

# WorkHub

Rapport

**Julien ROUGERON, Florian PAILLARD, Jean-Loup  
ROUSSEL-CLOUET, Martin SALLES, Louise  
GRANDJONG  
13/06/2013**

# Table des matières

Introduction.....	3
I. Fonctionnalités prévues pour l'application .....	4
A. Cahier des charges .....	4
B. Fonctionnalités.....	4
II. Etude ergonomique de l'application .....	6
A. Etude de l'ergonomie .....	6
B. Choix des couleurs.....	6
C. Police des textes.....	7
D. Disposition.....	7
E. Raccourcis .....	8
III. Architecture de l'application et structures de données.....	9
A. Présentation des modèles utilisés pour les éléments .....	9
B. Application Android .....	9
C. Application PC/Tablette .....	10
D. Lien avec le système multi-agent.....	11
IV. Modifications apportées au cahier des charges .....	13
V. Evaluation par rapport aux critères ergonomiques.....	14
A. Compatibilité .....	14
B. Guidage.....	14
C. Homogénéité.....	15
D. Adaptabilité.....	16
E. Gestion des erreurs .....	16
F. Concision .....	16
VI. Améliorations possibles .....	17
A. Fenêtre d'accueil.....	17
B. Toucher d'autres dispositifs.....	17
C. Augmenter les types d'éléments manipulables.....	17
D. Valuation des post-it.....	18
E. Couleur par défaut par utilisateur.....	18

F. Sauvegarde du Hub.....	18
Conclusion .....	19

## Introduction

L'émergence des technologies numériques pousse de plus en plus les groupes de travail à abandonner le support papier (ou tableau) pour se tourner vers des outils de travail sur support informatique. En effet, ces outils permettent une persistance et une diffusion de l'information beaucoup plus efficace. Cependant, nous estimons que les outils à notre disposition ne permettent pas de travailler dans les mêmes conditions. Les solutions de travail collaboratif existantes sont souvent limitées. Certaines sont trop linéaires et ne permettent pas de représentation spatiale des éléments (documents texte), ou proposent cette représentation spatiale, mais sans travail collaboratif. Des solutions en ligne existent (Prezi par exemple), mais l'information devient publique, ou la solution payante.

En nous basant sur ce constat, nous avons décidé de travailler à proposer une solution logicielle permettant des séances de brainstorming alliant les avantages du numérique en essayant de conserver ceux du support papier. Nous voulons donc offrir la possibilité aux utilisateurs de synchroniser les appareils qu'ils utilisent, à travers notre application, pour réfléchir ensemble à la problématique justifiant le brainstorming en manipulant autant d'éléments graphiques de type post-it qu'ils le souhaitent.

# I. Fonctionnalités prévues pour l'application

## A. Cahier des charges

Un cahier des charges précis a été construit au début du projet pour étudier les solutions existantes, les besoins du public concerné et établir décider des fonctionnalités à mettre en place.

Une étude a été réalisée auprès de 98 étudiants pour connaître leur utilisation des outils de réflexion collaboratifs. Nous avons pu établir qu'ils utilisent beaucoup d'applications collaboratives comme Google Drive ou Dropbox. Ils ont alors recours à des documents de type texte et à des images pour échanger de l'information. Environ 79% des étudiants sondés estiment que les fonctionnalités proposées par WorkHub sont intéressantes et pourraient leur permettre de travailler plus efficacement. Les 21% ayant exprimé l'avis contraire se sont déclarés satisfaits des outils existants.

Cette étude a donc montré que le public étudiant est plutôt intéressé par le projet WorkHub. Celui-ci correspond donc à un besoin, ce qui justifie son développement.

## B. Fonctionnalités

### Aspect collaboratif

L'application permet aux utilisateurs de travailler de manière collaborative sur un réseau local. L'un d'eux héberge le Hub et les autres viennent se connecter à lui. Les utilisateurs agissent ensuite sur le même Hub et partageront donc directement les mêmes informations, qu'ils peuvent éditer et partager.

Plusieurs applications ont été développées pour permettre aux utilisateurs d'utiliser plusieurs types de dispositifs. Une première application Android rend WorkHub disponible sur smartphones et tablettes Android. Une seconde développée en Java à l'aide de la bibliothèque MT4j propose une interface Ordinateur et table tactile.

