07
VI. Données	08
VII. Interface utilisateur	10
VIII. Maquettage & Prototypage	13
IX. Développement / Codage	14
X. Tests	17
XI. Sécurité & Protection	18
XII. Documentation	19
XIII. Bilan	20

Environnement de développement

Page 05

Maquettage & Prototypage

Page 13

Sécurité & Protection

Page 18

Contexte
Notre documentation

Qu'est-ce que Kommet ?
Komeet est une application mobile permettant de localiser des salles de réunion.

Principe de l'application :
L'application Komeet disponible sur mobile permet aux utilisateurs de découvrir les salles de réunions qui les entourent. Les salles de réunion sont alors localisées sur une carte et plusieurs renseignements sont visibles : sa taille, sa capacité, ses équipements, son prix... Komeet met donc en relation les utilisateurs cherchant des salles pour des réunions, entretiens ou séminaires et les entités proposant des salles aux différents usages.

Mission

Présentation



Komeet

Android App



Existant :

Nous avons entièrement imaginé ce projet, nous n'avions aucun cahier des charges, ni d'existant au niveau du code. Nous avons donc développé la totalité de la solution mobile, créé notre charte graphique ainsi que cette documentation.

Période :

Le commencement du projet s'est effectué en novembre 2020 et s'est achevé durant le mois d'avril. Ce projet fut principalement effectué durant notre temps libre.

État :

La solution proposée est fonctionnelle. La connexion, la recherche de salle ainsi que l'ajout aux favoris est effective. Néanmoins elle ne reste pas achevée : vous trouverez dans le dossier documentation, une maquette de l'application dans son état finalisé.

Environnement de développement

Notre environnement

Introduction :

Notre projet est né de notre seule créativité : il en découle une absence de cahier des charges ce qui nous permet une grande liberté sur le choix des technologies à utiliser. Le maquetage et le prototypage ont été réalisés à l'aide d'Adobe XD tandis que la conception de l'application a été effectuée à l'aide d'Android Studio. Cette application a bénéficié d'une base de données intégrée SQLite. Pour finir, la gestion de projet ainsi que le versioning devaient s'effectuer avec GitHub.



Adobe XD

Maquetage / Prototypage

Adobe XD nous a permis de concevoir le maquetage ainsi que le prototypage de notre application.



Android Studio

IDE

Cet IDE nous a permis de développer l'application mobile en Java et la BDD embarquée SQLite.



VS Code

IDE

Cet IDE nous a permis de développer le WebService (API) ainsi que la base de données (SQL).



PostMan

Outil développement API

PostMan nous a permis d'effectuer des tests fonctionnels sur les différents usages de notre API.



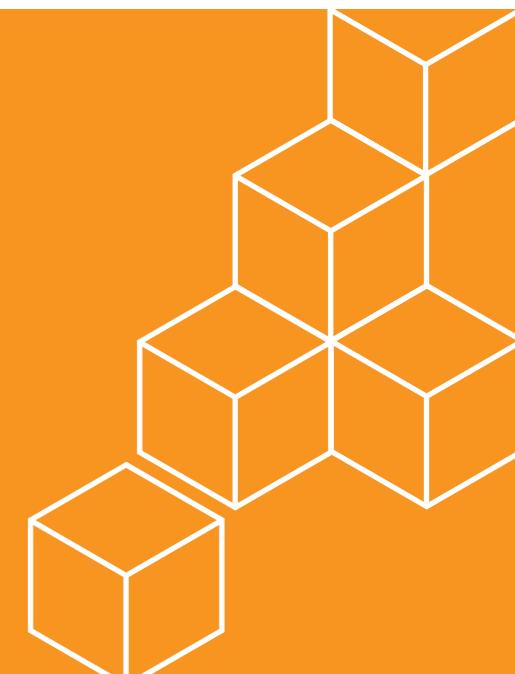
Github

Gestion de projet

GitHub nous a permis de visualiser l'avancement du projet et de mettre en place le travail collaboratif.

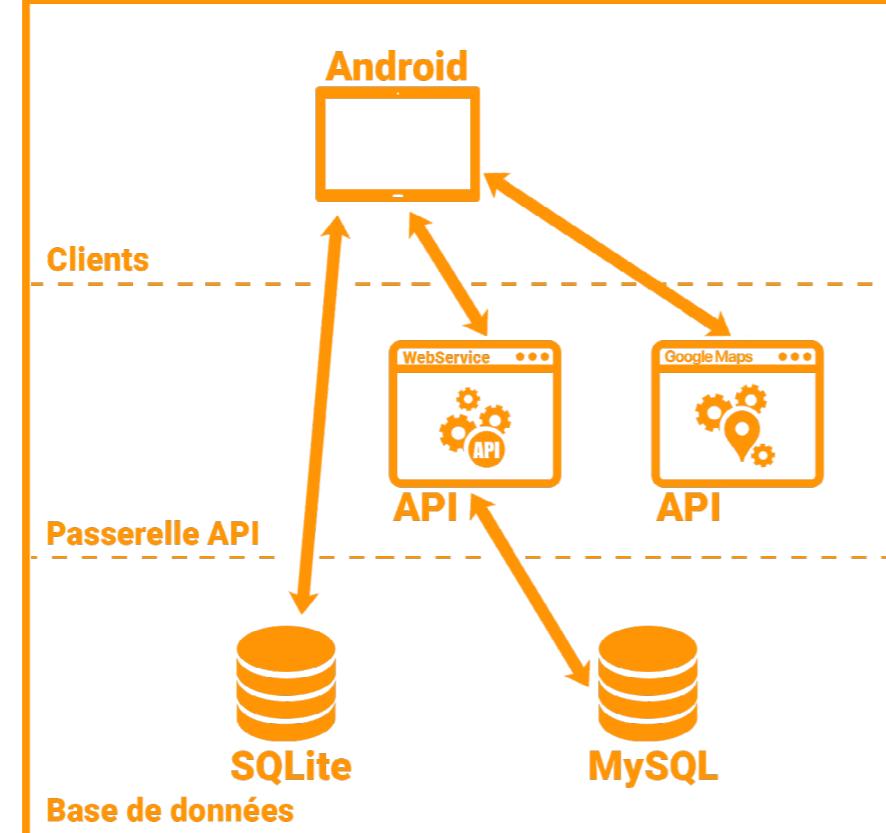
Conventions

Afin de permettre la bonne compréhension du code à l'ensemble des collaborateurs, des conventions ont été mises en place. Les fonctions ont été formulées en anglais et le "Camel case" a été la règle de nommage utilisée. Une indentation classique a été effectuée et la quasi intégralité du code a été commenté.



