|  |
| --- |
|  |

|  |
| --- |
| **Mobile Degreasor ©** |
| L’application Android qui s’occupe de vos tâches ! |
|  |
| Analyse des besoins, traitement de l’interface homme-machine et technologies associées à Android. |
|  |
| **Sébastien Chénais - Antoine Wollenburger** |
| **09/04/2013** |
|  |

Table des matières

[Présentation du projet 2](#_Toc353216733)

[Analyse des besoins 3](#_Toc353216734)

[Android et matériel 3](#_Toc353216735)

[Fonctionnalités attendues 4](#_Toc353216736)

[Ajout/suppression d’une catégorie 4](#_Toc353216737)

[Ajout/suppression d’une tâche 4](#_Toc353216738)

[Valider/invalider une tâche 4](#_Toc353216739)

[Visualiser les catégories et tâches 4](#_Toc353216740)

[Scénario d’utilisation 4](#_Toc353216741)

[But de l’application 4](#_Toc353216742)

[Scénario principal 4](#_Toc353216743)

[Story-board 4](#_Toc353216744)

[Interface Homme Machine 6](#_Toc353216745)

[Ajout/suppression d’une catégorie 6](#_Toc353216746)

[Ajout/suppression d’une tâche 6](#_Toc353216747)

[Valider/invalider une tâche 6](#_Toc353216748)

[Visualiser les catégories et tâches 6](#_Toc353216749)

[Développement Android 7](#_Toc353216750)

[Principale.java et cycle de vie 7](#_Toc353216751)

[Le cœur 7](#_Toc353216752)

[La persistance 7](#_Toc353216753)

[Extensions et conclusion 8](#_Toc353216754)

# Présentation du projet

Le but de ce projet est de créer un gestionnaire léger de tâches pour la plateforme mobile Android. « Léger » signifie qu’on ne gèrera que des catégories contenant des tâches non ordonnées. Les fonctionnalités attendues sont l’ajout et la suppression de catégories, de tâches et la persistance des données.

# Analyse des besoins

L’application a pour but la gestion simplifiée de tâches pour Android. Les utilisateurs visés sont les possesseurs d’un Smartphone Android ayant besoin de prendre des notes sur ce qu’ils ont à faire (ex : liste de courses). On peut en déduire qu’il s’agit d’une population assez habituée à l’usage d’Android mais sans aisance particulière. L’application devra donc respecter les habitudes liées à l’usage d’Android et se devra de rester simple (guides pour l’utilisateur).

## Android et matériel

L’application n’étant pas particulièrement complexe et de ce fait peu gourmande, il nous semble judicieux de la développer pour une version moyenne d’Android, la 2.2 (Froyo) et donc pour les versions supérieures. Une version inférieure ne présente pas les avantages techniques de Froyo, notamment concernant les menus et ne pénalise pas beaucoup d’utilisateurs (moins de 2% des utilisateurs globaux sont en 2.1 ou inférieur).

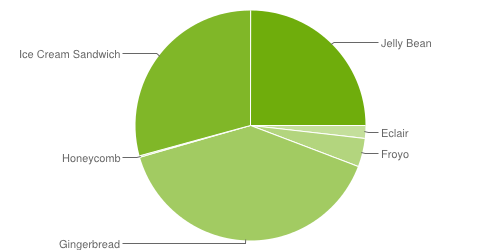


Figure 1 - répartition des versions d'Android - http://developer.android.com/about/dashboards/index.html

Les appareils visés sont uniquement des Smartphones. Nous avons décidé de ne pas supporter les tablettes (même si dans la plupart des cas, l’application fonctionnera), car l’application n’a pas vraiment de sens pour ce genre de support. En effet, il s’agit d’actions ponctuelles qui se font entre deux activités, dans des cadres très variés (au supermarché, en bricolant, en faisant ses devoirs…). De ce fait, l’usage d’une tablette n’est pas le support préféré des utilisateurs et ne se révèle pas pratique. La tablette n’aura donc pas d’interface particulière.

Concernant les autorisations, l’application n’aura besoin de rien, elle fonctionne sans aucune en particulier.

## Fonctionnalités attendues

Voici la liste des fonctionnalités de l’application. Nous verrons ensuite en détail l’interface associée à ces actions.

### Ajout/suppression d’une catégorie

L’utilisateur sera capable de créer et de supprimer des catégories. Une catégorie est un conteneur pour les tâches. Chaque catégorie aura un nom unique.

### Ajout/suppression d’une tâche

L’utilisateur sera capable de créer et de supprimer des tâches. Une tâche est contenue dans une catégorie et son nom n’est pas unique.

### Valider/invalider une tâche

Une tâche aura deux états possibles, validé et invalidé, et l’utilisateur pourra changer simplement cet état.

### Visualiser les catégories et tâches

L’utilisateur pourra consulter les tâches. Par défaut, l’affichage est celui du dernier lancement. L’interface sera composée d’une liste de catégories, les unes sous les autres. Chaque catégorie pourra être étendue pour dévoiler les tâches associées ou rétractée pour les cacher.

## Scénario d’utilisation

### But de l’application

Cette application vise les utilisateurs d’Android qui cherchent à prendre des notes sur leurs actions à venir afin de ne pas les oublier. Elle leur permettra en outre de suivre l’avancement desdites actions.

### Scénario principal

On prendra l’exemple d’une liste de course. L’utilisateur prend son Smartphone Android et lance l’application mobile Degreasor©. Il crée une catégorie nommée « liste de course » et lui ajoute diverses tâches correspondant aux éléments de la liste de course. Une fois dans le magasin, l’utilisateur sort de nouveau son téléphone et la liste est toujours présente. Il peut alors cocher au fur et à mesure les éléments qu’il a déjà mis dans son caddie et voir son avancement. Il est sûr de ne rien oublier et il peut réutiliser cette liste pour les prochains achats.