### Éléments manipulables

Différents types d'éléments sont mis à la disposition des utilisateurs pour représenter un maximum d'idées. Dans cette première version de l'application, nous proposons des éléments de type :

- Texte : Un titre et un texte de contenu modifiables, inscrits sur un post-it ;
- Image : Une image modifiable et son titre ;

- Lien Web : Un lien web et son titre explicatif, qui pourra s'ouvrir directement dans le navigateur par défaut du système ;
- Fichier : Un fichier qui pourra s'ouvrir dans le logiciel par défaut du système utilisé.

Pour faciliter la manipulation de ces éléments, il est possible de les regrouper par tas, de changer leur couleur, ou encore de masquer ceux qui ne sont pas utilisés.

### **Export PDF**

A la fin d'une séance de travail, il peut être intéressant de résumer l'ensemble des éléments créés. C'est pour cette raison que nous avons développé une fonctionnalité d'export PDF. Les utilisateurs peuvent ainsi disposer d'un fichier récapitulatif tout ce qui a été décidé pendant une réunion, qui pourra être imprimé ou diffusé.

## II. Etude ergonomique de l'application

### A. Etude de l'ergonomie

L'application WorkHub que nous proposons, peut être hébergée par plusieurs dispositifs : smartphone, tablette, pc, et table tactile. Une application multiplateforme pose des problèmes d'ergonomie supplémentaires.

L'interface proposée doit être étudiée pour satisfaire les besoins de l'utilisateur simplement et intuitivement. Cela implique d'étudier les réflexes qu'un utilisateur lors de l'utilisation d'un dispositif, et lui proposer les même outils d'ergonomie qu'il peut rencontrer dans d'autre logiciel. On conserve ainsi une homogénéité d'utilisation au sein d'un dispositif, mettant l'utilisateur en confiance.

Seulement, ces réflexes sont propres à un dispositif. Ainsi l'utilisation d'une tablette ou d'un pc implique chacune des notions d'ergonomies différentes. Par exemple l'accès à un menu se fait intuitivement par un clic droit sur ordinateur, alors que l'utilisateur aura tendance à chercher le menu grâce à un clic long sur tablette.

Il a été naturellement nécessaire d'adapter l'interface à chaque dispositif, impliquant alors une nouvelle contrainte : l'homogénéité de l'ergonomie de l'application indépendamment du support. En effet, adapter l'interface à chaque dispositif ne suffit pas, il faut également veiller à ce que l'utilisation de l'application en générale reste cohérente de manière à ne pas perturber l'utilisateur qui utilise alternativement chaque dispositif. Certains repères doivent être gardés, parfois au détriment de l'ergonomie naturelle du dispositif, évitant à l'utilisateur de devoir réapprendre à manipuler l'application.

Les questions de différences ergonomiques entre les dispositifs se sont d'autant plus posées que l'application MT4j est utilisée à la fois sur table tactile et sur ordinateur. Il a fallu procéder à des choix de conception pour éviter de produire une application hybride qui ne soit adaptée à aucun des deux dispositifs.

### B. Choix des couleurs

Un bon choix de couleur contribue à améliorer l'expérience de l'utilisateur. Une couleur judicieuse peut augmenter la lisibilité et l'attractivité d'une interface, il faut donc établir une charte graphique précise.

Le fond de l'application est un gris clair, avec le logo de l'application en filigrane. Cette couleur neutre et pale permet de créer un fond mettant en avant

les différents éléments. De plus, les post-it ont par défaut une couleur jaune rappelant l'objet post-it habituel.

Enfin, il est possible de modifier la couleur des post-its créés. Cela permet à l'utilisateur de créer une lisibilité qui lui est adaptée.

## C. Police des textes

Le choix de la police est un élément important car il va définir le confort de lecture de l'utilisateur, et la lisibilité de l'application.

Notre première décision a été de choisir une seule police pour tous les textes de l'application. En effet, l'utilisation de multiples polices réduit l'homogénéité de l'interface, et peut perturber l'utilisateur.

