Projet final : Rapport Final

SEG2505 – Introduction au Génie Logiciel
Automne 2017
École de Génie Électrique et Science Informatique

Travail fait par
Oliver Benning - 7798804
Walid Bounouar - 8917791
Thomas Charette - 8475447
Jean-Gabriel Gaudreault - 8865533
Vincent Harvey - 8780303

Présenté à
Professeur : Miguel Garzón

6 Décembre 2017
Université d'Ottawa

Introduction

Ce présent document présente les contributions de chaque membre de l'équipe au projet, les corrections apportées aux livrables précédents, les exigences finales, les diagrammes de séquence et d'état modifiés, les interfaces utilisateurs de l'application et les leçons apprises durant le projet.

De plus, vous pouvez avoir accès à notre repo GitHub avec le lien suivant : https://github.com/vharvey80/SEG2505_FinalProject

Contributions

Membre	*Walid*	*Thomas*	*Vincent*	*Oliver*	*Jean-Gabriel*
Livrable 1 ● Création des exigences ● Création des cas d'utilisations	**20 %**	**20 %**	**20 %**	**20 %**	**20 %**
Livrable 2 ● Création du diagramme de classe UML ● Création des diagrammes de séquences UML ● Création du diagramme de machine d'état UML	**20 %**	**20 %**	**20 %**	**20 %**	**20 %**
Livrable 3 ● Rédaction du rapport	**20 %**	**20 %**	**20 %**	**20 %**	**20 %**
Livrable 4 ● Rédaction du rapport	**20%**	**20%**	**20%**	**20%**	**20%**

Commentaires supplémentaires

Pas de commentaires

Corrections

Livrable	Modifications/Corrections apportées
1	Nous avons remplacé l'exigence par rapport à la récurrence d'une tâche avec l'exigence par rapport au calendrier
1	Modifié l'exigence fonctionnelle de créer et assigner une tâche à 7 clicks. 6 clicks est faisable, mais 7 couvre tous les cas.
1	Modifié l'exigence fonctionnelle de changer l'utilisateur à 5 clicks. 4 clicks est faisable, mais 5 couvre tous les cas.
1	Modifié l'exigence non-fonctionnelle de l'API de 26+ à 25.
2	Mise à jour de certains attributs, méthodes et associations.
2	Mise à jour des diagramme de séquence et du diagramme d'état
3	Ajout de captures d'écran pour l'interface "Calendar"
3	Modification du diagramme de séquence pour "Change user"
3	Raffinement général de toutes les vues
...	

Exigences

Exigences fonctionnelles

- ✓ Le système doit afficher toutes les tâches actuellement non complétées sur un même écran.
- ✓ Le système doit permettre à un utilisateur de seulement voir les tâches qui lui sont assignées.
- ✓ Le système doit permettre aux usagers de changer de profil utilisateur sur un même appareil.
- ✓ Le système doit offrir aux utilisateurs des profils “parent” et des profils “enfant”.
- ✓ Le système doit permettre de créer de nouveaux profils utilisateurs.
- ✓ Le système doit permettre à tout utilisateur d’ajouter une tâche au système.
- ✓ Le système doit permettre à un utilisateur de mettre une date limite sur une tâche.
- ✓ Le système doit permettre à un utilisateur d’ajouter des notes à une tâche.
- ✓ Le système doit permettre à un utilisateur d’ajouter de l’équipement requis à une tâche.
- ✓ Le système doit permettre à un utilisateur de voir toute l’information reliée à une tâche sur une même page (ou fenêtre “pop-up”).
- ✓ Le système doit offrir un calendrier permettant de voir les tâches dues à une certaine date.
- ✓ Le système doit permettre aux utilisateurs “parents” seulement d’ajouter des récompenses à une tâche.
- ✓ Le système doit permettre aux utilisateurs “parents” seulement de modifier les informations d’un tâche.
- ✓ Le système doit permettre aux utilisateurs “parents” seulement d’assigner une tâche, ou à un enfant d’assigner une tâche à soi même.
- ✓ Le système doit demander une confirmation de l’utilisateur lors de l’allocation d’une tâche.
- ✓ Le système doit permettre à un utilisateur de voir tous les autres utilisateurs sur un seul écran.
- ✓ Le système doit permettre à l’utilisateur de voir tous les éléments de la liste de courses sur un seul écran.
- ✓ Le système doit permettre à un utilisateur d’ajouter des objets dans la liste de courses.
- ✓ Le système doit mettre à jour automatiquement le garde manger à partir des achats fait dans la liste de courses.
- ✓ Le système doit permettre à l’utilisateur parent de supprimer une tâche.

Exigences non-fonctionnelles

- ✓ Le système doit enregistrer les tâches en utilisant une base de donnée Firebase.
- ✓ Le système doit être compatible avec le Android API 25.
- ✓ Le système doit toujours permettre d’accéder au menu principal en dessous de 4 clics.
- ✓ Le système doit permettre à un utilisateur de créer et d'allouer une tâche en 7 clics ou moins (à partir de l’écran des tâches).
- ✓ L’application doit n’avoir aucun coût de développement.
- ✓ Le système doit permettre de changer d’utilisateur en 5 clics ou moins.

Design UML

Diagramme de classes

Notes:

- Nous avons uniquement fait un diagramme de classe pour le modèle. Nous ne considérons pas la MainActivity ou toute autre composante fonctionnelle comme une partie de modèle. Certaines méthodes pourrait donc sembler absentes par rapport aux autre diagrammes, mais ceci est parce qu'elles n'appartiennent pas à une classe de modèle.
- Les associations de Task à User sont exclue de la base de donnée, car nous utilisons une base de donnée non-relationnelle et garder une telle relation poserait problème. Donc au niveau de la programmation, l'association est bidirectionnelle, mais au niveau de la base de données, elle est unidirectionnelle.

