

#### Contexte:

Cooking for all est une application créée dans le cadre d'un module d'Android de deuxième année de DUT Informatique, à l'IUT Informatique de Clermont-Ferrand par Philippe ROUGERIE et Thibault Lafaurie.

Cette application a pour but de permettre à l'utilisateur de créer des recettes puis de les regarder.

L'application nécessite une connexion WIFI ou réseau permanente pour fonctionner (Chargement de potos depuis internet, envoi de données sur le cloud...)

Pour pouvoir utiliser l'application il doit se connecter à la FireBase ou alors créer un compte s'il n'en a pas.

Une fois connecté, les données stockées dans la FireBase sont récupérées.

Il peut alors créer une recette. Il aura à saisir d'abord le nom, ensuite au moins un ingrédient, puis le descriptif des actions à faire pour effectuer la recette et finalement des photos pour accompagner celle-ci.

L'utilisateur a le choix entre prendre de nouvelles photos, importer des photos depuis son espace de stockage, ou les deux.

Quand il enregistre sa recette celle-ci est à la fois enregistrée en local et sur la Firebase. Pour les photos seul les liens pour accéder à l'image stockée dans le Storage de la FireBase sont stockées dans le Firestore.

L'utilisateur peut vouloir regarder l'ensemble de ses recettes. Il obtiendra une liste de ses recettes avec leur nom et en appuyant sur une recette une activité affichant la recette s'ouvre.

Sur cette activité il peut voir la recette. Les photos s'affichent une à une et peuvent être slidées.

**AddPhotoAndActionsActivity**  
(from activity)

```
-Ingredients: ListView
-IstPhotos: Bitmap[] (collection="List")
-mStorageRef: StorageReference
+KEY_LIST_String = "2222" (readOnly)
+PHOTO_CODE_int = 1046 (readOnly)
+REQUEST_IMAGE_CAPTURE_int = 1 (readOnly)
-fUrl: String =
-mAuth: FirebaseAuth
-currentUser: FirebaseUser
-appPermission: String[]
-name: String
-db: DatabaseHelper
-idRecipe: long
+PERMISSIONS_REQUEST_CODE_int = 1240 (readOnly)
-savedInstance: Bundle
-total: int

#onCreate(savedInstanceState: Bundle): void
-initAct(savedInstanceState: Bundle): void
+cameraOn(view: View): void
-mainAddPhoto(imageBitmap: Bitmap): void
+checkAndRequestPermissions(): boolean
+onRequestPermissionsResult(requestCode: int, permissions: String, grantResults: int[]): void
+showDialog(title: String, msg: String, positiveLabel: String, positiveOnClick: OnClickListerner, negativeLabel: String, negativeOnClick: OnClickListerner, isCancelable: boolean): AlertDialog
+compressBitmap(image: Bitmap): void
#onActivityResult(requestCode: int, resultCode: int, data: Intent): void
+searchPhoto(view: View): void
#onListItemClick(): ListView, v: View, position: int, id: long): void
+saveRecipe(view: View): void
+uploadNewRecipe(): void
```

**Ingredient**  
(from Module)

```
-name: String
-quantity: int
-unit: String
+CREATOR_Creator (readOnly)

+constructor+=Ingredient(name: String, quantity: int, unit: String)
+constructor+=Ingredient(in: Parcel)
+getName(): String
+getQuantity(): int
+getUnit(): String
+describeContents(): int
+writeToParcel(dest: Parcel, flags: int): void
+toString(): String
+setName(name: String): void
+setQuantity(quantity: int): void
+setUnit(unit: String): void
```

**AddingredientsActivity**  
(from activity)

```
-textView: TextView
+EXTRA_MESSAGE_NAME_String = "@string/name" (readOnly)
+EXTRA_MESSAGE_LIST_String = "liste" (readOnly)
+KEY_LIST_String = "1111" (readOnly)
-Ingredients: ListView

#onCreate(savedInstanceState: Bundle): void
#onSaveInstanceState(outState: Bundle): void
#onRestoreInstanceState(savedInstanceState: Bundle): void
+addingredient(view: View): void
+onItemSelected(parent: AdapterView, view: View, position: int, id: long): void
+onNothingSelected(parent: AdapterView): void
+validating(view: View): void
```