Le type de police est également un facteur essentiel. Une police à empattement tel que Time New Roman par exemple, facilite la lecture de document de taille importante, mais alourdit grandement un texte. Notre application proposant la gestion de post-it, et donc de texte très court, nous avons opté pour l'utilisation d'une police légère, sans empattement et facile à lire sur un écran.

## D. Disposition

WorkHub est une application aidant à la réunion de travail en équipe. Elle fonctionne comme un brainstorming. On manipule donc des éléments, que l'on organise ensuite comme bon nous semble pour regrouper des idées.

Il n'y a donc pas de disposition prédéfinie pour les post-its et éléments présents sur l'interface. Plutôt que de définir une disposition, il a plutôt fallu rendre l'interface la plus modulaire possible, permettant à chaque utilisateur de la personnaliser et l'adapter à ses besoins.

Les raccourcis nous permettant d'envoyer, recevoir, sont par défaut situés dans les coins de l'interface de manière à laisser le plus de place possible à la manipulation des éléments. De plus on peut les modifier et déplacer.

Les post-its apparaissent à l'endroit où l'on a cliqué pour les créer, et il est bien sûr possible de les déplacer.

Nous donnons donc à l'utilisateur la possibilité de contrôler complètement la disposition des éléments au sein de son interface.

## E. Raccourcis

Toujours dans une idée de modularité et pour laisser à l'utilisateur le total contrôle de son interface, il est possible sur l'application Android de modifier l'emplacement des raccourcis, d'en enlever, d'en rajouter.

Ces raccourcis sont :

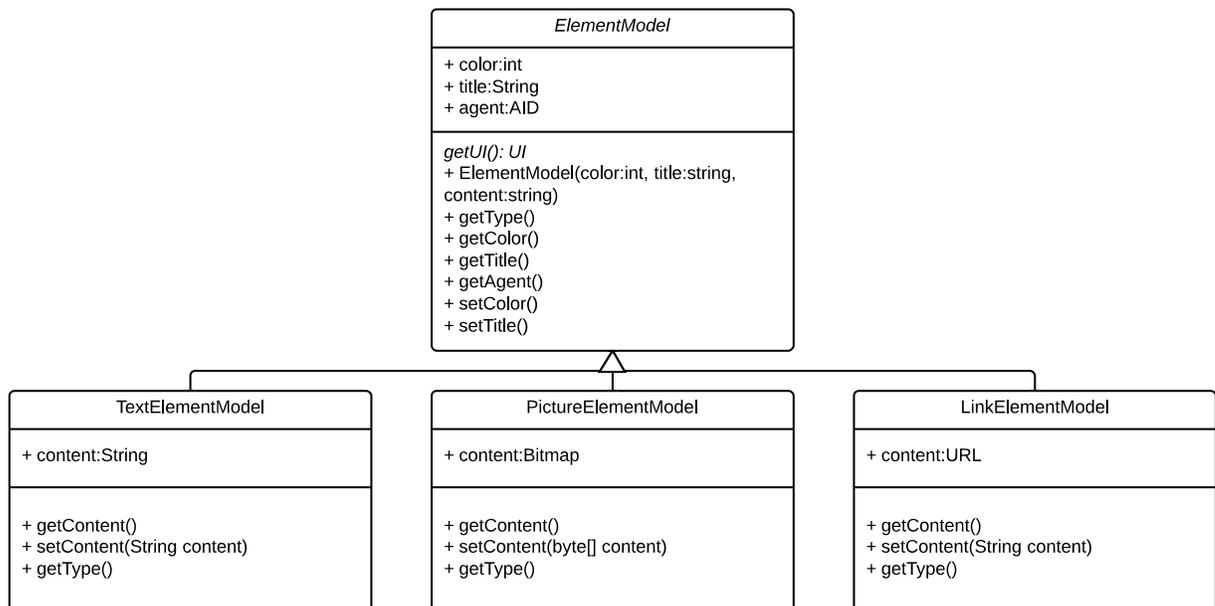
- Envoyer : envoyer un élément
- Recevoir : boîte aux lettres
- Editer : éditer un élément

L'utilisateur peut choisir de les afficher ou non dans son interface. Une fois qu'ils sont dans l'interface, les raccourcis se présentent sous forme de rond, qu'il est possible de placer où l'on veut. Ensuite un simple glisser-déposer d'un élément sur le raccourci l'active et effectue l'action.