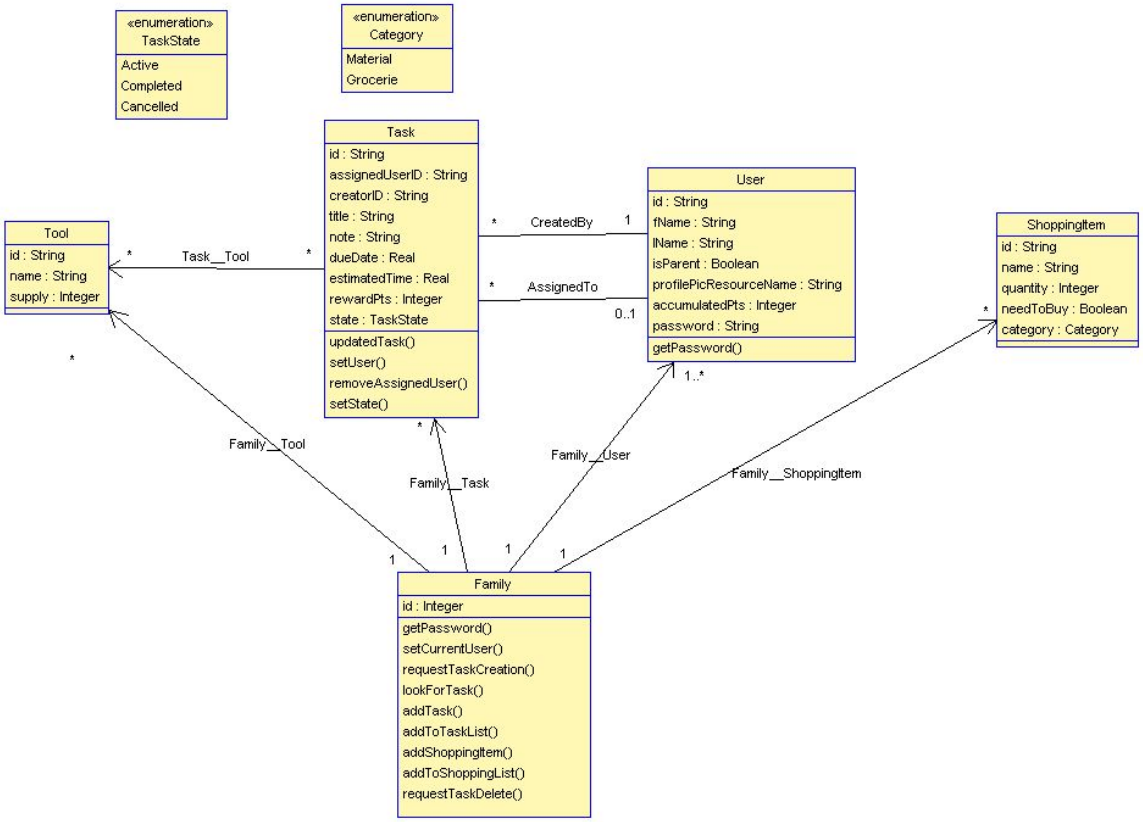


Diagramme de séquence “Create Task”

Note : Cette séquence est exclusivement pour créer une tâche. En réalité, dans notre système, la création d’une tâche s’accompagne toujours d’une mise à jour pour ajouter des Tools.

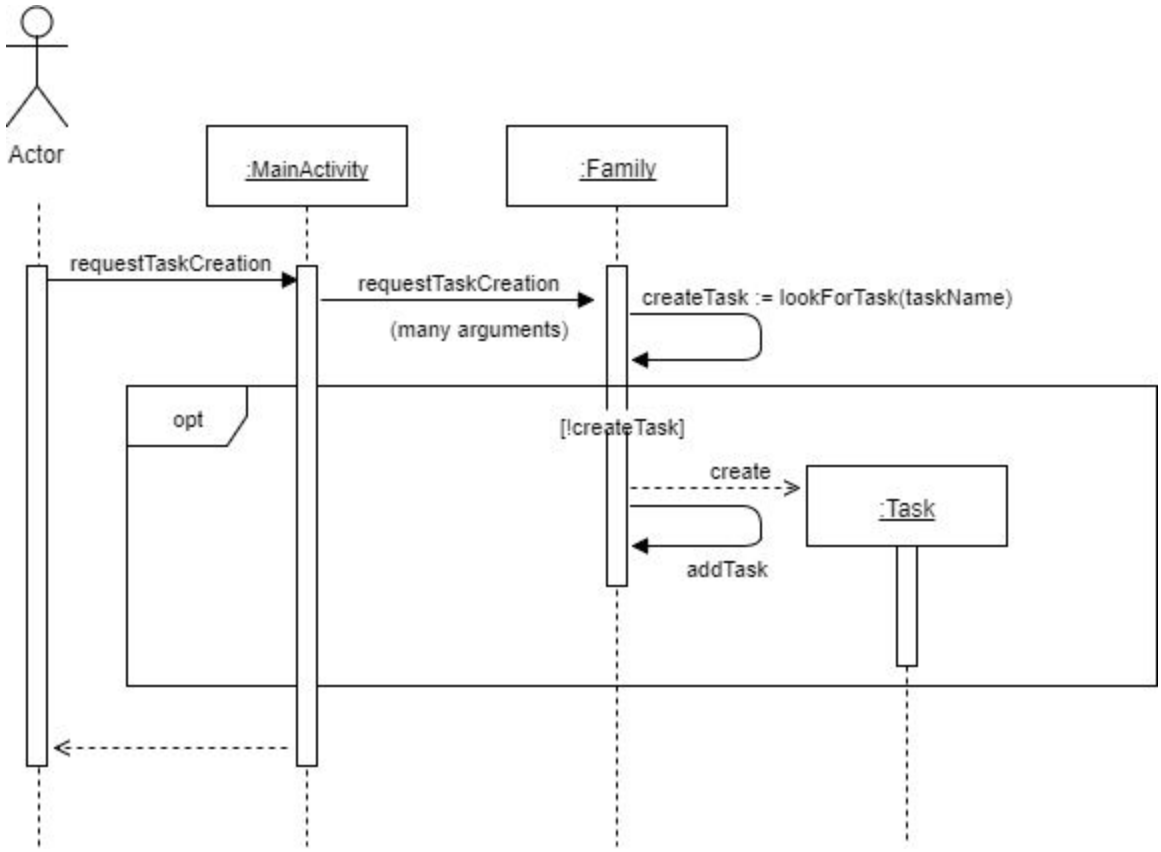


Diagramme de séquence “Change user”