Architecture de la solution applicative

Notre architecture



Périmètre fonctionnel

Diagrammes cas d'utilisation

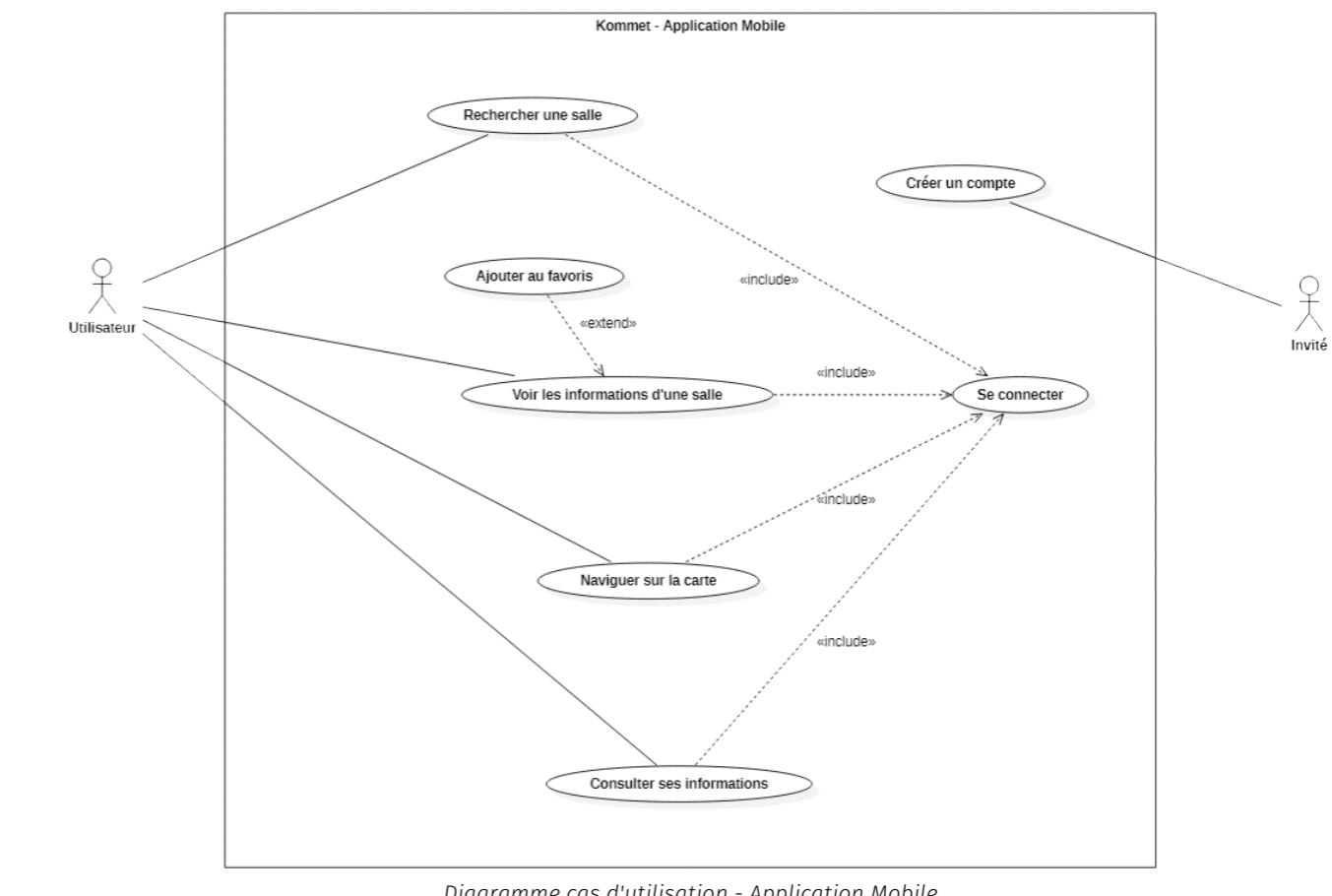
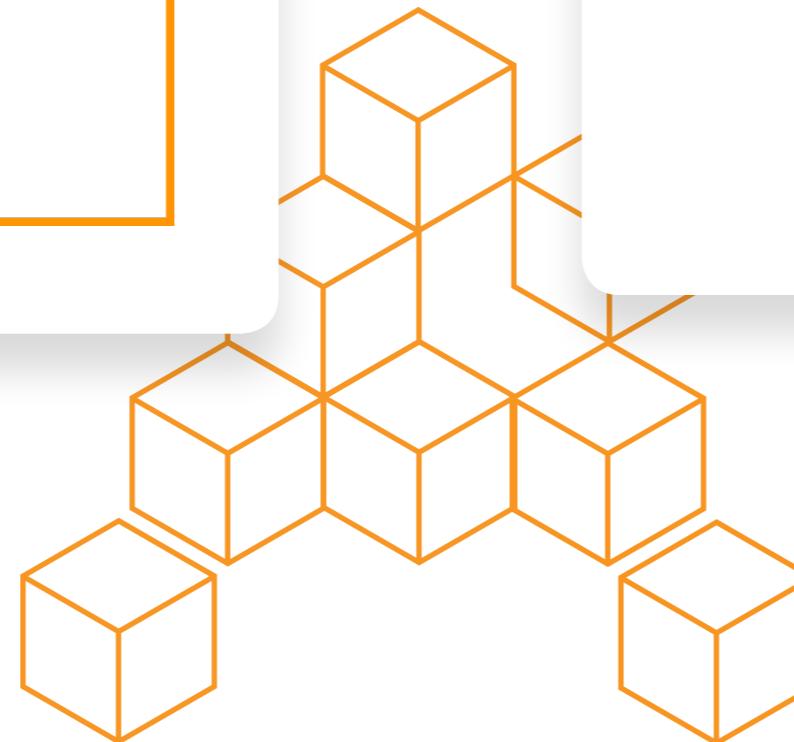
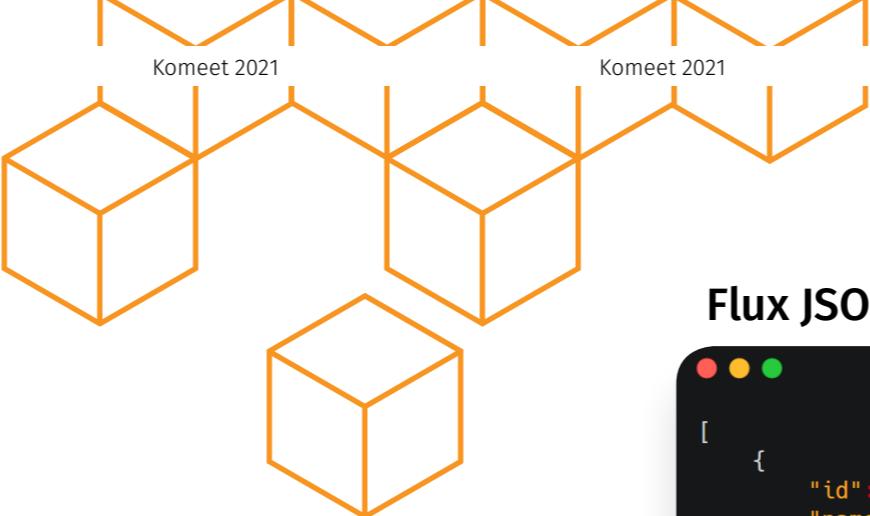