### Story-board



# Interface Homme Machine

Nous verrons dans cette section les choix effectués quant à l’interface graphique, en partant sur la base de ce qui a déjà été convenu plus tôt.

### Ajout/suppression d’une catégorie

L’utilisateur pourra ajouter une catégorie à l’aide du bouton menu de son Smartphone. Le menu ouvert lui proposera l’option « new category ». Une fois ce bouton cliqué, l’utilisateur pourra donner le nom de sa catégorie ou annuler la création. Une vérification est faite au moment de la validation afin de ne pas avoir deux catégories de même nom.

La suppression se fera par clic long sur la catégorie. Le menu proposera une action « delete ». Attention, par souci de rapidité d’exécution, il n’y a pas de confirmation sur cette action! Cette confirmation présente à l’origine était vraiment handicapante lors de grosses modifications de données. La suppression d’une catégorie entraine la suppression de son contenu.

### Ajout/suppression d’une tâche

L’utilisateur pourra ajouter et supprimer des tâches. L’ajout se fait par clic long sur la catégorie qui la contiendra. Le menu proposera une action « new task ». Une boite de dialogue s’ouvre et permet à l’utilisateur de saisir le nom de la tâche. Il peut ensuite valider la création ou l’annuler.

La suppression se fait par clic long sur la tâche. Le menu propose l’option « delete ». DE la même façon que pour les catégories, cette action ne propose pas de confirmation.

### Valider/invalider une tâche

L’utilisateur pourra valider et invalider une tâche en cliquant dessus. Un indicateur visuel sur la partie gauche de la ligne indique dans quel état est la tâche : cochée si la tâche est validée, décochée sinon.

### Visualiser les catégories et tâches

L’utilisateur pourra faire défiler l’affichage de ses catégories et tâches par un glissement vertical. Les catégories sont représentées par des listes extensibles et les tâches par des items de ces listes.

La catégorie possède un indicateur visuel qui indique si la liste est déroulée (flèche vers le haut) ou non (flèche vers le bas). Les signaux sont ceux d’Android. La catégorie affiche son nom et une barre de progression en arrière-plan. Cette barre progresse en largeur en fonction du nombre de tâches validées par rapport au nombre de tâches total. Cette progression est associée à une couleur qui va du rouge vers le vert.

Le bouton menu propose une option qui permet de réduire automatiquement toutes les listes.

# Développement Android

Le choix de l’API pour le développement de cette application est donc l’API 8 pour Froyo.

## Principale.java et cycle de vie

Il s’agit de la classe principale qui sous-classe Activity. Il s’agit donc du point d’accroche de l’application. Il s’agit aussi de la seule classe d’affichage, il n’y a pas d’Intent.

L’application à un fonctionnement très simple qui ne nécessite pas d’approfondir le cycle de vie des applications Android en général. On implémente les fonctions onPause() et onResume() pour la sauvegarde et la restauration de l’état du téléphone.

Nous avons implémenté cette méthode à la fin du développement, l’enregistrement se faisant à chaque modification auparavant. Nous supposions qu’il y aurait des problèmes lors du passage de l’application en onStop() mais nous n’avons pas réussi à produire le bug, les données étaient toujours conformes. Par souci de précaution, nous avons quand même centralisé la sauvegarde et la restauration dans onPause() et onResume().

Cela a posé un gros problème au niveau des listeners, les références mémoire étant écrasées lors de la restauration. Nous avons donc du déplacer une partie du code de onCreate() dans onResume() afin de créer ces références au moment de la restauration.

## Le cœur

La structure interne des listes est la suivante.

La classe principale possède une instance de CategoryManager. CategoryManager possède une liste de Category, possédant à son tour une liste de Task.

Pour la partie graphique, on utilise une ExpandableListView interfacée avec un BaseExpandableListAdapter. Il suffisait d’implémenter les fonctions de la superclasse pour que ça fonctionne. Chaque modification est actualisée graphiquement par un appel à notifyDataSetChanged().

Nous avons surchargé le textEdit pour qu’il n’accepte pas les retours à la ligne, ceci pour faciliter la sauvegarde des données.

On se rend compte de la facilité d’usage d’Android, nous n’avons eu besoin de créer que quelques listeners pour la gestion de clic et clic long (onCheckedChangeListener et onClickListener) et d’implémenter les fonctions des superclasses, qui se résumaient dans la plupart des cas à une ligne ou deux.

## La persistance

Nous utilisons la technique des préférences partagées. On stocke nos informations dans un fichier. La seule information présente est appelée « main » et représente la sérialisation de CategoryManager. La restauration se fait de la même façon.

On ne crée pas un fichier spécifique, mais on utilise le fichier par défaut. On ne connait pas de ce fait le mode de fichier, mais on suppose qu’il est privé.

# Extensions et conclusion

A l’utilisation, nous nous sommes rendu compte que la position de l’utilisateur dans la liste était perdue lors d’un passage en onPause(). Sur des petites listes, ce n’est pas un problème, mais nous pensons que l’application gagnerait en ergonomie si la position était sauvegardée.

Pour conclure, on a ici une technologie très aboutie pour les développeurs qui permet de faire rapidement des applications complexes et les distribuer au monde entier.

Lien vers l’application sur le play store : <https://play.google.com/store/search?q=Mobile+Degreasor&c=apps>