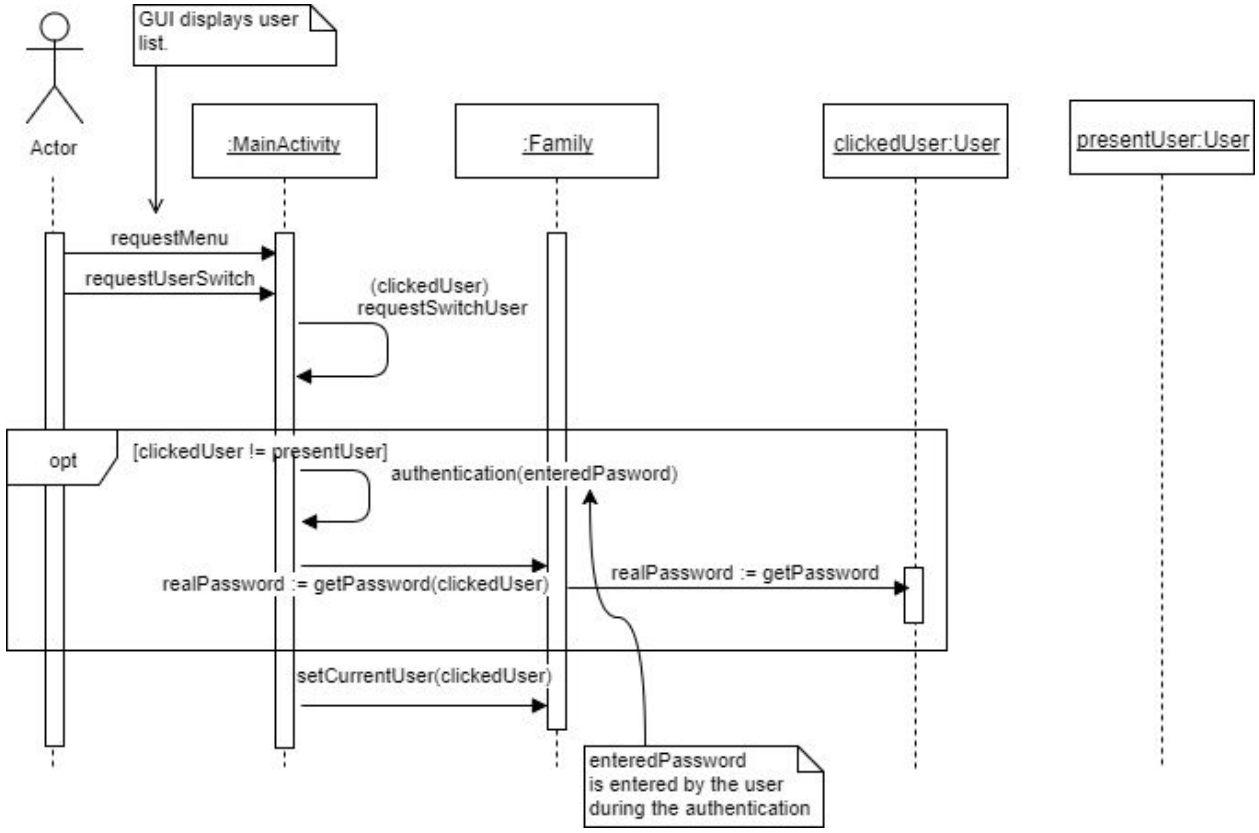


Diagramme de séquence “Add shopping item”

