

# Projet HyperStack Android - Rapport de Développement

Réalisé par Gabriel EYAF AA JEEVAN KUMAR, Paul MARQUES, Julien MERCIER,  
Sébastien POSTANSQUE et Tom REDON - M2 Logiciels

Lien GitHub : <https://github.com/TomRedLev/Android>

|   |          |
|---|----------|
| <b>Architecture générale du projet :</b>                | <b>3</b> |
| <b>Organisation adoptée :</b>                           | <b>4</b> |
| <b>Difficultés rencontrées et solutions apportées :</b> | <b>5</b> |
| Gabriel EYAF AA JEEVAN KUMAR :                          | 5        |
| Paul MARQUES :  | 6        |
| Julien MERCIER :  | 7        |
| Sébastien POSTANSQUE :                                  | 8        |
| Tom REDON :   | 9        |

## Architecture générale du projet :

Pour ce projet, nous devons réaliser l'application HyperStack.

Pour cela, nous avons décidé de réaliser l'application en plusieurs étapes.

Tout d'abord, nous avons implémenté en diverses applications les différentes fonctions que peuvent avoir une planche.

Avant tout, nous avons réalisé un "hub" qui permettaient d'accéder aux différentes activités via divers boutons.

Ainsi, nous avons réalisé l'implémentation des différentes planches sous formes de différentes activités.

Nous avons d'abord réalisé la planche permettant de réaliser des croquis. L'une des améliorations possibles sur cette partie serait de rajouter des outils de dessin autre que le crayon basique implémenté.

La planche suivante réalisée était l'application permettant d'intégrer du texte. L'application peut donc prendre un texte de grande dimension et le reproduire.

Puis, la planche d'image a été réalisée. Pour cela, nous avons dû accéder au storage externe du téléphone qui nous permettra d'accéder aux photos disponibles sur le téléphone.

L'utilisateur pourra choisir la photo qu'il souhaite ou en prendre une pour pouvoir l'utiliser dans une planche.

Par la suite, nous avons réalisé la planche de son.

La planche sera capable d'enregistrer avec les boutons record et stop et pourra lire le texte avec le bouton play. Pour cela, il aura fallu accéder au paramètre d'accès au micro.

Enfin nous avons réalisé la planche GPS. Pour cela, nous avons accédé aux divers paramètres de géolocalisation. Les paramètres de géolocalisation renvoient la position et l'utilisateur peut donc se servir de ça. Une amélioration possible serait de créer des liens avec les coordonnées.

Ceci n'a été que la première partie de notre réflexion.

Par la suite, nous avons essayé de mettre en place la multitude de planches possibles.

Ainsi, nous avons tenté de pouvoir faire en sorte d'avoir une liste de notes et que chaque note puisse avoir un traitement différent en fonction des choix réalisés autour de cette dernière.

Nous nous sommes donc servis de nouveau des planches développées précédemment pour en faire des versions plus poussées et plus simples d'utilisation. Cela nous a permis de répondre un peu mieux à la problématique de superposition des planches mais aussi à celle de la multitude de planches sur une seule "note".

Par la suite, nous avons modifié grandement le projet. Il se décompose ainsi de la façon suivante :

A l'ouverture de l'application, vous pouvez cliquer sur le bouton Add Note pour ajouter une nouvelle note.

Choisissez son nom et appuyez sur Ok.

Cliquez sur la note souhaitée.

Cela vous conduira sur un nouvel affichage. En haut à droite, vous pourrez observer quatre boutons : Draw, Images, Audio et GPS.

Si vous cliquez sur Draw, vous pourrez réaliser un dessin avec le doigt ou bien alors utiliser les divers outils programmés : Eraser, Stroke, Fill, Black, Red.

Si vous cliquez sur Images, vous pourrez sélectionner une image dans le stockage de l'appareil ou prendre une photo avec la caméra.

Si vous cliquez sur Audio, vous pourrez choisir un fichier audio dans le stockage de l'appareil ou réaliser un enregistrement en vous servant des boutons Record et Stop.

Pour écouter l'enregistrement, vous pourrez appuyer sur Play.

Si vous cliquez sur GPS, vous pourrez appuyer sur le bouton Get Your Location qui renverra votre localisation actuelle.