### III. Architecture de l'application et structures de données

#### A. Présentation des modèles utilisés pour les éléments

Un élément est un agent jade sur la plateforme multi-agents. Il contient un modèle d'élément de type `ElementModel`. `ElementModel` est une classe abstraite dont héritent les différents types d'éléments (`TextElementModel`, `PictureElementModel` et `LinkElementModel`).

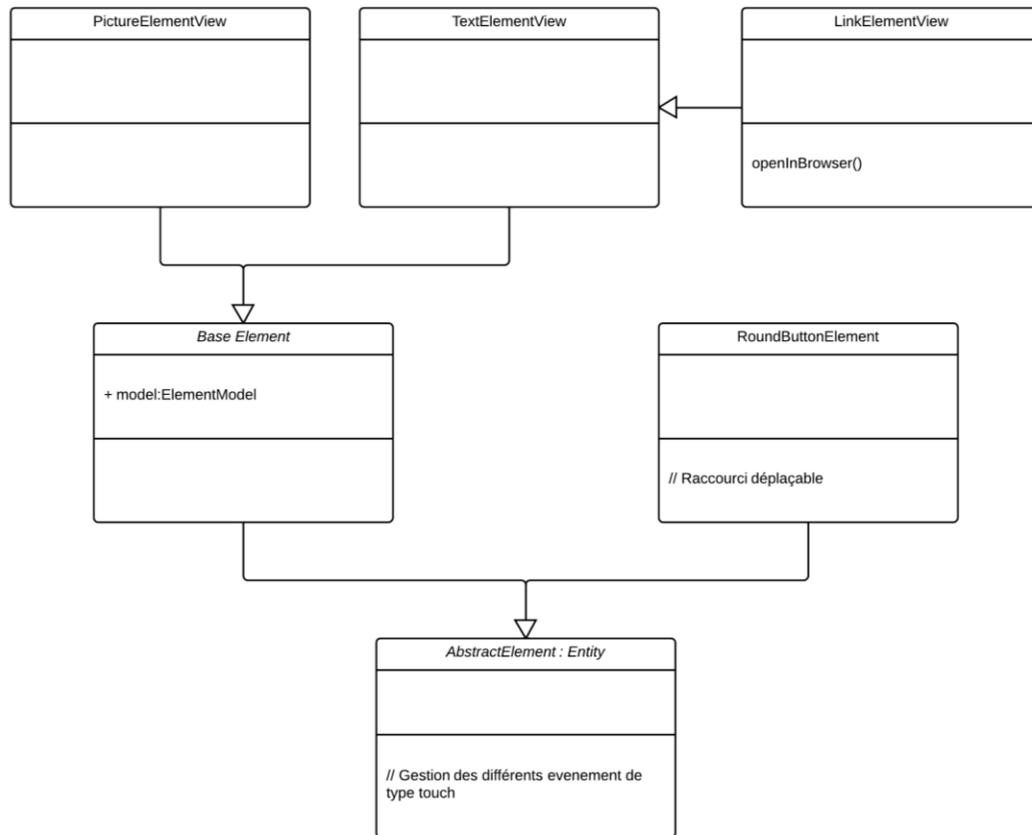


#### B. Application Android

L'application Android est réalisée à l'aide de la librairie `AndEngine` (<http://www.andengine.org/>). Cette librairie facilite la manipulation d'éléments en `OpenGL2`. Cette technologie permet une meilleure fluidité que les fenêtres ou canvas natifs Android.

L'ensemble des éléments manipulables dans `AndEngine` hérite de la classe `Entity`. Chacun d'eux peut être ajouté sur la « `Scene` » principale afin d'être dessiné. La classe `BaseElement` permet la gestion des fonctionnalités de base d'un post-it. Chacune des classes héritant de `BaseElement` se spécialise pour mieux s'adapter à son `ElementModel`.