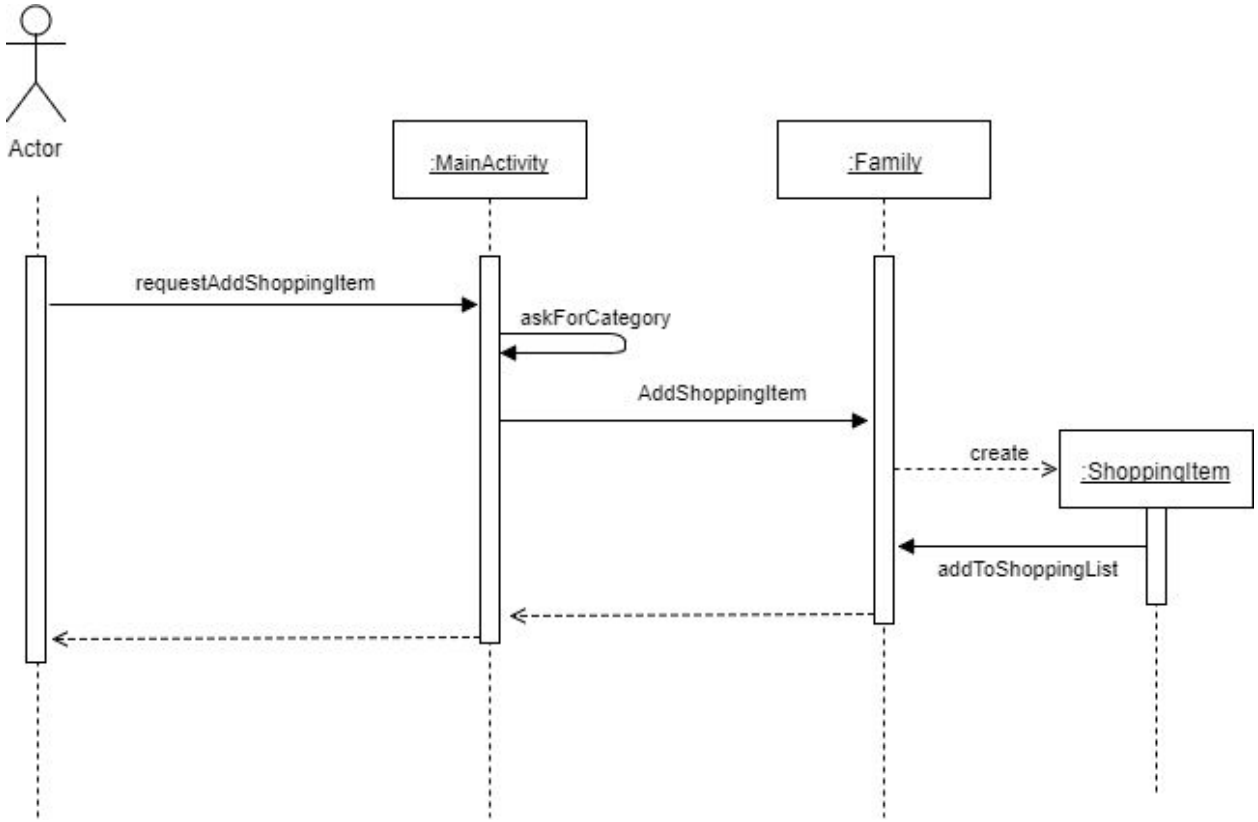
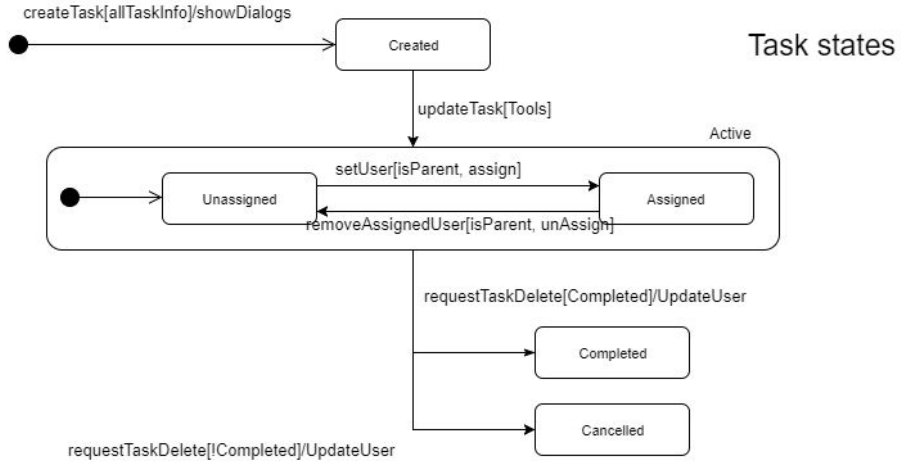


Diagramme d’état d’une “Task”

Note :

- Il est techniquement possible de sauter la transition “updateTask” si l’utilisateur appui cancel au bon moment. Cependant, la tâche est tout de même considérée maintenant Active (commence “unassigned”).
- La méthode “requestTaskDelete” appartient à la classe Family
- Svp notez que les méthodes de transition sont générales. La méthode exacte pour changer d’état est simplement setState().



1) [allTaskInfo] : Toutes les informations nécessaire pour la création d'une tâche (title, dueDate, etc.)

2) [isParent] : Variable qui nous dit si le user déclenchant les méthodes pour assigner et désassigner est un parent.

3) [Tools] : Variable qui nous donne si la task doit être updater pour ajouter des tools.

4) [assign] : Variable qui nous dit si la task a été assignée à quelqu'un.

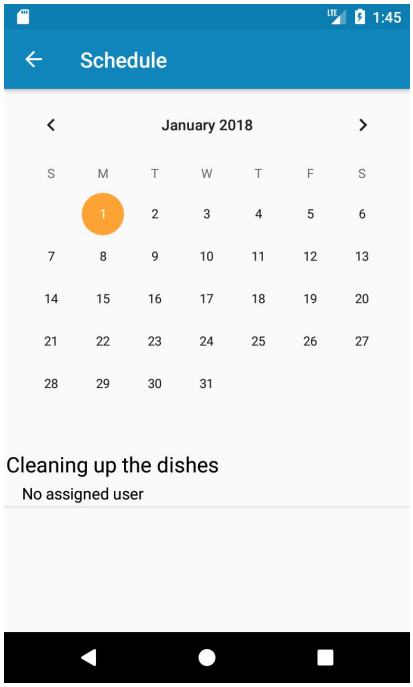
5) [unAssign] : Variable qui nous dit si la task a été désassignée de quelqu'un.

6) [Completed] : Variable qui nous dit si un user veut changer le state de la task a "complété" (Traitement presque identique).

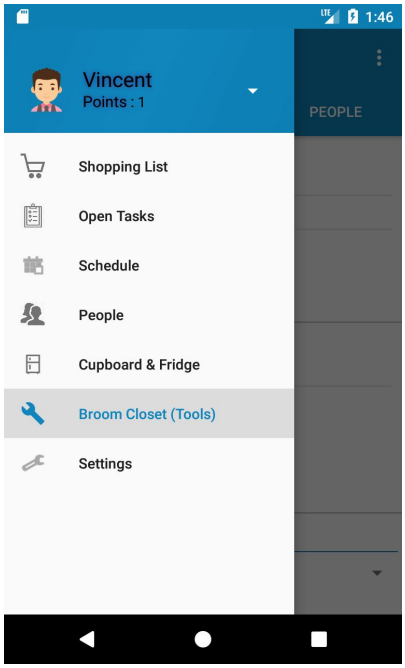
7) [!Completed] : Variable qui nous dit si un user veut changer le state de la task a "Cancelled" lors de l'action delete ou cancel (Traitement presque identique).

