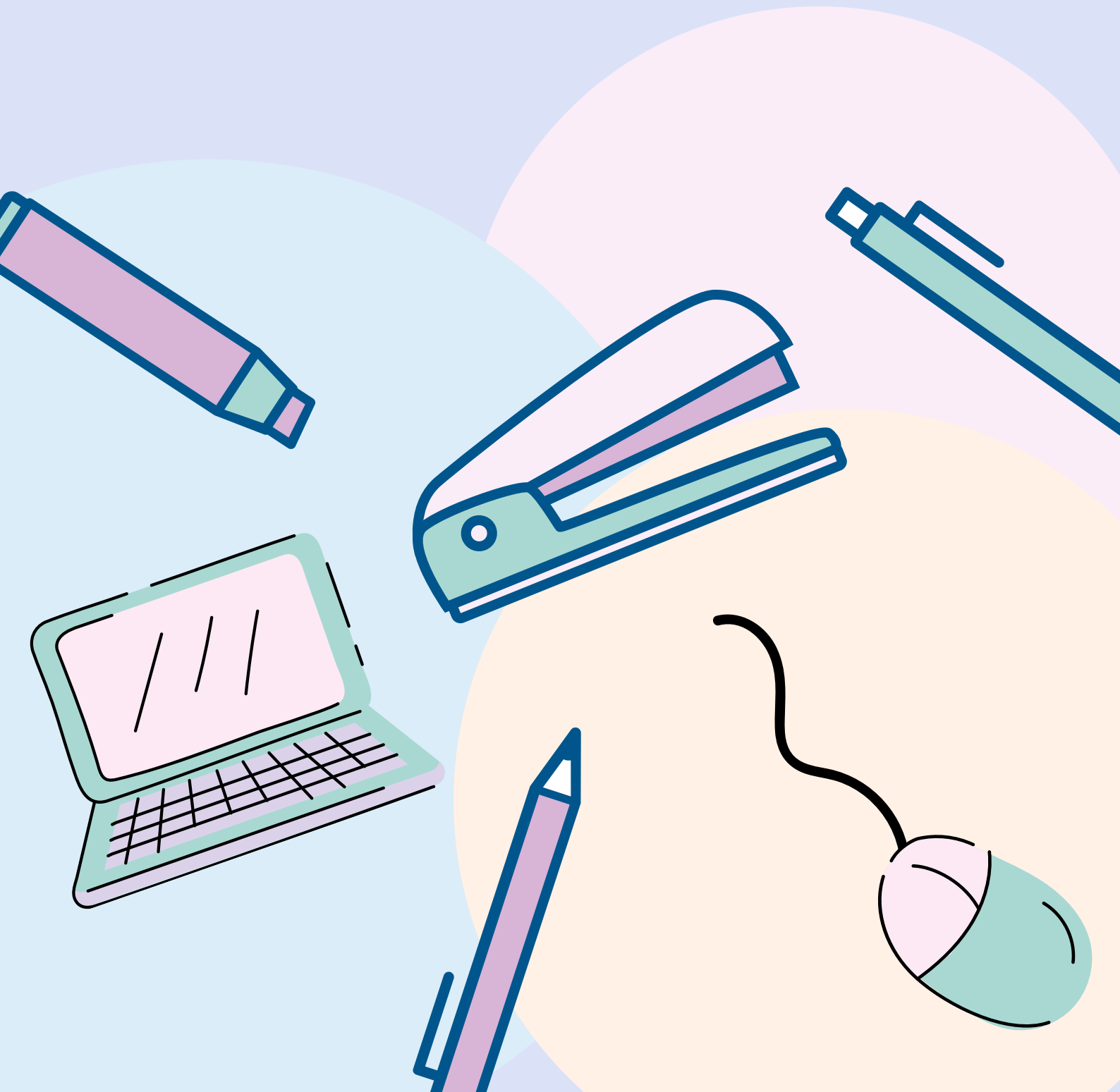
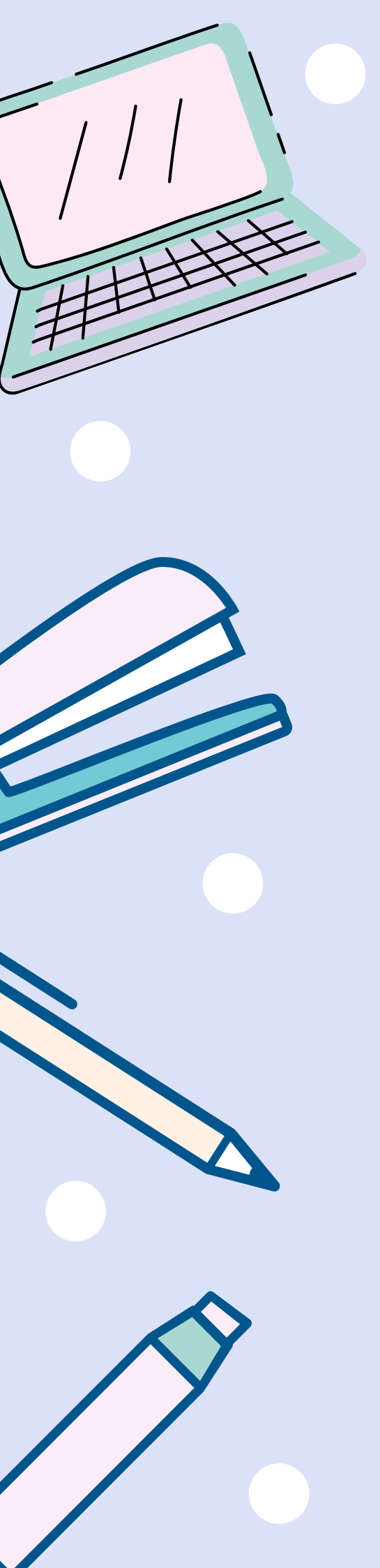