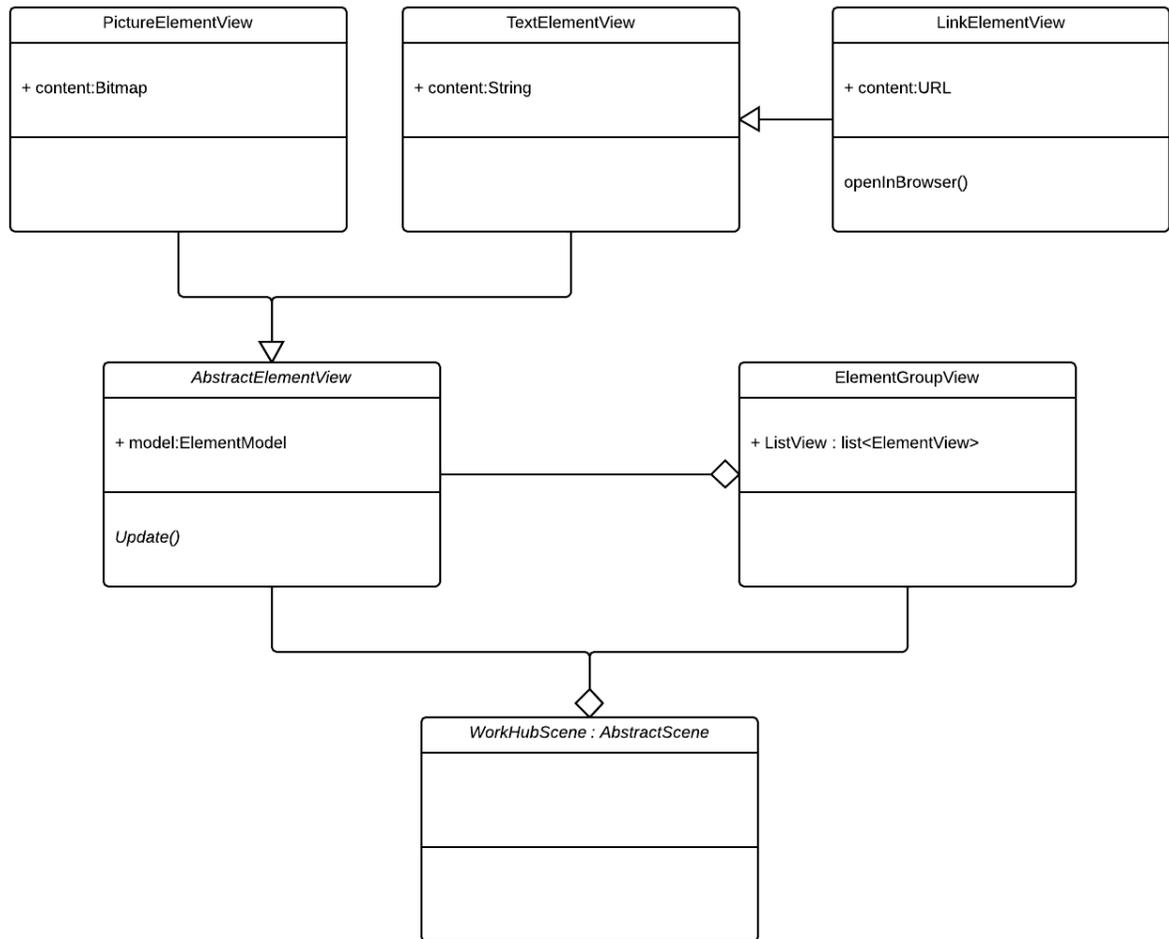
Voici l'UML reprenant les différents objets graphiques implémentés dans l'application :



## C. Application PC/Tablette

L'application PC/Tablette est conçue à l'aide de la bibliothèque MT4j (Multi-Touch For Java). Elle est composée d'une scène qui est la fenêtre principale. Cette scène contient tous les éléments graphiques de l'application et les met à disposition de l'utilisateur.

Les éléments sont à nouveau organisés dans une hiérarchie de classes. Cependant, certaines modifications ont été apportées à la hiérarchie des classes ElementModel. Tout d'abord, les éléments de type lien héritent des éléments textes, car les attributs des deux classes sont les mêmes. Les éléments liens disposent simplement d'une méthode permettant d'ouvrir le lien dans le navigateur web du système. Par ailleurs, des groupes d'éléments font leur apparition pour permettre de regrouper et manipuler facilement plusieurs éléments en même temps.



