



uOttawa

SITE

SEG 2505- Introduction au Génie Logiciel

Automne 2017

Professeur: Miguel Alejandro Garzón

Assistants: Cédric Clément & Abdelkarim Sellami

Rapport final du projet Android:

Gestionnaire des tâches ménagères

Présenté par:

(Groupe MaJES)

Colombe Schellariël Tolokin, #8002350

Stéphane Rurema, #8862803

Jérémie Tsai, #8227028

Maxime Côté-Gagné, #8851539

Date de la soumission: 3 Décembre 2017

Tables des matières

Introduction.....	3
Contribution.....	3
Corrections.....	3
Exigences.....	4
Design UML.....	5
Captures d'écran de l'interface.....	10
Leçons et suggestions.....	12

1. Introduction

Le Gestionnaire des Tâches Ménagères est une application Android mobile pour aider à gérer les différentes interactions quotidiennes auxquelles des familles ou colocataires font face quotidiennement. Elle permet de :

- Ajouter / éditer / supprimer des tâches
- Ajouter / éditer / supprimer des utilisateurs
- Accorder des récompenses selon le nombre de tâches effectuées
- Donner des commentaires rapport aux tâches
- Voir sa liste des tâches
- Voir à qui chaque tâche est assignée

Ce document vous donnera une vue des différentes livrables pour l'application mobile et les captures des interfaces (UI) pour donner du fonctionnement de l'application.

2. Contributions

Le tableau ci-dessus indique les rôles et les contributions des membres de notre groupe pour chaque livrable.

Membres	Livrable 1	Livrable 2	Livrable 3	Livrable 4
Colombe Schellariël Tolokin	25%	25%	25%	20%
Stéphane Rurema	25%	25%	25%	20%
Jérémie Tsai	25%	25%	25%	30%
Maxime Côté-Gagné	25%	25%	25%	30%

3. Corrections

Les différentes modifications/corrections apportées à nos livrables précédents sont les suivantes :

Livrable	Modifications/ Corrections
1	Modification de la 4 ^{ème} exigence non-fonctionnelle
2	Modification des diagrammes de séquences

4. Exigences (du premier livrable)

Nous avons eu à déterminer des exigences pour notre application. Elles sont aussi bien fonctionnelles que non-fonctionnelles.

4.1 Fonctionnelles

- Le système doit être connecté à l'internet pour accéder à une base de données.
- Le système doit permettre à l'utilisateur d'ajouter des tâches.
- Le système doit permettre l'ajout d'utilisateurs.
- Le système doit permettre à l'utilisateur la suppression des tâches.
- Le système doit tenir du nombre de tâches réalisées par utilisateur.
- Le système doit avoir une date limite pour chaque tâche.
- Le système doit permettre à l'utilisateur d'attacher des commentaires aux tâches.
- Le système doit permettre l'ajout d'une image pour un utilisateur.
- Le système doit permettre à l'utilisateur de changer ses informations.
- Le système peut stocker une liste des tâches terminées.
- Le système peut permettre l'ajout d'une description qui explique ce qui doit être réalisé lors de la tâche.
- Le système peut afficher une liste journalière des tâches.
- Le système peut permettre l'ajout du temps nécessaire à l'accomplissement d'une tâche.
- Le système doit permettre à l'utilisateur d'éditer une tâche.
- Le système peut permettre la suppression d'un utilisateur.
- Le système doit avoir une liste d'utilisateurs.
- Le système doit pouvoir envoyer des messages accessibles à tous les utilisateurs.
- Le système peut préciser la fréquence à laquelle une tâche doit être accomplie.
- Le système doit avoir une liste générale de tâches.
- Le système doit avoir une liste de tâches par utilisateurs.

4.2 Non-fonctionnelles

- Le système doit être implémenté à partir de l'application Android Studio.
- Le système doit être implémenté en suivant une méthodologie Agile.
- Le système doit être prêt à l'utilisation au plus tard le 6 décembre 2017.

- Le système doit être facile à apprendre et manipuler.
- Le système doit fonctionner sur les appareils dotés d'Android 4.4 et plus.

5. Design UML (du deuxième livrable)

5.1 Cas d'utilisation

5.1.1 1er cas

Nom: Créer une tâche

Acteurs: Utilisateur

Préconditions: L'utilisateur démarre l'application

Postconditions: La tâche est ajoutée à la liste de tâche

Utilisateur	Système
1.L'utilisateur démarre application	Affiche le menu d'accueil
2.L'utilisateur clique sur nouvelle tâche	Une nouvelle fenêtre de création pour une nouvelle tâche s'affiche demandant l'ajout d'un nom et d'une description optionnelle
3.L'utilisateur ajoute le nom de la tâche et sa description (explication, date limite...)	Une fenêtre de confirmation apparaît.
4.L'utilisateur confirme la création de la nouvelle tâche	La tâche est créée et s'affiche dans la liste de tâches.