# DUTSUP

Imane ABDOU | Samy BERNOU | Farah FERGUENE  
Emmie KIEFFER | Hakim OUMAATI





# PRÉSENTATION

Introduction.....

Architecture.....

## BACK-END

Base de donnés.....

## FRONT-END

Design.....

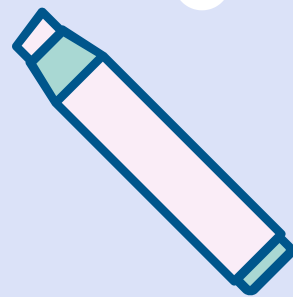
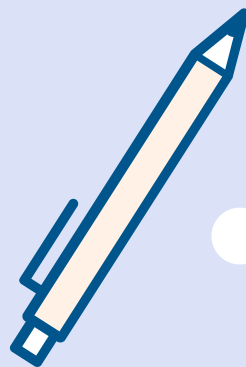
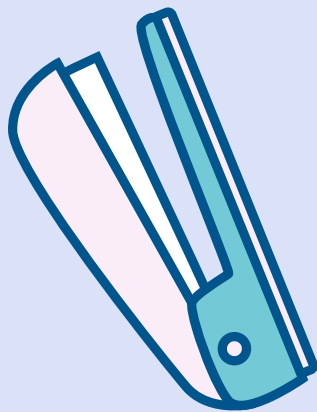
Vue d'application.....

## CONCLUSION

Difficultés.....

Améliorations possibles.....

Bilan.....