**Recipe**  
(from Module)

```
-recipeName: String
-steps: String
-photos: String[] (collection="ArrayList")
+CREATOR_Creator (readOnly)

+constructor+=Recipe(recipeName: String, ingredient: Ingredient[], steps: String, photos: String[])
+constructor+=Recipe(recipeName: String, steps: String)
+toString(): String
+constructor+=Recipe(in: Parcel)
+getRecipeName(): String
+getIngredient(): Ingredient[]
+getSteps(): String
+getPhotos(): String[]
+setRecipeName(recipeName: String): void
+setIngredient(ingredient: Ingredient[]): void
+setSteps(steps: String): void
+setPhotos(photos: String[]): void
+describeContents(): int
+writeToParcel(dest: Parcel, flags: int): void
```

**InscriptionActivity**  
(from activity)

```
-emailInscription: EditText
-passwordInscription: EditText
-passwordVerif: EditText
-mAuth: FirebaseAuth

#onCreate(savedInstanceState: Bundle): void
+onStart(): void
+inscription(view: View): void
+success(): void
+validateForm(): boolean
+connection(mail: String, pass: String): void
```

**ImageAdapter**  
(from Adapter)

```
-items: Bitmap[] (collection="List")
-mAuth: FirebaseAuth

+getView(position: int, convertView: View, parent: ViewGroup): View
```

**ConnectActivity**  
(from activity)

```
-email: EditText
-password: EditText
-mAuth: FirebaseAuth
-NAME_String = "name" (readOnly)
-QUANTITES_String = "quantities" (readOnly)
-UNITS_String = "unit" (readOnly)
-PHOTOS_String = "photos" (readOnly)
-STEPS_String = "steps" (readOnly)
-INGREDIENTNAME_String = "ingredientNames" (readOnly)

#onCreate(savedInstanceState: Bundle): void
+onStart(): void
+connect(view: View): void
+success(): void
+fail(): void
+register(view: View): void
+validateForm(): boolean
```

**DatabaseHelper**  
(from Database)

```
-DATABASE_VERSION_int = 6 (readOnly)
-DATABASE_NAME_String = "myDatabase" (readOnly)
-ID_NB_PHOTO_String = "0" (readOnly)
-TABLE_RECIPE_String = "recipe" (readOnly)
-TABLE_INGREDIENT_String = "ingredient" (readOnly)
-TABLE_PHOTO_String = "photo" (readOnly)
-TABLE_PHOTO_RECIPE_String = "photoRecipe" (readOnly)
-TABLE_INGREDIENT_RECIPE_String = "ingredientRecipe" (readOnly)
-TABLE_NB_PHOTOS_String = "nbPhotos" (readOnly)
-KEY_RECIPE_String = "keyRecipe" (readOnly)
-RECIPE_NAME_String = "recipeName" (readOnly)
-STEPS_String = "steps" (readOnly)
-KEY_INGREDIENT_String = "keyIngredient" (readOnly)
-INGREDIENT_NAME_String = "ingredientName" (readOnly)
-QUANTITY_String = "quantity" (readOnly)
-UNIT_String = "unit" (readOnly)
-KEY_PHOTO_String = "keyPhoto" (readOnly)
-KEY_PHOTO_PR_String = "keyPhotoPR" (readOnly)
-KEY_RECIPE_PR_String = "keyRecipePR" (readOnly)
-KEY_INGREDIENT_IR_String = "keyIngredientIR" (readOnly)
-KEY_RECIPE_IR_String = "keyRecipeIR" (readOnly)
-NB_PHOTOS_String = "nbPhotos" (readOnly)
-TABLE_CREATE_RECIPE_String = "createRecipe" (readOnly)
-TABLE_CREATE_INGREDIENT_String = "createIngredient" (readOnly)
-TABLE_CREATE_PHOTO_String = "createPhoto" (readOnly)
-TABLE_CREATE_PHOTO_RECIPE_String = "createPhotoRecipe" (readOnly)
-TABLE_CREATE_INGREDIENT_RECIPE_String = "createIngredientRecipe" (readOnly)
-TABLE_CREATE_NB_PHOTOS_String = "createNbPhotos" (readOnly)

+getDatabaseVersion(): int
+constructor+=DatabaseHelper(context: Context)
+insertNbPhotosWithId(SQLiteDatabase): void
+onCreate(db: SQLiteDatabase): void
+onUpgrade(db: SQLiteDatabase, oldVersion: int, newVersion: int): void
+addRecipe(name: String, steps: String): long
+getRecipeId(): long
+getNbPhotos(): int
+getAllRecipes(): Recipe[]
+getCount(recipeName: String): long
+addingredient(ingredient: Ingredient): long
+addingredientRecipe(ingredient: long, idRecipe: long): void
+getIngredient(id: long): Ingredient
+updateNumberPhoto(nb: int): void
+getAllIngredientsRecipe(idRecipe: long): Ingredient[]
+addPhoto(photo: String): void
+addPhotoRecipe(photo: String, idRecipe: long): void
+getAllPhotosRecipe(idRecipe: long): String[]
```