## D. Lien avec le système multi-agent

Dans notre plateforme, les différents clients et échangent des documents grâce à un système multi-agent JADE, qui a constitué notre projet IA04.

Chaque client et élément de la plateforme est un agent jade. Un échange entre eux correspond donc à un message de requête d'une information à un agent, puis à un message de retour contenant la réponse.

Du point de vue du système JADE, les clients sont des « GUI-Agents ». C'est à dire que ce sont des agents qui possèdent une interface graphique. C'est ici que se fait le lien entre notre interface et notre système.

Lorsque que l'utilisateur effectue une demande via l'interface, cette dernière crée un "guiEvent". L'agent qui possède cette interface intercepte cet évènement,

l'interprète, et peut enfin communiquer avec les autres agents pour répondre à cette demande.

Une fois la réponse obtenue, l'agent renverra à son interface graphique un message de type "FirePropertyChange" pour lui notifier les changements.

Le travail principal pour effectuer le lien entre le système multi-agent et notre interface, consistait à normaliser les messages d'échange. Au travers d'une liste de constantes partagées par l'interface et le système multi-agent, les différents types de message ont été définis.

Enfin, pour le système comme pour l'interface, nous avons mis en place une fonction de réception d'un message testant le type de message, pour effectuer la bonne fonction correspondante.

## IV. Modifications apportées au cahier des charges

Le prototype de l'application WorkHub réalisé lors de ce semestre respecte le cahier des charges que nous avons réalisé en début de semestre. En effet, la manipulation d'éléments est complète et disponible sur tous les supports prévus (tablette, smartphone, ordinateur et table tactile).

La seule modification que nous avons faite concerne la gestion de fichiers. Nous avons constaté que passer un fichier dans un message était problématique pour plusieurs raisons. Tout d'abord, la contrainte était technique. En effet, nous passons le contenu d'un élément dans un message du côté du système multi-agent. Ce message est sous forme de Json. Encoder un fichier pour le partager via ce système impliquait des messages très lourds ou une grande limitation de la taille des fichiers disponibles.

De plus, dans le cadre du développement de ce prototype d'application de brainstorming, nous n'avons pas trouvé que cela était très pertinent. En effet, le concept le plus important reste de pouvoir partager des idées spontanées.

C'est donc pour ces raisons que les post-its contenant des fichiers ne sont pas fonctionnels.

## V. Evaluation par rapport aux critères ergonomiques

Lors de la conception de l'application WorHub, nous avons réfléchi aux différents critères ergonomiques présentés lors de cours de NF28 afin de la rendre plus pertinente et agréable à utiliser. Voici donc un résumé de la réflexion que nous avons portée concernant ce problème.

### A. Compatibilité

L'application WorkHub est conçue pour être utilisée dans le cadre d'un brainstorming. L'utilisateur type va donc être un étudiant ou un professionnel travaillant au sein d'un groupe.

Dans l'idée de numériser un système de brainstorming papier et proposer des techniques innovantes, nous avons réfléchi aux méthodes mises en œuvre par l'utilisateur. Afin d'être en adéquation avec cet utilisateur qui cherche à communiquer physiquement des idées en les posant sur du papier, nous avons proposé la création de post-its. L'utilisateur peut facilement naviguer entre les différentes idées, les différents liens ou images qu'il a créés. De plus, il peut interagir avec autrui, intervenir sur une idée d'un autre utilisateur, envoyer et recevoir des post-its provenant d'un de ses collaborateurs. Enfin, l'ensemble du groupe peut avoir une interface collective, comme la table tactile, où tous peuvent discuter sur un même support.

Nous nous sommes penchés sur le problème de la compatibilité lorsque nous avons conçu les menus et sous menus. Nous avons tenu à suivre le cheminement de pensée d'un utilisateur aillant accès à l'application afin de la rendre intuitive.

Nous pensons donc que l'application WorkHub respecte le critère de compatibilité.