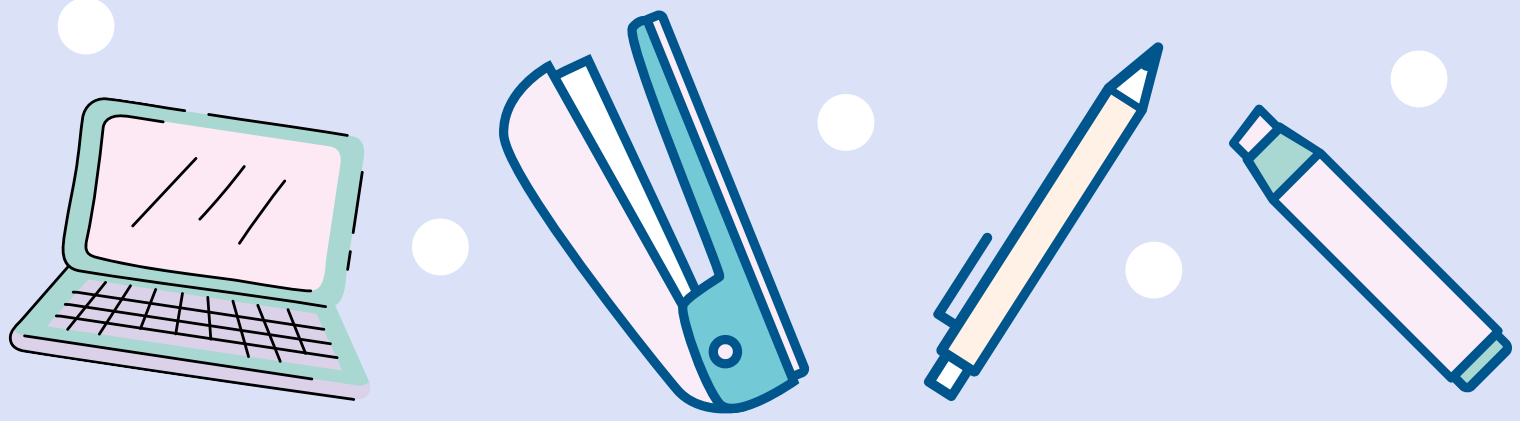


# INTRODUCTION

Dans le cadre de la réalisation de notre projet d'Application Mobile, nous avons décidé de réaliser une application de poursuite d'études qui permettrait aux étudiants en DUT informatique de trouver la formation de leur rêve.

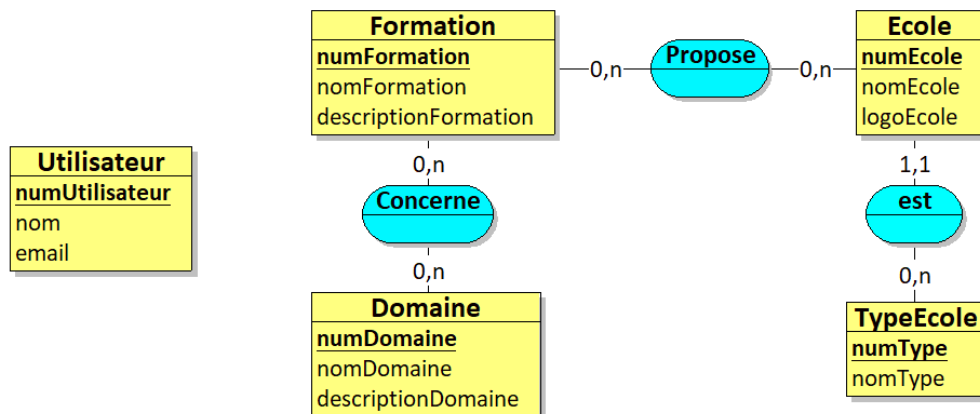
En effet, selon les centres d'intérêts de l'étudiant, notre application répertorie les écoles et les formations qui pourraient potentiellement lui plaire afin que ce dernier puisse obtenir les informations nécessaires sur celles-ci.





# BASE DE DONNÉES

Pour stocker nos données, nous avons décidé d'utiliser SQLite. En effet, c'est ce qui nous paraissait le plus simple dans l'utilisation que nous voulions en faire. Les données sont donc stockées en local chez chacun des utilisateurs de l'application.



Nous avons donc fait pour chaque table présente dans notre MCD une classe JAVA correspondante. Nous avons fait une classe DatabaseHelper qui est une extension de SQLiteOpenHelper et qui donc nous a permis de faire toutes les requêtes liées à la création, la suppression ainsi que la modification de la base. Pour chaque table, nous avons écrit les requêtes qui permettent de créer la table, récupérer un élément de la table à l'aide de son numéro, récupérer tout les éléments de la table, modifier un certain élément d'une table ainsi que supprimer un certain élément d'une table.

Voici des exemples des requêtes mise en place pour la modification des tables. Nous vous présentons celles pour la table Utilisateur, mais nous avons à peu près la même chose pour toutes les tables.