Diagramme cas d'utilisation - Application Mobile

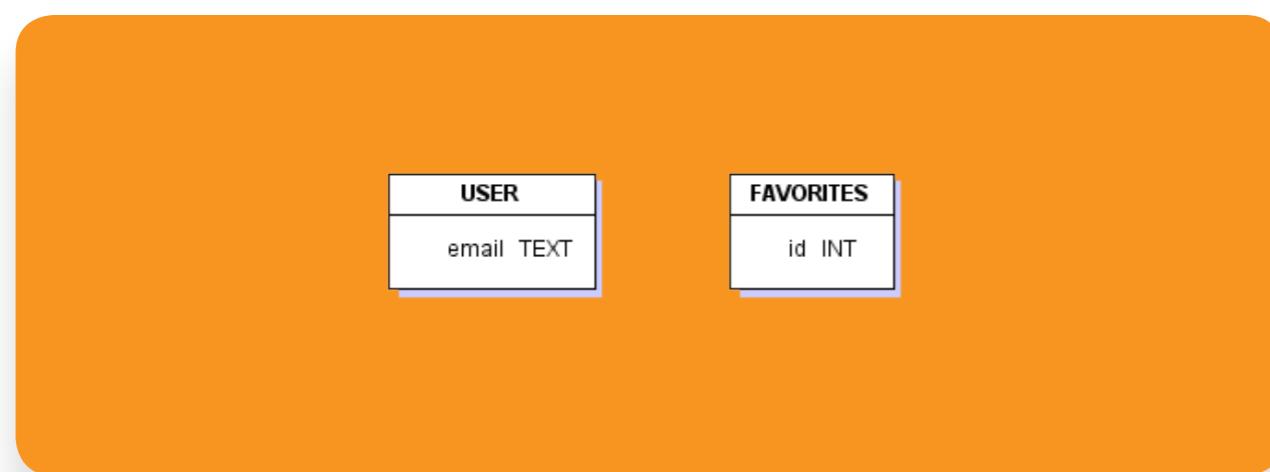
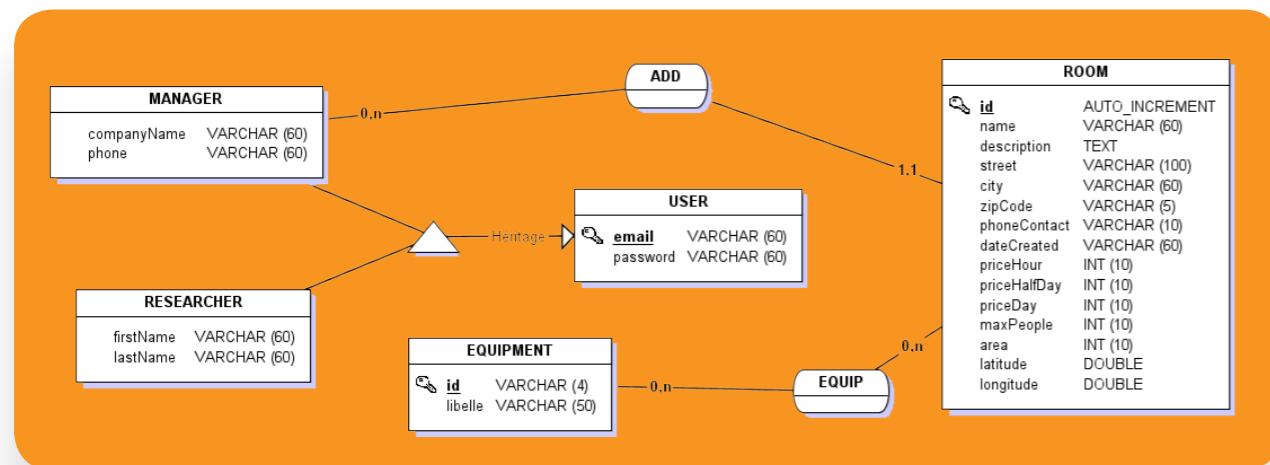




Données

Nos formats de données

MCD: Modèle Conceptuel des Données



Explication MCD SQLite

La table "USER" permet de garder en mémoire l'adresse email de l'utilisateur afin de permettre à celui-ci de rester identifié lors de ses futures connexions. La table "FAVORITES" permet de garder en mémoire les ids des salles que l'utilisateur a ajoutés à ses favoris pour les lui restituer lors de ses prochaines connexions.