## Organisation adoptée :

Pour l'organisation générale du projet, nous nous sommes répartis les tâches comme ceci :

- Gabriel EYAFKAA JEEVAN KUMAR : Outils de dessin et sauvegarde (non réussie c'est Sébastien qui l'a fait).
- Paul MARQUES : Forme linéaire dans dessin et des tentatives diverses.
- Julien MERCIER :
- Sébastien POSTANSQUE : Globalisation de l'utilisation des planches et de leur superposition.
- Tom REDON : Implémentation des diverses planches possibles et de leurs fonctionnalités diverses (Croquis, Texte, Image, Sons, GPS). J'ai aussi tenté de réaliser le Service et le Broadcast Receiver nécessaire.

Dans la prochaine partie, nous détaillerons chacun l'organisation que nous avons eu pour réaliser notre tâche.

## Difficultés rencontrées et solutions apportées :

Chaque personne détaillera ici les problèmes qu'elle a pu rencontrer.

Nous détaillerons aussi la réalisation de nos différentes tâches et la façon dont notre réflexion a pu évoluer au cours du développement du programme.

En effet, notre vision de l'application a énormément changé au cours du développement et nous avons essayé de réaliser l'application la plus simple d'utilisation et la plus pratique possible.

## Gabriel EYAF AA JEEVAN KUMAR :

J'ai essayé de faire la sauvegarde des données en utilisant une BDD SQLite sans succès, au final Sébastien a réussi à le faire sans en avoir besoin (avec Shared Preferences).

Ensuite j'ai créé les outils pour dessiner (reset/fill/stroke/ 2 couleurs) et ça fonctionne correctement. Pour faire le système de dessin Tom à utiliser une CustomView donc j'ai dû l'encapsuler dans un layout afin que je puisse avoir ma barre d'outils. Afin que tous les outils fonctionnent j'ai dû changer le comportement de base de sa fonction onTouchEvent() car sinon le fait de changer de couleur/style de pinceau modifiait les tracés déjà effectués.

Finalement j'ai essayé de faire les tâches restantes que Sébastien m'a indiqué, sans succès.

## Paul MARQUES :

Il n'était pas évident de lire les url comme des hyperlinks car le texte était contenu dans un EditText.

Une méthode fonctionnait sur un petit programme de test avec `editText.setMovementMethod(LinkMovementMethod.getInstance());` après avoir set du text. Cela reconnaît les url mais ne fonctionnait pas sur le programme.



Julien MERCIER :

## Sébastien POSTANSQUE :

Pour ce projet, mon travail a été de regrouper tous les outils (Texte, GPS, Dessin, Ajout d'image et de fichier audio) dans les planches.

La difficulté principale a été de rendre l'édition d'une note possible avec tous ces outils. J'ai commencé par essayer d'implémenter chaque outil en utilisant une view principale dans laquelle les outils se superposent. La contrainte principale a été de faire en sorte que ces outils aient une disposition cohérente dans une note. Je n'ai pas réussi cette tâche en utilisant une view. J'ai donc choisi d'utiliser un EditText.

Chaque note est composée d'un EditText qui regroupe le contenu de tous les outils. En effet, les images, le texte et les coordonnées GPS peuvent être ajoutés dans un EditText. Cela permet de rendre la note éditable facilement. Il est possible d'ajouter et supprimer du texte, des images et des coordonnées GPS très facilement. La difficulté principale a été d'ajouter les images dans un EditText. J'ai compris après beaucoup d'essais que cela était possible. J'ai adapté chaque outils pour les rendre fonctionnel avec une note.

Parmi les difficultés rencontrées, j'ai dû réaliser beaucoup de tests pour les demandes de permissions. En effet, les permissions d'accès en écriture et lecture des fichiers sont nécessaires pour récupérer et créer une image. A plusieurs reprises, les permissions pour l'activity Image étaient affichées par l'application mais l'activity s'arrêtait de fonctionner. L'erreur était dû au fait que l'activity exécutait des méthodes qui avaient besoin des permissions pour fonctionner. Les demandes de permissions étant asynchrone, l'application n'attendait pas que l'utilisateur accepte les droits avant d'exécuter les méthodes.