## CRÉATION

```
public void createUtilisateur(Utilisateur u){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues value = new ContentValues();
    value.put(UTILISATEUR_NOM,u.getNomUtilisateur());
    value.put(UTILISATEUR_EMAIL,u.getEmailUtilisateur());
    db.insert(TABLE_UTILISATEUR, nullColumnHack: null,value);
}
```

## RÉCUPÉRATION UN ÉLÉMENT

```
public Utilisateur getUtilisateur (int num){
    SQLiteDatabase db = this.getReadableDatabase();
    String selectQuery = "SELECT * FROM "+TABLE_UTILISATEUR+" WHERE "+UTILISATEUR_NUMUTILISATEUR+" = "+num;
    Log.e(LOG,selectQuery);
    Cursor c = db.rawQuery(selectQuery, selectionArgs: null);
    if(c!=null){c.moveToFirst();}
    Utilisateur u = new Utilisateur();
    u.setNumUtilisateur(c.getInt(c.getColumnIndex(UTILISATEUR_NUMUTILISATEUR)));
    u.setNomUtilisateur(c.getString(c.getColumnIndex(UTILISATEUR_NOM)));
    u.setEmailUtilisateur(c.getString(c.getColumnIndex(UTILISATEUR_EMAIL)));
    return u;
}
```

## RÉCUPÉRATION DE TOUT LES ÉLÉMENTS

```
public List<Utilisateur> getAllUtilisateur(){
    List<Utilisateur> utilisateurs = new ArrayList<>();
    String selectQuery = "SELECT * FROM "+TABLE_UTILISATEUR;
    Log.e(LOG,selectQuery);
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor c = db.rawQuery(selectQuery, selectionArgs: null);
    if(c.moveToFirst()){
        do {
            Utilisateur u = new Utilisateur();
            u.setNumUtilisateur(c.getInt(c.getColumnIndex(UTILISATEUR_NUMUTILISATEUR)));
            u.setNomUtilisateur(c.getString(c.getColumnIndex(UTILISATEUR_NOM)));
            u.setEmailUtilisateur(c.getString(c.getColumnIndex(UTILISATEUR_EMAIL)));
            utilisateurs.add(u);
        } while (c.moveToNext());
    }
    return utilisateurs;
}
```

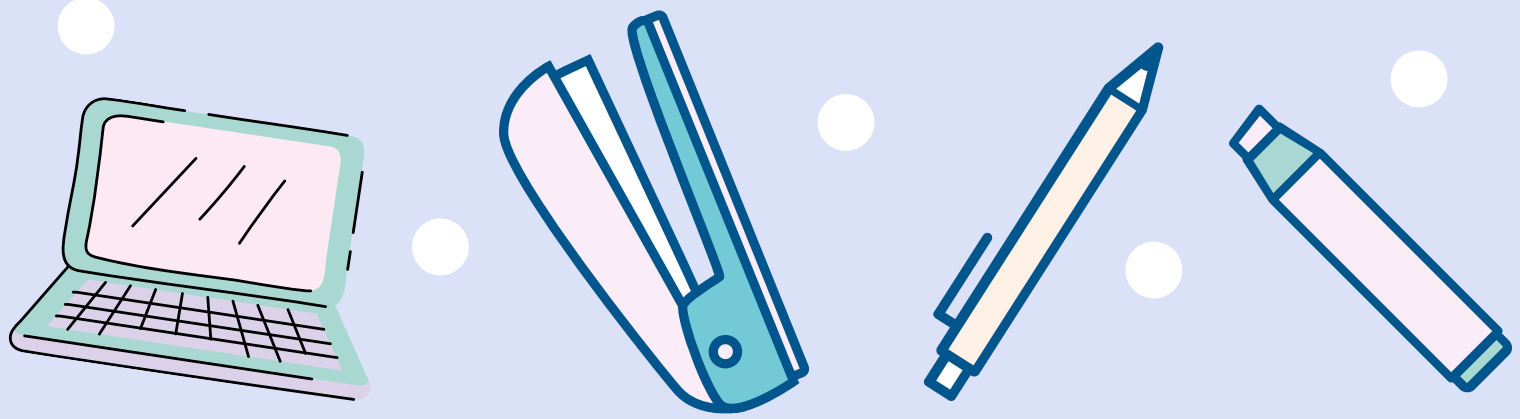
## MODIFICATION D'UN ÉLÉMENT

```
public void updateUtilisateur(Utilisateur u){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues value = new ContentValues();
    value.put(UTILISATEUR_NOM,u.getNomUtilisateur());
    value.put(UTILISATEUR_EMAIL,u.getEmailUtilisateur());
    db.update(TABLE_UTILISATEUR,value, whereClause: UTILISATEUR_NUMUTILISATEUR+" = ?",new String[]{String.valueOf(u.getNumUtilisateur())});
}
```

## SUPPRESSION D'UN ÉLÉMENT

```
public void deleteUtilisateur(int num){
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_UTILISATEUR, whereClause: UTILISATEUR_NUMUTILISATEUR+" = ?",new String[]{String.valueOf(num)});
}
```

Les deux tables qui font référence aux associations - Propose et Concerne - ont une petite particularité par rapport aux autres. En effet, elles ne possèdent pas la requête de modification.