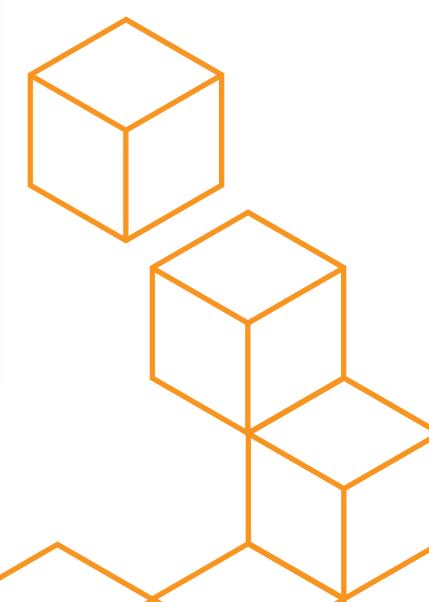
Flux JSON: Flux retourné

```

[{"id": "1", "name": "H\u00f4tel Campanile Lyon", "street": "17 Place Carnot", "city": "Lyon", "zipCode": "69002", "latitude": "45.750021038407596", "longitude": "4.828527539306008", "emailManager": "manager@gmail.com", "phoneContact": "0478655232", "description": "Nous mettons \u00e0 la disposition...", "priceHour": "70", "priceHalfDay": "200", "priceDay": "350", "maxPeople": "45", "area": "70", "dateCreated": "2021-02-25", "companyName": "CAMPANILE", "0": ["Climatisation", "Prise RJ45", "WIFI", "Fontaine \u00e0 eau"]}, {"id": "2", "name": "H\u00f4tel Sofitel Lyon Bellecour", "street": "20 Quai du Dr Gailleton", "city": "Lyon", "zipCode": "69002", "latitude": "45.75476230411873", "longitude": "4.834041205442844", "emailManager": "manager@gmail.com", "phoneContact": "0472669074", "description": "La salle est situ\u00e9e au 1er...", "priceHour": "120", "priceHalfDay": "320", "priceDay": "500", "maxPeople": "70", "area": "110", "dateCreated": "2021-03-01", "companyName": "SOFITEL", "0": ["Paper board", "WIFI", "T\u00e9l\u00e9viseur", "Vid\u00e9oprojecteur"]}], [...]

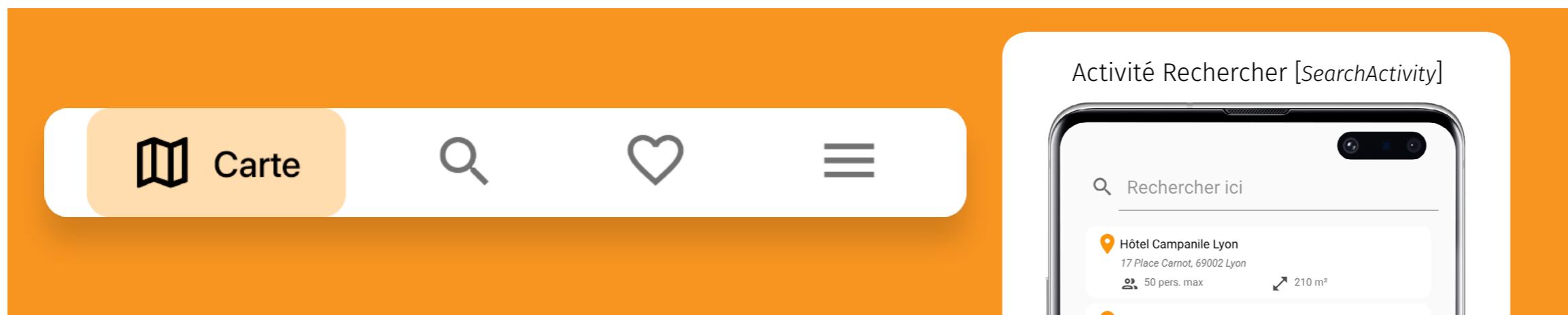
```

Flux issu de Postman - <http://joristhivard.fr/komeet/getallroom.php>



Interface utilisateur

Application Web



La Navigation utilisateur

Nous voulions une navigation standard, sobre et simple de compréhension pour tous les utilisateurs.

Notre menu se décompose en 4 parties:

Carte

L'activité « Carte » situe les différentes salles de réunion sur la map Google. Lors du clic sur une salle, la fiche descriptive de celle-ci apparaît.

Recherche

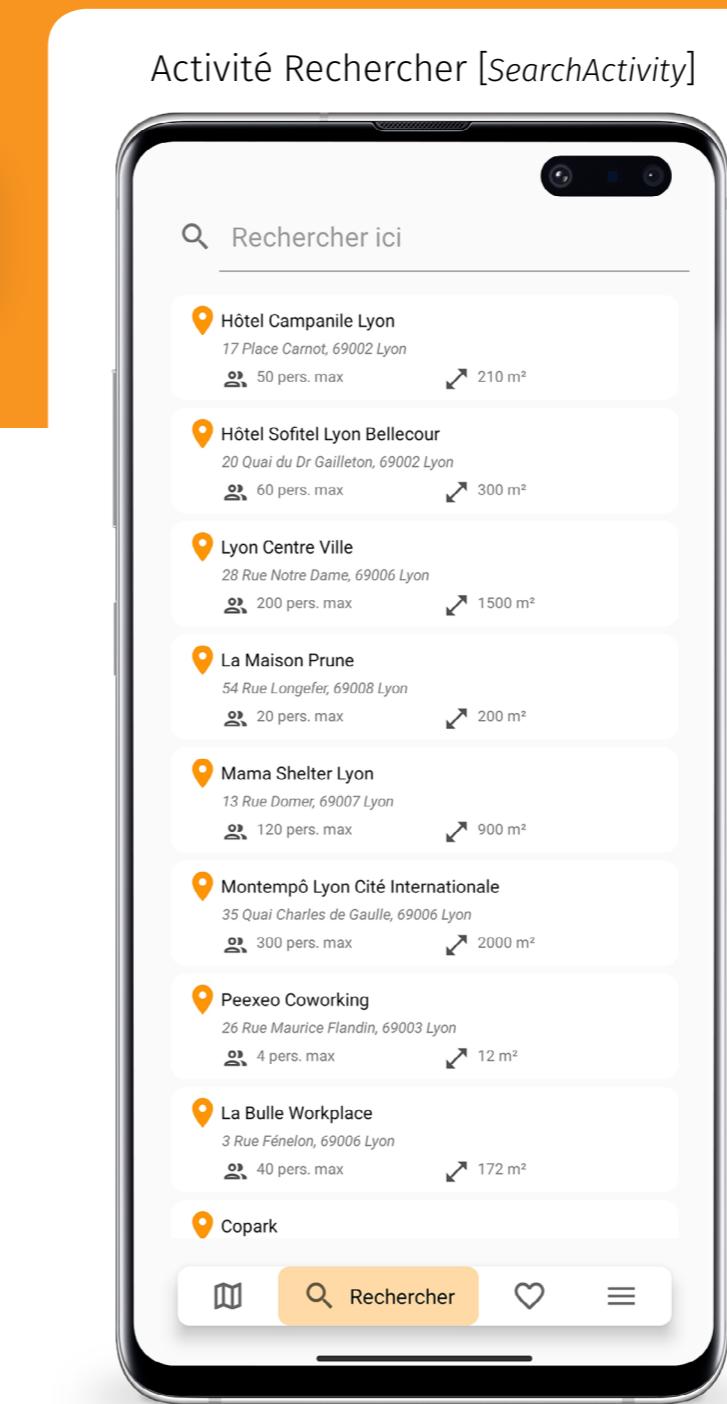
L'activité « Recherche » permet à l'utilisateur de rechercher une salle de réunion à l'aide de son nom. L'utilisateur peut aussi consulter la totalité des salles sous forme d'une liste.