### B. Guidage

L'aspect incitation de notre application est plutôt implicite. Nous n'avons pas fait de bulles d'aide explicites. Néanmoins, nous y avons accordé une grande importance. Par exemple, lorsqu'un utilisateur souhaite éditer un élément, la réponse de l'application n'est pas la même selon le type d'élément ce qui diminue le risque d'erreur. Si l'utilisateur veut apporter des modifications sur un élément de type texte, il l'ouvrira et, si l'édition est disponible pour cet élément, l'interface d'édition apparaîtra avec les différents champs sur lesquels il peut agir. La liste des saisies attendues est donc

explicite. Quant au format des données, il est explicite puisque, si l'utilisateur accède à un champ texte, le clavier pour écrire s'affiche, et pour les couleurs, un cadre de choix de couleur apparaît.

Les retours utilisateurs sont divers et aident au guidage de ceux-ci dans l'application. Par exemple, si un utilisateur souhaite rejoindre un hub, l'application Android affiche un pop up indiquant si la connexion a réussi. De même, si l'édition est impossible car quelqu'un d'autre édite cet élément, alors un pop up apparaît.

Sur l'interface Android, le clic long n'avait au départ pas d'indicateur. Si l'utilisateur n'appuyait pas assez longtemps, il pouvait croire qu'il n'y avait pas d'action possible. C'est pourquoi nous avons ajouté un élément permettant de voir lors de l'appui qu'un menu allait apparaître.

Au niveau du design, nous avons accordé de l'importance au groupement d'information du même type. Par exemple, les raccourcis apparaissent en cercle bleus clair, les menus ont tous la même ergonomie afin d'apporter une grande cohérence à l'application.

Tous ces éléments aident l'utilisateur à prendre en main facilement une interface intuitive. Ainsi, l'apprentissage n'est pas fastidieux et n'est pas une perte de temps lors du brainstorming. Une équipe peut donc rapidement choisir de passer à ce support sans craindre de devoir abandonner un moment de réflexion qui pourrait être important dans la réunion.

## C. Homogénéité

Le problème de l'homogénéité a été complexe à régler. En effet, l'application WorkHub est disponible sur de nombreux supports. Il fallait donc en même temps gérer une cohérence entre les différents dispositifs afin que naviguer entre n'implique pas un réapprentissage, et en même temps adapter l'interface aux différents dispositifs. Les actions possibles sur une tablette, un ordinateur et une table ne peuvent pas être exactement les mêmes. Alors que sur une tablette, on va utiliser les doigts, ceux-ci vont être remplacés par une souris lors du passage sur ordinateur. De plus, l'affichage des menus sur Android ne peut pas être le même que sur un table tactile dont le format est bien plus grand. Nous avons donc dû concevoir des solutions qui répondaient aux nécessités des différents dispositifs tout en gardant une homogénéité au niveau des possibilités (création, édition etc.) et du design.

## D. Adaptabilité

L'application peut être adaptée par l'utilisateur selon ses préférences. L'interface Android propose par exemple d'ajouter et supprimer des raccourcis ce qui permet d'avoir rapidement accès aux commandes les plus utilisées. Ainsi, l'utilisateur a une maîtrise plus importante de son interface et peut agir plus rapidement. Mais les commandes restent tout de même accessibles à l'utilisateur via les menus.

De plus, cette possibilité d'ajouter des raccourcis rend notre application flexible, ce qui est important pour l'ergonomie.

## E. Gestion des erreurs

Les erreurs qui peuvent arriver au cours de l'utilisation sont diverses. Par exemple, un utilisateur peut perdre la connexion et ne plus être sur le réseau. Auquel cas, il peut se reconnecter facilement en rejoignant le Hub sur lequel il travaillait. Mais il ne va pas perdre ses éléments. En effet, lors de la reconnexion, grâce au menu « importer », il peut récupérer ses post-its à l'état où il les avait laissés.

D'autres types d'erreurs ont été gérés comme l'édition simultanée de deux utilisateurs d'un même élément. L'utilisateur a un retour de la part de l'interface qui lui indique quel type de problème a été rencontré. L'utilisateur ne fait donc pas d'action qui n'aurait aucun retour et le déstabiliserait.

## F. Concision

Deux aspects font que notre application est concise. Tout d'abord, les noms de menus ont été réfléchis afin qu'ils soient évidents et rapide à comprendre. Lorsqu'un utilisateur veut importer un élément, la liste affichée correspond au titre du post-it (et non pas au nom de l'agent associé).