# DESIGN

## CHOIX DU VISUEL DE L'APPLICATION

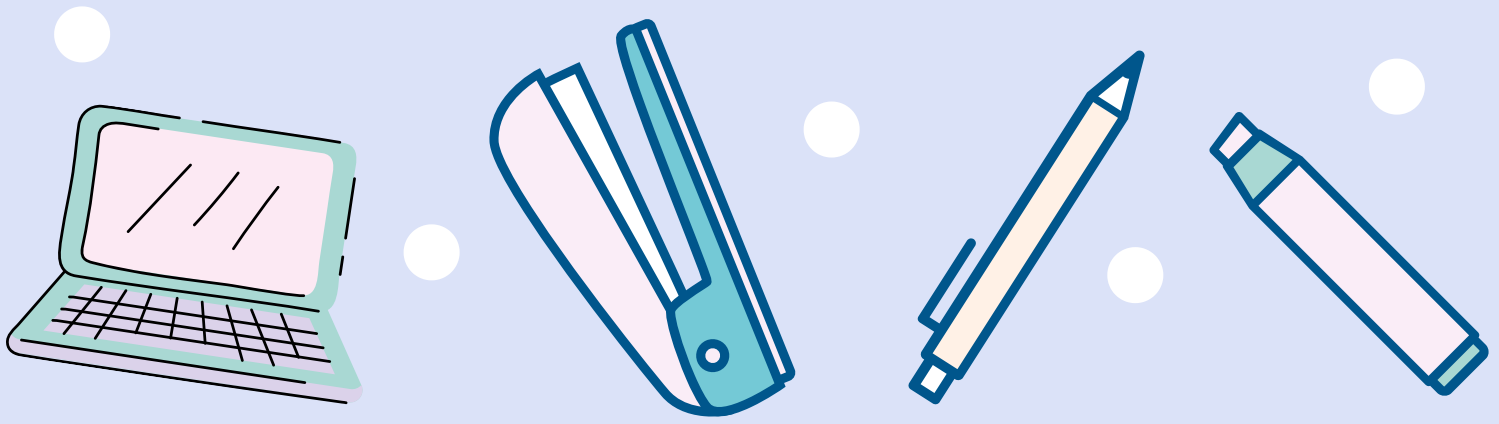
En ce qui concerne la partie visuelle, plusieurs design ont été créés et développés à l'aide de Figma.

Ce qui nous a permis de nous projeter et d'avoir une base sur laquelle nous lancer afin de développer l'application finale sur android studio.

Voici un aperçu du design sélectionné et développé sur figma:







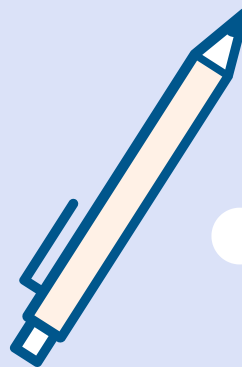
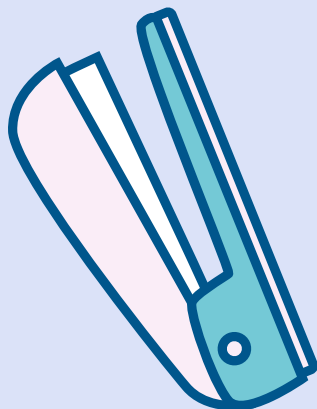
# DIFFICULTÉS

Ce projet ayant été la première application mobile que nous ayons eue à réaliser, nous avons été confrontés à de nombreuses difficultés. Toutefois, nous avons su les surmonter afin d'obtenir le rendu que voici et gagner en compétences.

En premier lieu, nous avons au départ été déstabilisés par notre manque de compétences en la matière. Bien qu'aides par nos enseignants et les ressources disponibles sur internet, la prise en main d'Android Studio fut complexe et a requis un réel effort de notre part.

Par ailleurs, le choix de la gestion de la base de données a aussi soulevé de nombreuses interrogations au sein du groupe. En effet, nous ne savions pas pour quelle option opter et avons finalement décidé d'utiliser SQLite.