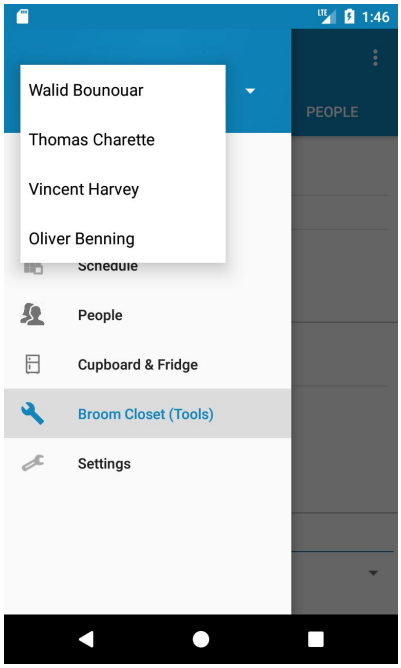
Interface Utilisateur

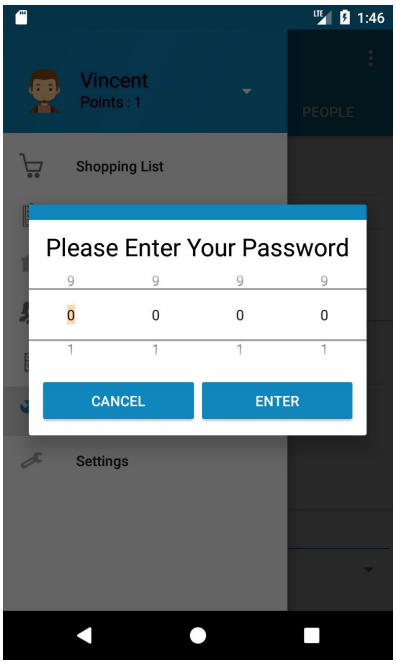
Calendrier

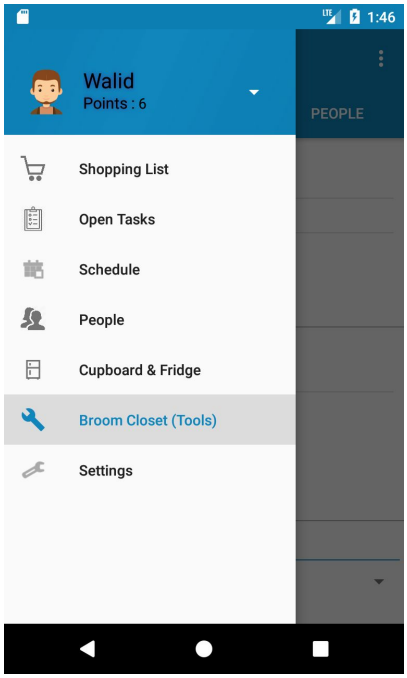


Menu de navigation et changement d’utilisateur

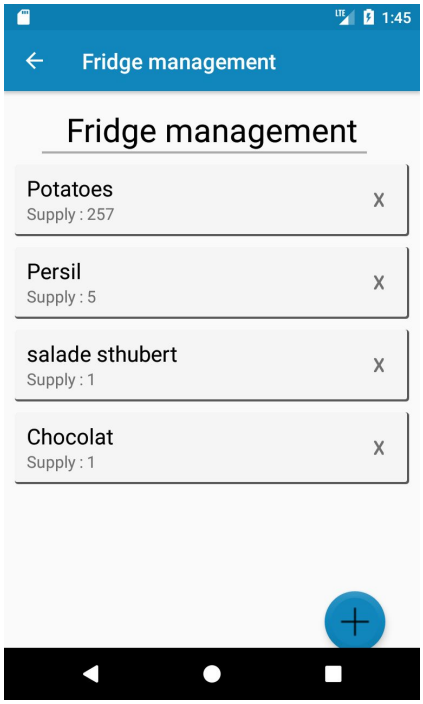


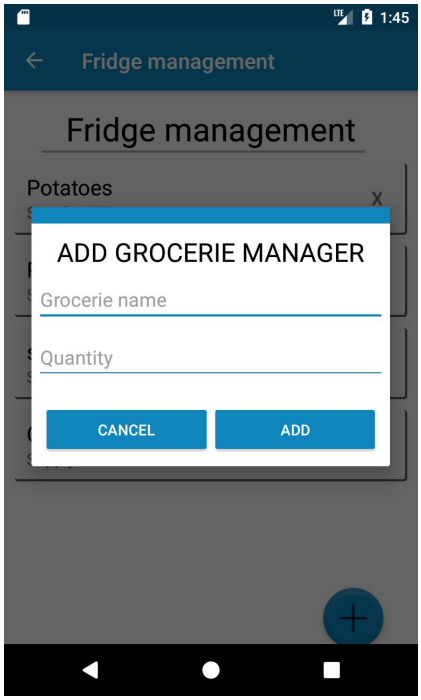




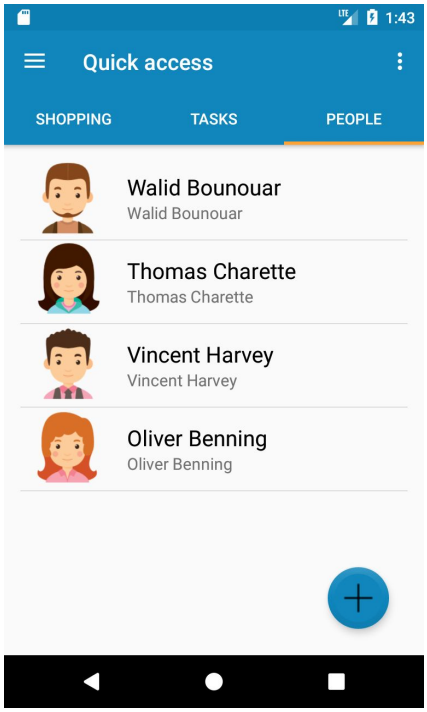


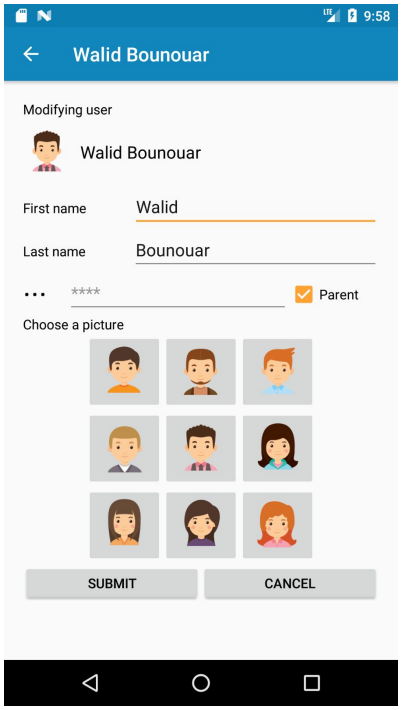
Frigidaire