Par ailleurs, si un utilisateur désire une plus grande concision et ne pas passer par les menus, il est toujours possible d'ajouter des raccourcis.

En général, en tant qu'utilisateur, nous avons tendance à chercher à utiliser des outils qui ne demandent pas d'effort et se rapprochent de notre méthode de pensée. C'est donc pour cela que nous avons porté beaucoup d'attention lors de la conception du prototype WorkHub aux critères d'ergonomie.

## VI. Améliorations possibles

Cette application a pour but d'être une preuve de concept. Nous voulions surtout proposer un dispositif utilisable facilement et efficacement dans le cadre d'une réunion de projet type brainstorming. Le résultat est satisfaisant dans cette optique mais peut bien sûr être amélioré. Voici donc quelques pistes auxquelles nous avons pensé qui pourraient être explorées dans le futur.

### A. Fenêtre d'accueil

La fenêtre qui s'ouvre quand l'utilisateur lance l'application pourrait gagner en ergonomie. Il pourrait être intéressant de rajouter du texte explicatif pour indiquer à l'utilisateur comment utiliser l'application.

### B. Toucher d'autres dispositifs

Afin d'ouvrir au maximum la possibilité d'utilisation de WorkHub, il serait intéressant que l'application soit disponible sur un plus grand nombre de dispositifs. Le développement a visé les téléphones et tablettes Android, et la bibliothèque MT4j utilisée pour créer l'application PC est limitée aux ordinateurs Windows (et n'est pas compatible avec Windows 8). Il pourrait être intéressant de penser à élargir aux utilisateurs ayant des iPhones ou iPad, des ordinateurs Windows 8 ou des Windows Phone etc.

### C. Augmenter les types d'éléments manipulables

Il est envisageable de proposer de nombreux types de post-its différents comme :

- Post-it de dessin
- Vidéos
- Musique
- Diaporama
- Fichiers

Bien sûr, il faut faire attention à garder une certaine cohérence au sein de l'application afin de ne pas ajouter une infinité de types de post-its qui ne seraient pas pertinents car non utilisés au sein d'une réunion.

## **D. Valuation des post-it**

Il pourrait être envisageable de proposer une fonction de valuation des post-it pour pouvoir les classer par ordre d'importance.

## **E. Couleur par défaut par utilisateur**

À l'heure actuelle, l'utilisateur peut paramétrer la couleur du post-it mais la couleur par défaut est la même pour tous. Cela pourrait être intéressant d'initialiser une couleur par défaut par utilisateur lorsqu'il se connecte. Les ajouts des différentes personnes seraient donc plus clairs.

## **F. Sauvegarde du Hub**

Dans la version de WorkHub que nous avons développée, si la connexion d'un utilisateur est interrompue, il perd son Hub. Si cet utilisateur n'était pas l'hôte, il peut récupérer les éléments en se reconnectant et en utilisant le menu « importer ». Néanmoins, si l'hôte est déconnecté, les autres utilisateurs ne peuvent plus récupérer les éléments. Avoir une possibilité de sauvegarder le Hub permettrait de résoudre ce problème, de quitter l'application et de revenir avec le même état que précédemment.

# Conclusion

À l'issue de ce projet, nous pensons avoir respecté les objectifs que nous nous étions fixés dans le cahier des charges. L'application est fonctionnelle et répond bien aux besoins de l'utilisateur dans le cadre d'un brainstorming.

Le fait d'avoir étudié l'ergonomie a permis de rendre un prototype dont l'utilisabilité est bonne. Les deux applications mises en place ont un fonctionnement proche pour ne pas perdre l'utilisateur, tout en restant adapté au dispositif utilisé. Nous remplissons ainsi l'objectif de permettre différentes utilisations simultanées de WorkHub.

Enfin, nous avons établi plusieurs pistes d'améliorations pertinentes qu'il pourrait être intéressant de creuser pour une meilleure ergonomie de l'application et un panel de fonctionnalités plus étendu. Ces améliorations ne nécessitent pas de refonte des applications et pourraient relativement facilement être mises en place progressivement.