# AMÉLIORATIONS POSSIBLES

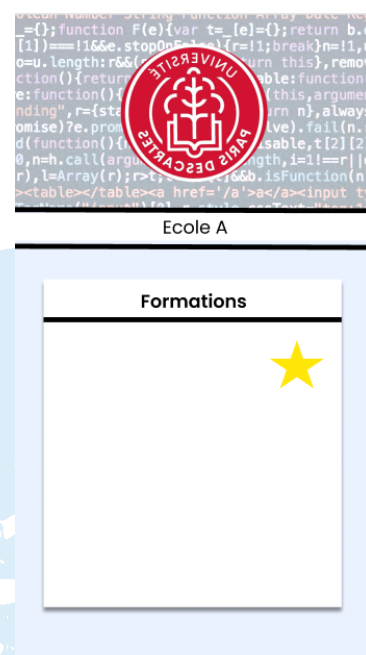
Les améliorations possibles sont multiples.

Tout d'abord, nous avons pensé au fait que chaque utilisateur puisse enregistrer les formations qui l'intéressent. Par la suite, dans une page "favoris", disponible pour chaque utilisateur, on aurait la possibilité de voir la liste des formations préférées.



<- Page "Favoris"

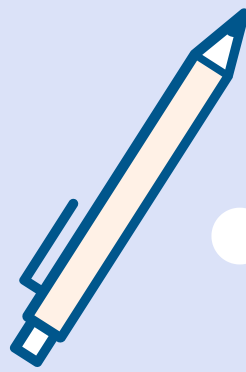
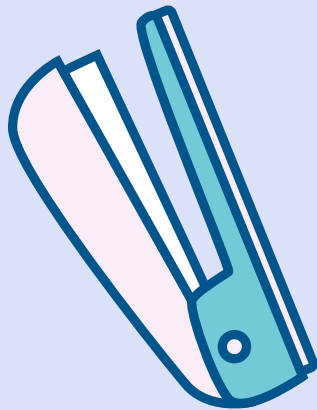
Bouton ajouté sur la page formation pour pouvoir ajouter la formation au favoris ->



Une autre amélioration serait de pouvoir contacter directement l'école par le biais de l'application. Pour l'instant, nous pouvons seulement afficher les informations de la formation que nous recherchons. Nous pourrions donc rajouter un bouton lié à leur site web pour que l'on puisse voir toutes les informations complémentaires qui ne sont pas disponibles sur notre application.

Il pourrait aussi y avoir un lien vers l'espace de candidature afin de pouvoir faciliter cette dernière pour les utilisateurs qui ont parfois du mal à trouver ces informations sur les sites des écoles en question.

Dernière amélioration à laquelle nous avons pensé : rajouter un rôle pour faire une distinction entre les clients de l'application et les élèves qui sont inscrits dans les écoles afin de pouvoir créer des espaces de conversation sur l'application. Sur chaque page de formation, il y aurait donc un bouton pour pouvoir accéder à un espace de discussion avec tous les élèves de la formation en question ainsi que tous les utilisateurs intéressés par cette dernière. Cela permettrait de pouvoir avoir accès aux réponses à des questions fréquentes et aussi poser nos propres questions à notre tour. On pense donc plus judicieux, si on fait une suite, de faire des conversations groupées et non individuelles.



# BILAN

Pour conclure, ce projet a permis à l'ensemble de notre équipe d'aborder les différentes étapes de mise en place d'application Android.

En effet, l'élaboration de ce dernier a été pour nous l'occasion d'expérimenter et de comprendre la programmation sous Android Studio mais également d'acquérir de nouvelles compétences dans l'utilisation du langage Java mais également dans la prise en main d'Android Studio afin de mener notre projet à bien.

La réalisation de ce projet a été pour nous une expérience enrichissante qui a su solliciter notre curiosité ainsi que des compétences travailler dans d'autres matières (SGBD, Programmation) . La répartition du travail ainsi que la mise en place d'entraide au sein de notre groupe ont su renforcer nos liens ainsi que notre expertise dans la mise en place des principales fonctionnalités.

Le travail de groupe a donc été bénéfique car il a permis à notre groupe d'évoluer dans une ambiance agréable malgré la charge de travail conséquente de ce projet ainsi que ceux des autres matières.