**DisplayRecipesActivity**  
(from activity)

```
-Ingredients: ListView
-Recipe: Recipe
-viewPager: ViewPager

#onCreate(savedInstanceState: Bundle): void
```

**LookListActivity**  
(from activity)

```
-lRecipes: ListView
-IstRecipe: Recipe[] (collection="ArrayList")
+EXTRA_MESSAGE_RECIPE_String = "recipe" (readOnly)
-NB_PHOTOS_String = "nbPhotos" (readOnly)
#onCreate(savedInstanceState: Bundle): void
#onListItemClick(): ListView, v: View, position: int, id: long): void
```

**IngredientAdapter**  
(from Adapter)

```
+constructor+=IngredientAdapter(context: Context, resource: int, objects: Ingredient[])
+getView(position: int, convertView: View, parent: ViewGroup): View
```

**MainActivity**  
(from activity)

```
-LOG_TAG_TAG_String (readOnly)
-mAuth: FirebaseAuth
-connectivityManager: ConnectivityManager

#onCreate(savedInstanceState: Bundle): void
+onStart(): void
+launchCreateRecipeActivity(view: View): void
+onRecipe(view: View): void
+connect(view: View): void
```

**CreateRecipeActivity**  
(from activity)

```
-LOG_TAG_TAG_String (readOnly)
+EXTRA_MESSAGE_String = "com.example.cookingforall" (readOnly)
#onCreate(savedInstanceState: Bundle): void
+sendName(view: View): void
```

**PhotoPagerAdapter**  
(from Adapter)

```
-numItems: int
-photos: String[] (collection="List")
+constructor+=PhotoPagerAdapter(fragmentManager: FragmentManager, photos: String[])
+getCount(): int
+getItem(position: int): Fragment
+getPageTitle(position: int): CharSequence
```

**FragmentPhoto**  
(from Fragments)

```
-photo: String
+newInstance(photo: String): FragmentPhoto
#onCreate(savedInstanceState: Bundle): void
#onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View
```

**RecupImage**  
(from Threads)

```
-bmiImage: ImageView
+constructor+=RecupImage(bmiImage: ImageView)
+doInBackground(urls: String): Bitmap
+onPostExecute(result: Bitmap): void
```

**RecipeAdapter**  
(from Adapter)

```
-items: Recipe[] (collection="List")

+constructor+=RecipeAdapter(context: Context, resource: int, objects: Recipe[])
+getView(position: int, convertView: View, parent: ViewGroup): View
```

## Description de l'UML :

Ce diagramme permet de mettre en relief un premier point : il est plutôt simple et n'utilise pas de patrons de conception.

En effet notre application utilise une succession d'activités pour effectuer les diverses actions. Donc la plupart des liens entre les activités sont des dépendances.

On remarque que certaines activités contiennent des listes d'objets du model. Ces listes ont été placées en composition. Ces listes seront placées dans des listes View. C'est pour cela que certaines des activités possèdent un adapter.

La DisplayRecipesActivity possède un PhotoPagerAdapter qui possède des fragments pour afficher des photos. Comme montré sur le diagramme, le fragment utilise une Task pour récupérer les images.

Finalement le DataHelper est mis à part (il surchargerait trop le diagramme). Celui-ci est lié à la DisplayRecipesActivity et à 3 autres activity.

Application Cooking For All