Favoris

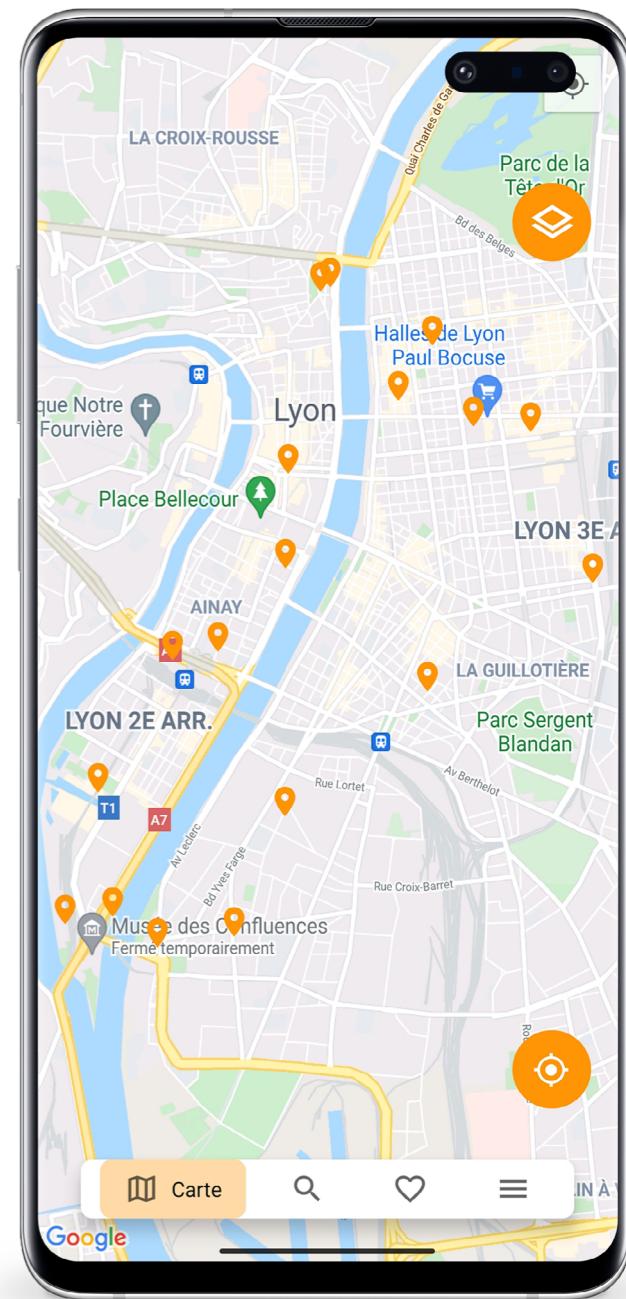
L'activité « Favoris » affiche les différentes salles que l'utilisateur aura ajoutées à ses favoris.

Menu

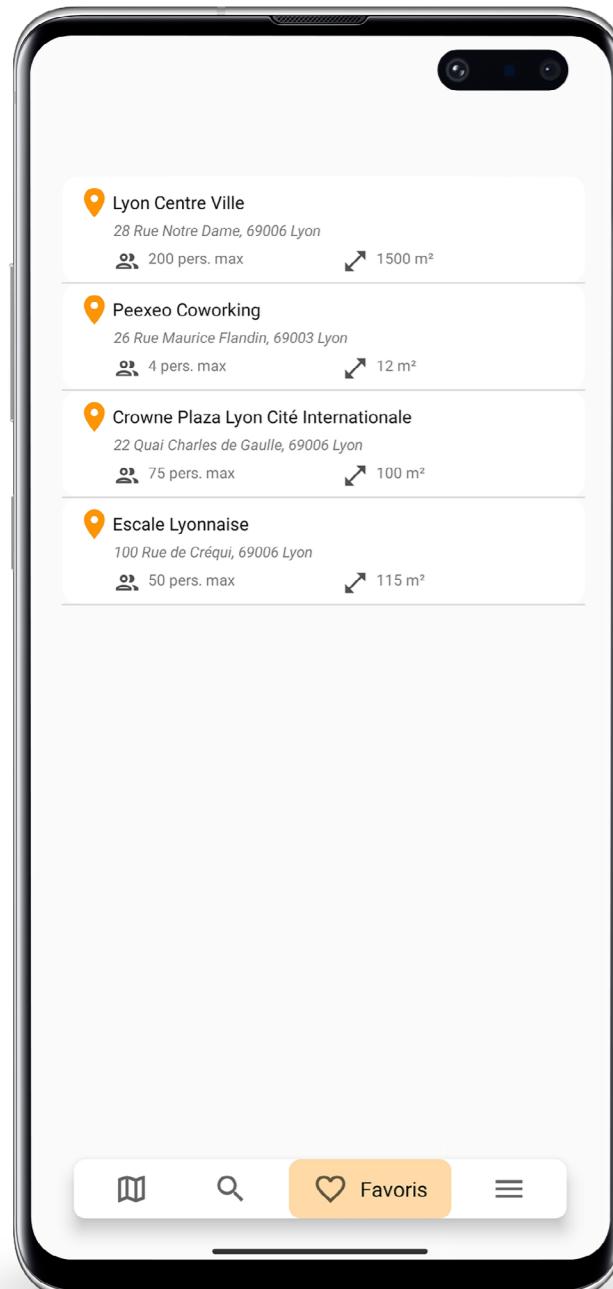
L'activité « Menu » permet les différentes actions de gestion de l'application. La déconnexion y est possible ainsi que de visualiser les différentes informations utilisateurs.