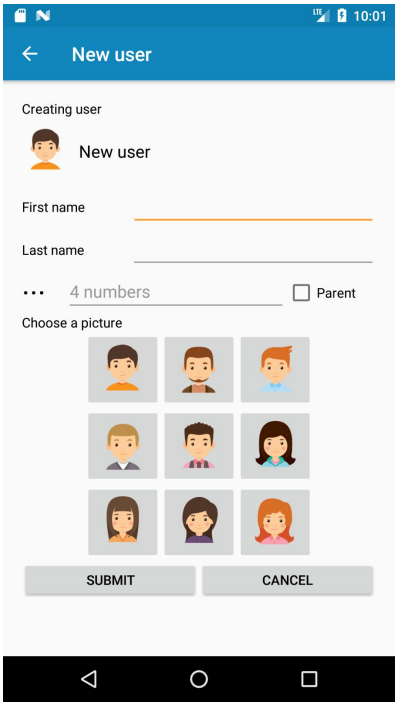

1:45
Fridge management
Fridge management
Potatoes
Supply : 257
X
Persil
Supply : 5
X
salade sthubert
Supply : 1
X
Chocolat
Supply : 1
X

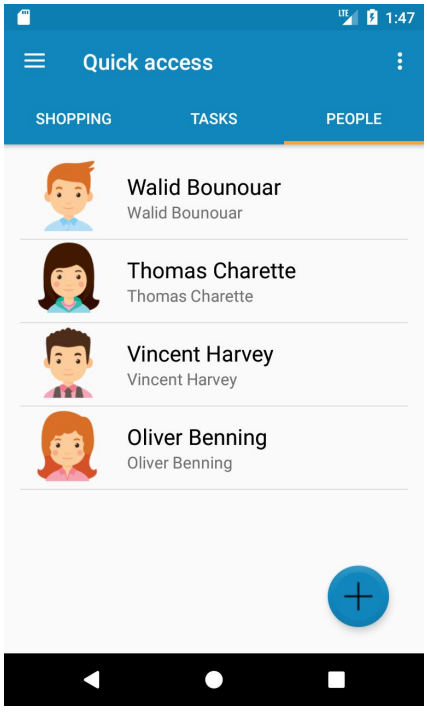

1:45
Fridge management
Fridge management
Potatoes
X
ADD GROCERIE MANAGER
Grocerie name
Quantity
CANCEL
ADD

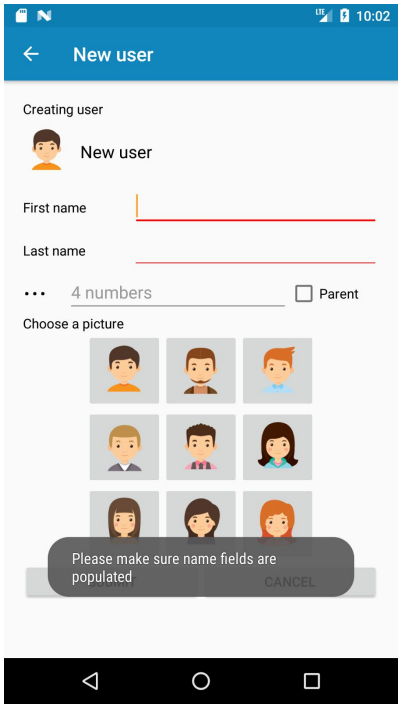
Liste d’utilisateurs, ajouter et modifier des utilisateurs