La difficulté principale a été d'implémenter le système de sauvegarde des notes. Au début du projet, j'ai considéré d'utiliser une base de données sqllite pour sauvegarder toutes les données d'une note. J'ai rencontré des soucis durant l'implémentation, en particulier pour sauvegarder les images. L'implémentation me semblait assez longue et j'ai changé la méthode de sauvegarde en utilisant les préférences partagées (SharedPreferences). Pour sauvegarder une image, je récupère son chemin et je l'ajoute dans la sauvegarde. Lorsque la note est chargée (ouverte), les données sont récupérées dans les préférences et j'utilise un regex pour détecter les liens que je souhaite récupérer pour créer les images sous format Drawable. Pour faire le lien entre le contenu d'une note et la note listée sur la page principale de l'application, j'utilise le nom de la note en tant que clé dans les préférences partagées. Ensuite j'envoie le nom de la note directement à l'activity qui récupère ainsi ses données.

Ce système d'envoi de données entre activity a été une solution assez pratique, en particulier pour récupérer le nom d'une image depuis l'activity Image à l'activity Note.

L'un des objectifs était de pouvoir cliquer sur un lien internet inscrit dans une note. Paul a commencé l'implémentation et j'ai par la suite finalisé la fonctionnalité. Il est possible d'ouvrir un lien internet, un numéro de téléphone ainsi qu'un email. Le lien hypertexte est détecté automatiquement lorsqu'on écrit dans une note.

## Tom REDON :

J'ai rencontré divers problèmes pour réaliser les différentes fonctionnalités de ces planches.

En premier lieu, j'ai voulu réaliser l'application un peu plus globale qui permettaient de lancer les différentes applications, tel un "hub".

Pour cela, je n'ai rencontré aucun problème de réalisation car l'application était assez simple et nous avons eu l'occasion de voir divers exemples en cours.

Puis j'ai réalisé la planche de croquis. Pour cela, j'ai dû trouver une solution pour pouvoir représenter le dessin réalisé au doigt par l'utilisateur.

J'ai tenté de le faire sans programmer de "view" en premier lieu. Je n'ai pas réussi à le faire donc j'ai réalisé une reprogrammation d'une "view" qui s'appelle DrawView.

Cette vue est ensuite utilisée dans l'activité pour prendre en charge les dessins au doigt de l'utilisateur.

Pour améliorer cette application, j'aurais pu réaliser divers modifications telles qu'ajouter des types d'outils différents pour réaliser des croquis. J'ai préféré me concentrer à réaliser des versions plus simples de chaque type de planche pour pouvoir proposer de nombreuses planches utilisables.

Ensuite, j'ai réalisé l'application Texte.

Pour cela, je n'ai pas rencontré de problème car je me suis servi des différents exemples vus en cours, mais aussi des différents TP tels que le GeoChatRoom pour le réaliser.

Puis, j'ai réalisé l'application Images.

Pour cela, j'ai eu divers problèmes. En premier lieu, il m'a fallu identifier les différentes permissions nécessaires à la réalisation de cette application. J'ai donc finalement trouvé une solution simple en utilisant la permission d'utiliser le stockage externe.

Enfin, j'ai dû me débrouiller pour simplifier l'affichage au maximum pour faire quelque chose de joli et d'utilisable et cela a été le plus compliqué. J'ai tout d'abord voulu réaliser une image cliquable mais je me suis dit que c'était une mauvaise idée et que l'interface ne serait peut-être pas très clair. J'ai donc décidé d'ajouter un bouton pour pouvoir aider l'utilisateur à comprendre ce qu'il devait faire.

Par la suite, j'ai réalisé l'application Sons.

Pour réaliser l'application Sons, j'ai dû trouver un moyen de faire en sorte que le micro soit utilisable. Puis j'ai dû comprendre comment lancer l'enregistrement du micro et le stopper. Enfin, j'ai dû réaliser la partie qui me permet d'écouter l'enregistrement. J'ai finalement trouvé toutes ces réponses dans la documentation Android. Le test a été plus complexe, car je ne possédais pas de micro sur mon PC de programmation.

Enfin, j'ai réalisé l'application GPS.

Pour cela, je n'ai pas rencontré de nombreux problèmes car j'ai pu me servir de l'exemple programmé dans le TP GeoChatRoom.

Le plus gros problème rencontré était le fait de trouver un moyen d'activer la géolocalisation sur l'émulateur. Une fois résolu, cela s'est déroulé sans accroc.