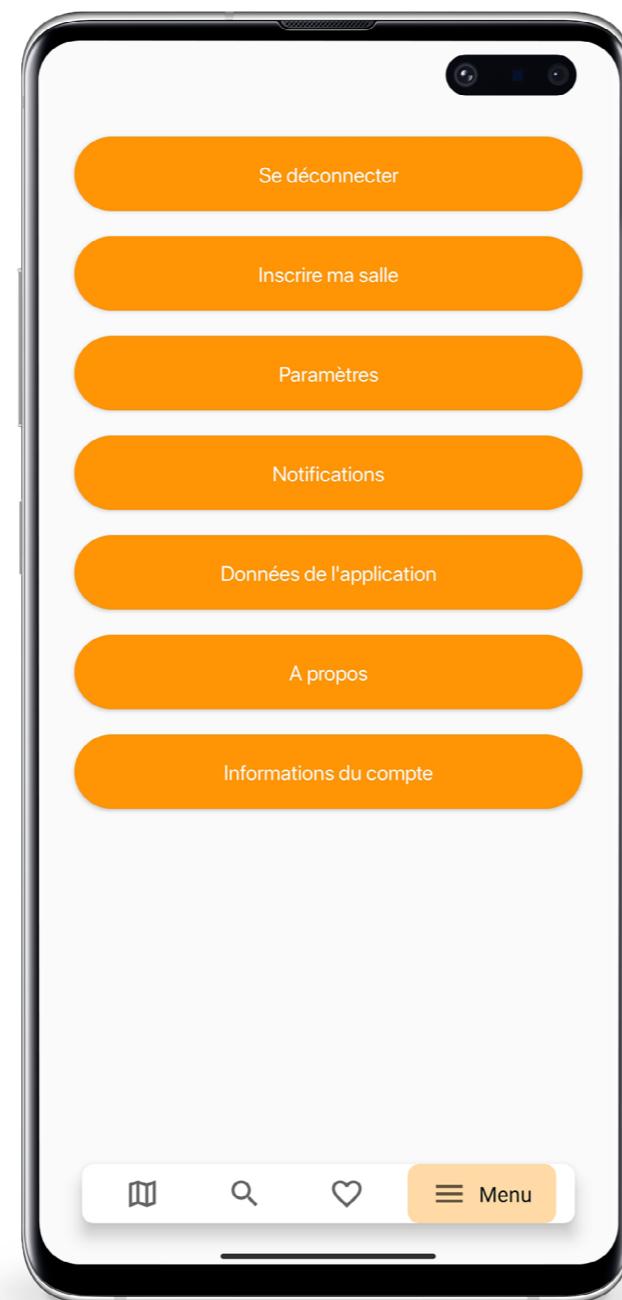
Activité Carte [MapsActivity]



Activité Favoris [FavoriteActivity]



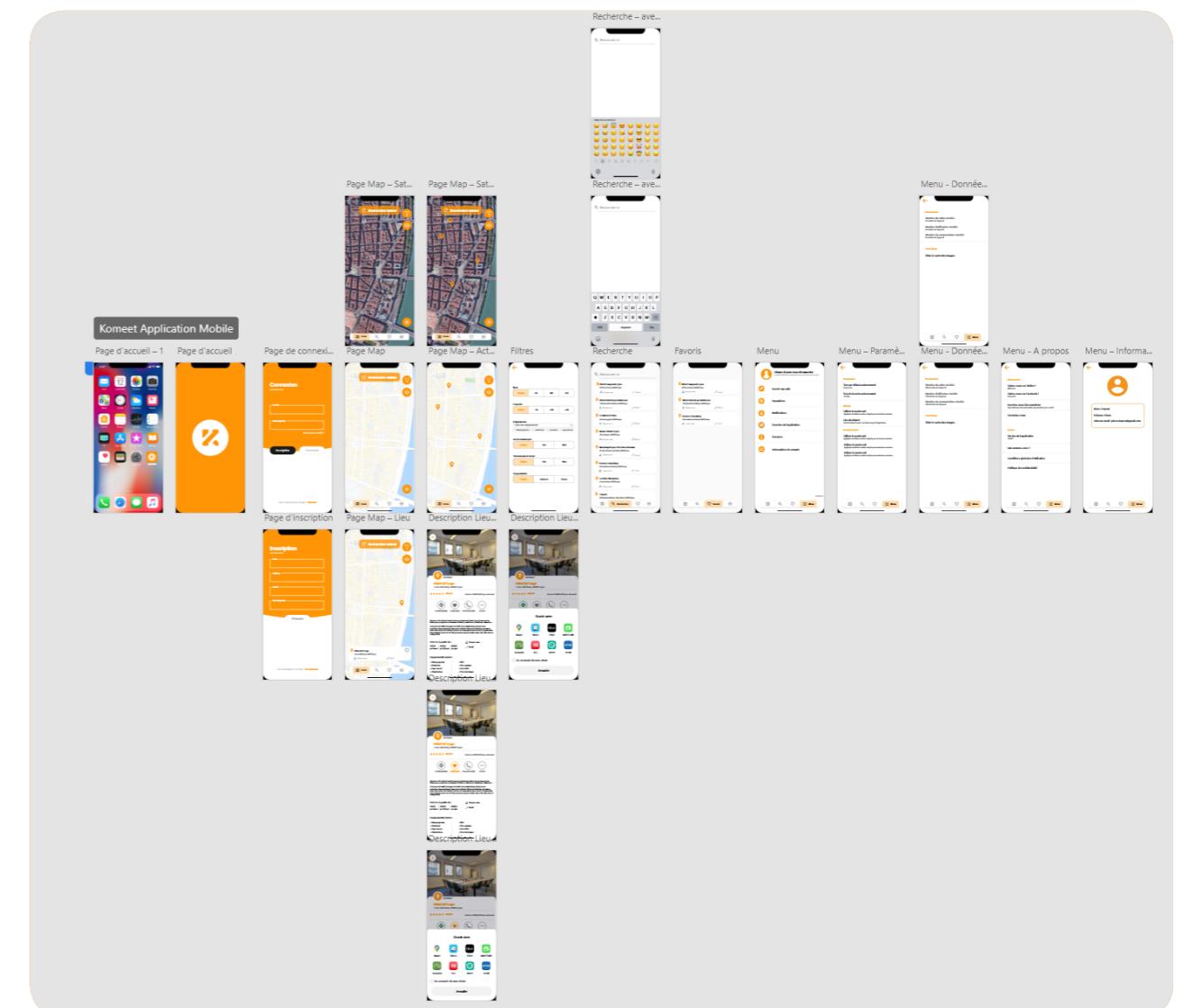
Activité Menu [MenuActivity]



Maquettage & Prototypage Adobe XD

Pourquoi le maquettage & prototypage ?