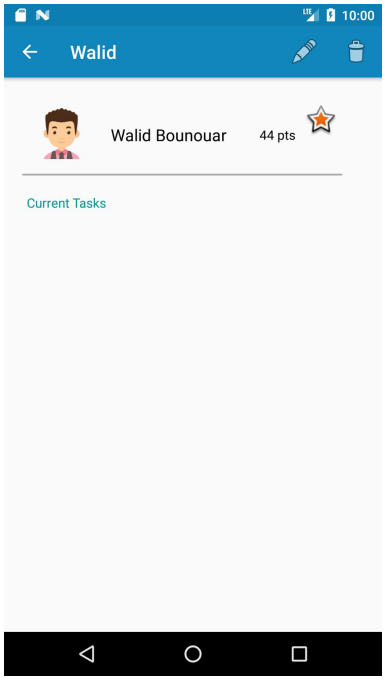


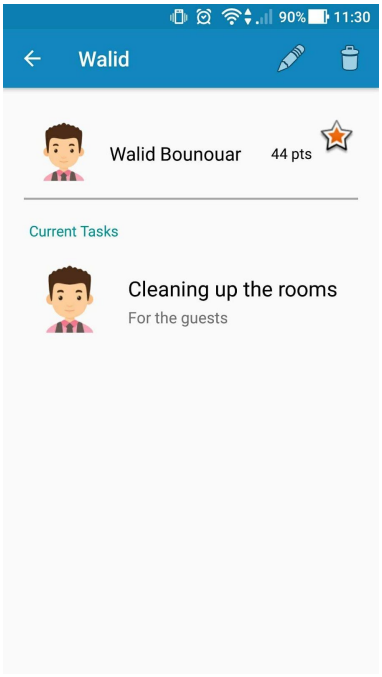




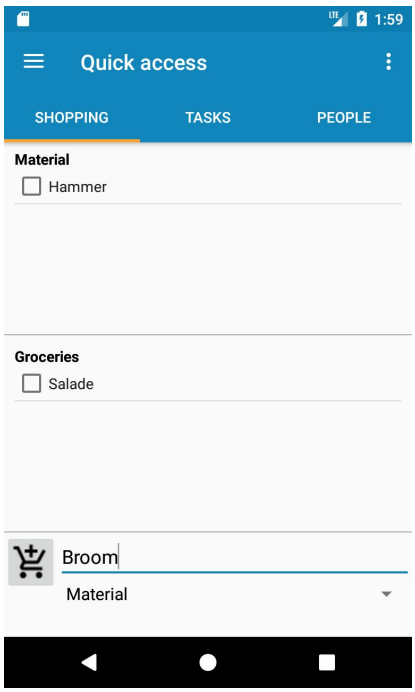


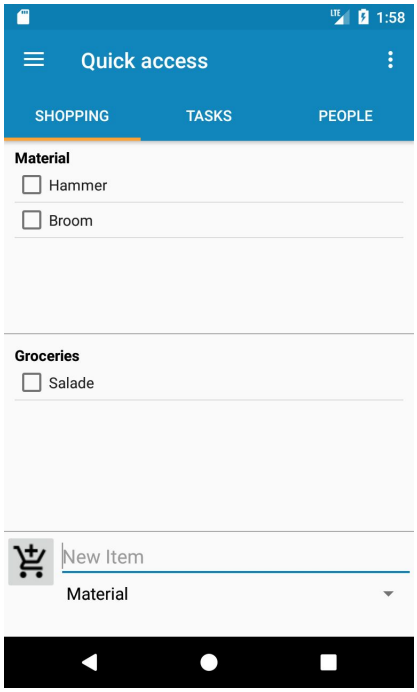
UserDetail

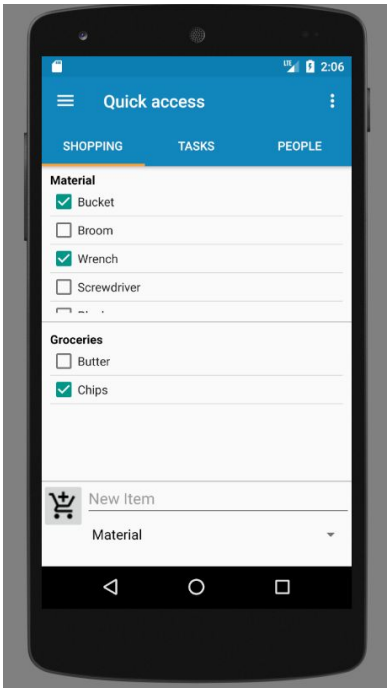

10:00
Walid
Walid Bounouar
44 pts
Current Tasks


90% 11:30
Walid
Walid Bounouar
44 pts
Current Tasks
Cleaning up the rooms
For the guests

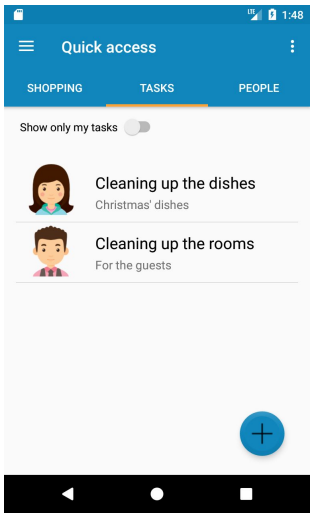
Liste de courses

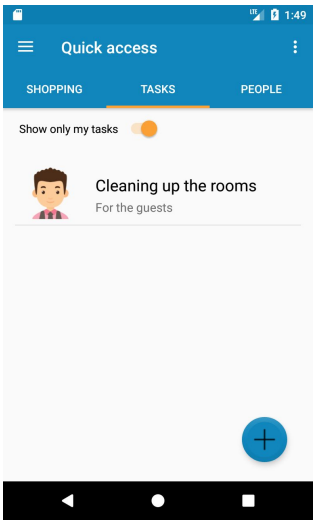


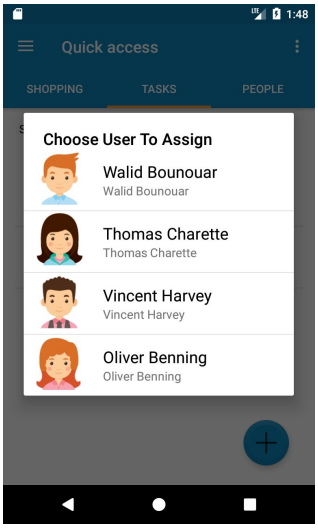


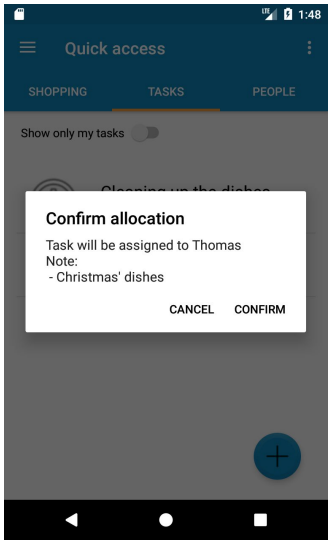


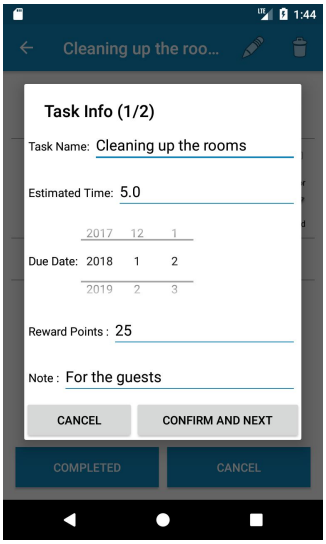
Liste de tâches, ajouter, modifier, annuler des tâches