Afin d'avoir une vue d'ensemble du projet, de mettre en collaboration nos idées, de pouvoir effectuer correctement nos choix techniques et de voir le comportement de l'application. Nous avons choisi de réaliser une maquette prototype.



Pour consulter le maquettage suivre ce lien :

<https://xd.adobe.com/view/3356cb5e-9d26-4d50-a24e-0f864676f8c5-6605/?fullscreen>

Développement / Codage

Les choix de développement

Organisation des fichiers

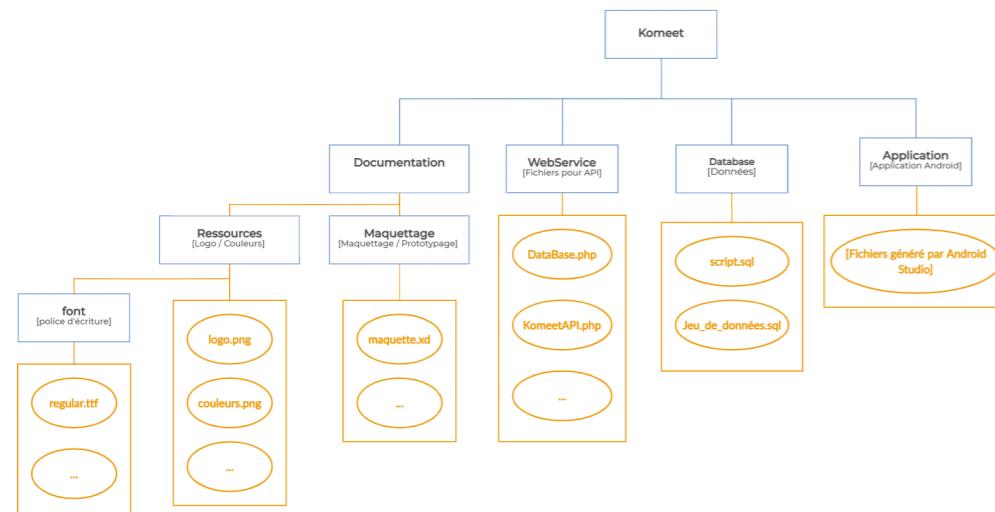


Diagramme d'arborescence des fichiers du projet

Les différents fichiers Android :

EquipmentGridAdapter : Fichier permettant d'intégrer les équipements dans la GridView de InfoRoomActivity

RoomListAdapter : Fichier permettant d'intégrer les salles dans la ListView de SearchActivity & FavoriteActivity

KomeetDAO : Fichier regroupant les fonctions permettant de communiquer avec la base de données embarquée

KomeetHelper : Fichier permettant la création de la base de données embarquée

Room : Objet regroupant toutes les caractéristiques d'une salle

RoomManager : Fichier permettant de gérer toutes les salles

FavoriteActivity : Activité dédiée à l'affichage des salles favorites de l'utilisateur

InfoAccountActivity : Activité dédiée à l'affichage des informations d'un utilisateur

InfoRoomActivity : Activité dédiée à l'affichage des informations d'une salle

LoginActivity : Activité dédiée à la connexion d'un utilisateur

MapsActivity : Activité dédiée à l'affichage des salles sur la carte

MenuActivity : Activité dédiée à l'affichage du menu

RoomInfoFragment : Fichier généré automatiquement par Android

SearchActivity : Activité dédiée à la recherche d'une salle

SignupActivity : Activité dédiée à l'inscription d'un utilisateur

SplashActivity : Activité dédiée au chargement des données

SplashPopupActivity : Popup dédiée au lancement lorsque l'utilisateur ne dispose pas d'une connexion internet

Les différents fichiers du WebService (API) :

DataBase.php : Fichier de configuration permettant l'accès à la base de données

getallroom.php : Récupère les salles et les transmets à l'application mobile

KomeetAPI.php : Fichier regroupant les méthodes servant au web service

login.php : Récupère les informations de l'utilisateur et les transmets à l'application mobile

signup.php : Inscrit l'utilisateur dans la base de données distante MySQL

verifypassword.php : Fichier vérifiant la conformité du mot de passe

verifyaccount.php : Fichier vérifiant la conformité de l'adresse email

Fichiers des différentes activités

Explications des choix techniques

Google Map Platform :

Pour la création de notre application, nous devions bénéficier d'une carte afin de pouvoir naviguer et positionner les différentes salles de réunions dessus. Nous avions donc dû faire un choix entre plusieurs API proposant des services similaires. MapBox, Fencer, Mapillary : de nombreux choix était donc possible.

Leader dans le domaine, propriétaire de son OS et disposant d'une documentation détaillée et de nombreuses ressources, c'est naturellement que notre choix c'est arrêté sur [Google Map Platform](#).

```

<resources>
    <string name="google_maps_key" templateMergeStrategy="preserve"
    translatable="false">XXXXXXXXXX</string>
</resources>
  
```

google_maps_api.xml (Ligne 1)

```

<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent" />
  
```

activity_maps.xml (Ligne 9)

La partie flouté correspond à notre clé personnelle permettant l'identification de l'application par Google.

Création d'un fragment dans l'activité maps supportant l'affichage de la carte Google.

Menu de navigation :

Pour notre menu de navigation, nous avons cherché à créer un menu intuitif et créatif. Pour cela, nous avons fait le choix d'utiliser un composant externe open source disponible sur [Github](#) appelé : Chip navigation. Celui-ci nous a permis de simplifier la création du menu tout en ayant une barre de navigation plaisant et qui aidait à garder une bonne expérience

```
dependencies {
    implementation 'com.ismaeldivita.chipnavigation:chip-navigation-bar:1.3.4'
    implementation 'org.jetbrains.kotlin:kotlin-stdlib:1.4.32'
}
```

build.gradle (Ligne 46)

Implémentation du menu ChipNavigation dans le build.gradle.

```
<com.ismaeldivita.chipnavigation.ChipNavigationBar
    android:id="@+id/chipnav"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cnb_menuResource="@menu/bottom_nav_menu"
    android:layout_alignParentBottom="true"
    android:background="@drawable/round_bottom_nav"
    app:cnb_unselectedColor="@color/black"
    app:cnb_radius="8dp"
    app:cnb_orientationMode="horizontal"
    android:layout_margin="30dp"
    android:elevation="8dp"/>
```

activity_maps.xml (Ligne 65)

Création de la barre de navigation (ChipNavigation) dans l'activité maps. Il prend comme ressource le fichier bottom_nav_menu.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <item
        android:id="@+id/ic_map"
        android:title="Carte"
        android:icon="@drawable/ic_map"
        app:cnb_iconColor="@color/black"
        app:cnb_backgroundColor="@color/orangeopacity"
        app:cnb_textColor="@color/black"/>

    <item
        android:id="@+id/ic_search"
        android:title="Rechercher"
        android:icon="@drawable/ic_search"
        app:cnb_iconColor="@color/black"
        app:cnb_backgroundColor="@color/orangeopacity"
        app:cnb_textColor="@color/black"/>

    <item
        .../>

    <item
        .../>

</menu>
```

bottom_nav_menu.xml (Ligne 2)

Fichier de ressource pour la création de la barre de navigation déclarant alors le nombre d'item ainsi que le nom, l'icone et la couleur de chacun.

Nous avions 4 items (Carte, Rechercher, Favoris, Menu) correspondant aux 4 sections de notre menu.

Tests

Tests fonctionnels, débogage & API

Tests fonctionnels

Afin de veiller au bon déroulement de notre solution, nous avons pris la décision d'effectuer des tests dit "fonctionnels" tout au long du développement de l'application. À la fin de chaque nouvelle fonctionnalité, nous procédons à différents tests afin de nous assurer du bon agissement des nouveaux ajouts mais aussi à conserver l'intégrité des plus anciens.

Débogage Android

Lors de la conception de l'application mobile via Android Studio (cf. *Environnement de développement*), nous procédons au débogage grâce à la console et au « *Logcat*¹ ». Lorsque l'application ne fonctionnait pas comme prévu initialement, ses outils nous permettaient de trouver l'erreur et de la corriger.

```
2021-05-17 17:43:27.842 9263-9263/fr.btssio.komeet E/AndroidRuntime: FATAL EXCEPTION: main
Process: fr.btssio.komeet, PID: 9263
java.lang.RuntimeException: Unable to start activity ComponentInfo{fr.btssio.komeet/fr.btssio.komeet.MainActivity}:
java.lang.NullPointerException: Attempt to invoke virtual method 'void android.widget.Button.setOnClickListener(android.view.View$OnClickListener)' on a null object reference
    at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:3449)
    at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:3601)
    at android.app.servertransaction.LaunchActivityItem.execute(LaunchActivityItem.java:85)
    at android.app.servertransaction.TransactionExecutor.executeCallbacks(TransactionExecutor.java:135)
    at android.app.servertransaction.TransactionExecutor.execute(TransactionExecutor.java:95)
    at android.app.ActivityThread$H.handleMessage(ActivityThread.java:2066)
    at android.os.Handler.dispatchMessage(Handler.java:106)
    at android.os.Looper.loop(Looper.java:223)
    at android.app.ActivityThread.main(ActivityThread.java:7656)<1 internal call>
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:592)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:947)
Caused by: java.lang.NullPointerException: Attempt to invoke virtual method 'void android.widget.Button.setOnClickListener(android.view.View$OnClickListener)' on a null object reference
    at fr.btssio.komeet.MainActivity.onCreate(MenuActivity.java:50)
    at android.app.Activity.performCreate(Activity.java:8000)
    at android.app.Activity.performCreate(Activity.java:7984)
    at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1309)<8 more ...>
```

LogCat indique une erreur ligne 50 de la classe MainActivity

WebService - API

Pour la conception de notre API, Postman nous permettait de tester notre WebService en visualisant les données communiquer par l'API. Nous devions donc veiller à ce qu'elles soient conformes à nos attentes.

¹ Logcat : outil sous forme de ligne de commande permettant de visualiser grâce à des filtres les erreurs

Sécurité & Protection

Vérification & RGPD

Vérification de sécurité

Tout au long de l'utilisation de la solution, plusieurs vérifications sont effectuées.

```
● ● ●
if(strlen($email) <= 60 && strlen($password) <= 255 && strlen($firstName) <= 60 && strlen($lastName) <= 60){
    signUp($firstName,$lastName,$email,$password); //Inscription du nouvel utilisateur
}
```

signup.php (Ligne 10)

Dans cette partie, on vérifie que la taille de l'email n'excède pas 60, que le mot de passe n'excède pas 255 et que le nom et prénom ne dépasse pas 60 caractères.

```
● ● ●
$request = "SELECT * FROM researcher WHERE email = '".$email."'";
$stmt = $con->prepare($request);
$stmt->execute();
```

KomeetAPI.php (Ligne 40)

Des requêtes préparées sont utilisées afin d'éviter les injections SQL.

Règlement Général sur la Protection des Données

Collectant des informations personnelles sur les utilisateurs, il nous était imposé de suivre les recommandations de la « CNIL¹ ». Nous avons donc réalisé le « *registre des activités de traitement*² » prévu par l'article 30 du RGPD.

Son élaboration permet au délégué à la protection des données ou au référent sur ces questions au sein d'un organisme :

- de recenser les traitements de données personnelles mis en œuvre sous la responsabilité de l'organisme
- d'évaluer les risques
- de rassembler les renseignements nécessaires à l'information des personnes identifiées dans les données de l'organisme

Consulter le registre des activités de traitement en [cliquant ici](#)

Documentation

Charte graphique



Logo :



Couleurs :



Font :

SF Pro Display

abcdefghijklmnopqrstuvwxyz
 ABCDEFGHIJKLMNOPQRSTUVWXYZ



Bilan

Retours, acquis & amélioration

Acquisition de nouvelles compétences



Acquisition de nouvelles compétences

Techniques

- » WebService (API)
- » Java (Application Mobile)
- » Maquettage & Prototypage



Acquisition de nouvelles compétences

Organisationnelles

- » Travail collaboratif (Github)
- » Choix de solutions
- » Recherche d'informations

Axes d'amélioration éventuels

Malgré le bon déroulement de notre projet, des axes d'amélioration auraient pu être perfectionnés.

Grâce à une plus grande réflexion en amont, l'organisation et l'optimisation du code auraient peu été corrigé (utilisation de Fragment à la place d'Activity).

Remerciement

Nous tenons à remercier l'ensemble de l'équipe pédagogique et plus précisément nos professeurs de spécialité qui nous ont apporté une aide concrète pour l'élaboration de ce projet.

Merci !



Komeet

2021

BTS SIO opt. SLAM

Antoine RIGOT

Joris THIVARD