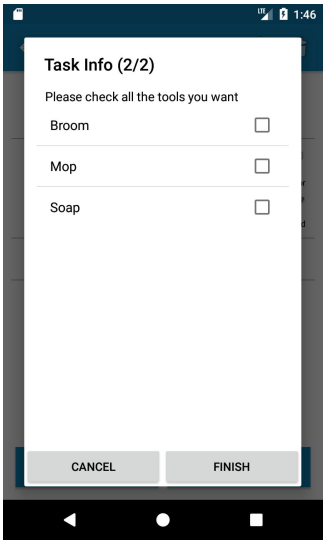


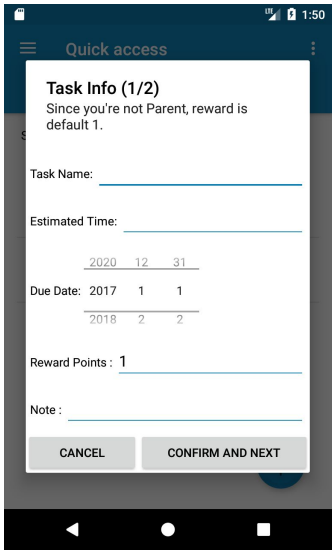




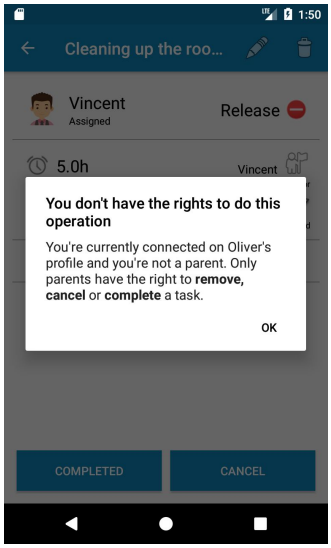


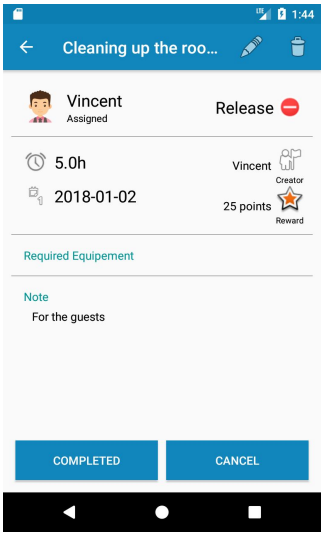


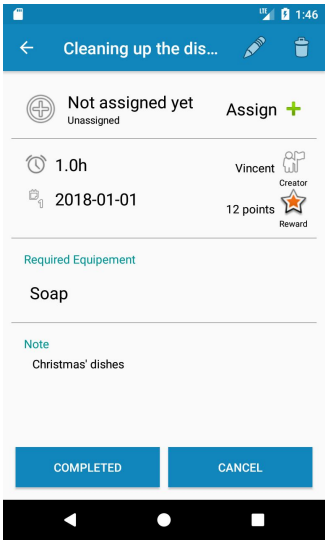


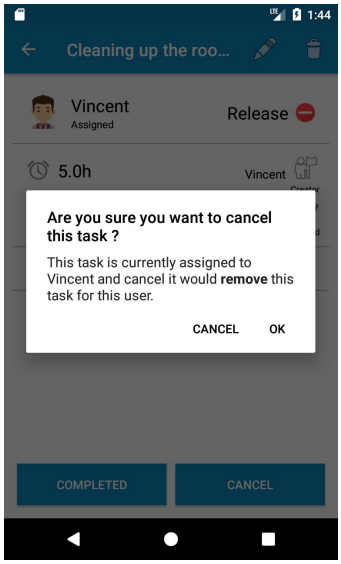


Parent rights, task details

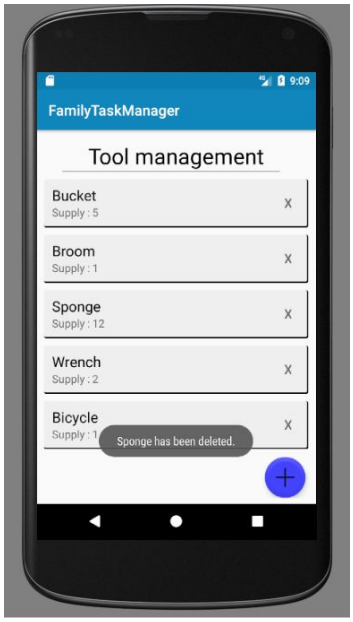


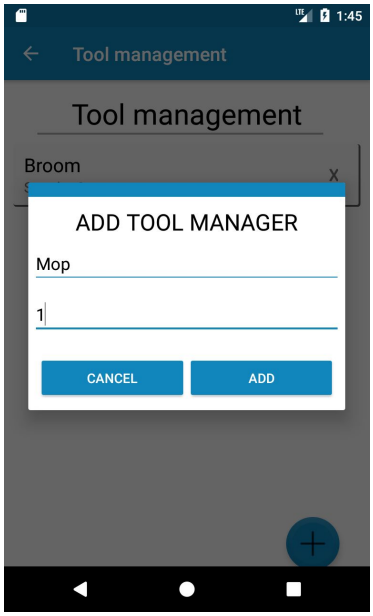






Tools





Leçons apprises/Suggestions

Au cours de la création de cette application, plusieurs choses ont attirées notre attention. Entre autres, nous avons pu s'apercevoir de l'importance d'avoir une bonne planification et des diagrammes de classe bien détaillés. Ceux-ci nous ont permis dès le début de savoir quelles classes seront incluses dans notre projet ainsi que les relations entre elles.

La communication a également pris une part plus qu'importante dans notre projet, en ayant créé un groupe de discussion, nous avons pu partager plus aisément nos craintes, avancement et problèmes sur le projet et ainsi informé rapidement toute l'équipe de l'état courant du projet. Nous avons également organisé des rencontres hebdomadaires avec tout le groupe pour pouvoir discuter en personne de notre vision de l'implémentation.

De plus, en ayant établis des exigences biens définies dès le début, cela nous a permis d'avoir des buts bien précis à atteindre et ainsi savoir précisément quand devoir s'arrêter et quoi implémenter.

Nous avons aussi réalisé l'importance de faire une recherche approfondie sur les technologies avant de les utiliser. Par exemple, notre système était modélisé avec beaucoup de relations entre les objets et nous avons choisi une technologie de base de données non relationnelle, ce qui a mené à beaucoup de problèmes.