5.1.2 2ème cas

Nom: Assigner une tâche

Acteurs: Utilisateur

Préconditions: L'utilisateur démarre l'application

Postconditions: La tâche est assignée à une personne

Utilisateur	Système
-------------	---------

1.L'utilisateur démarre l'application	Affiche le menu d'accueil
2.L'utilisateur clique sur une tâche	Montre l'information de la tâche
3.L'utilisateur choisit à qui alloué la tâche	Une fenêtre de confirmation apparaît.
4.L'utilisateur confirme l'assignation	La tâche est maintenant assignée à un utilisateur

5.1.3 3ème cas

Nom: Ajouter un utilisateur

Acteurs: Utilisateur

Préconditions: Le système affiche le menu d'accueil

Postconditions: L'utilisateur est ajouté à la liste d'utilisateurs

Utilisateur	Système
1.L'utilisateur démarre l'application	Affiche le menu d'accueil
2.L'utilisateur clique sur nouvel utilisateur	Une nouvelle fenêtre de création pour un nouvel utilisateur s'affiche demandant l'ajout d'un nom et de ses identifiants
3.L'utilisateur ajoute un nom au nouvel utilisateur ainsi que ses identifiants	Une fenêtre de confirmation apparaît.
4.L'utilisateur confirme la création d'un nouvel utilisateur	L'utilisateur est créé et s'affiche dans la liste d'utilisateurs

5.2 Diagramme de classe UML (modèle du domaine)

Diagramme de classe représentant notre gestionnaire de tâche:

Une classe Utilisateur ayant un nom et une image peut être soit un enfant ou un parent.

L'Enfant peut avoir des récompenses données par un Parent.

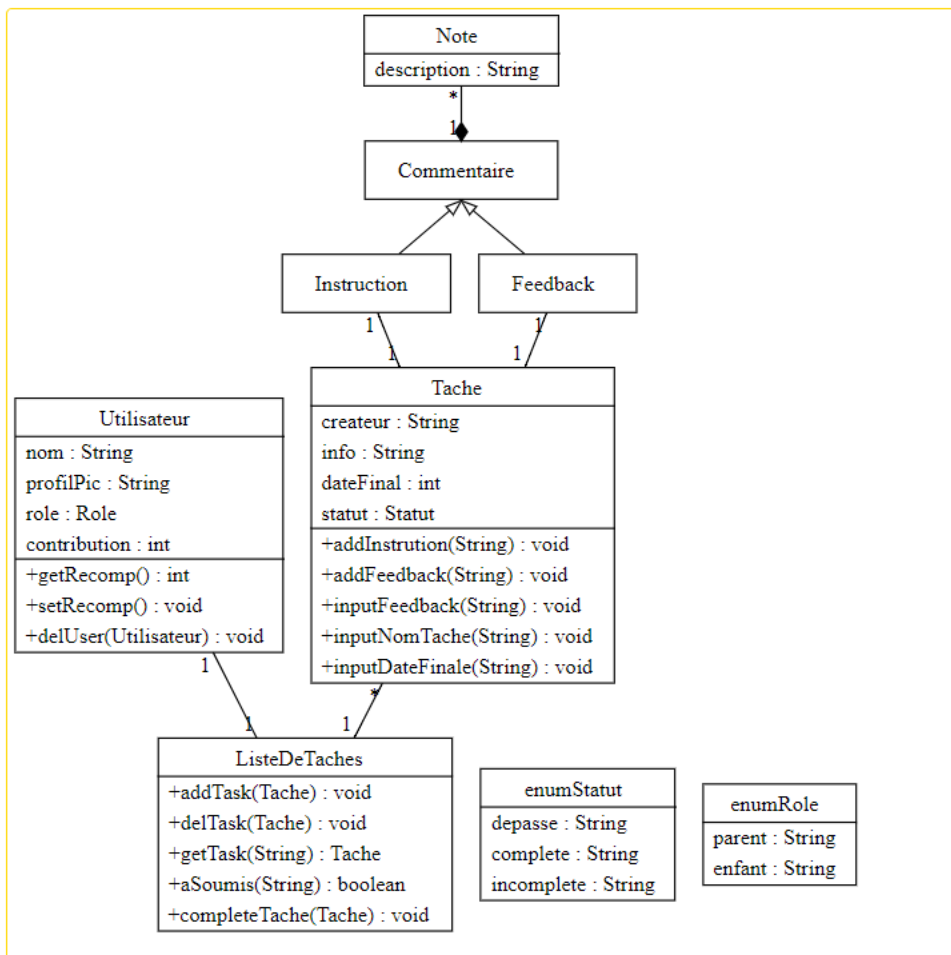
Elle peut aussi savoir si une tâche est complète ou incomplète ou encore en retard.

La classe ListeDeTaches permet d'ajouter, de supprimer ou d'obtenir une tâche (réfère à la prochaine classe). Elle peut aussi mettre à jour son statut.

La classe Tache permet d'avoir un son créateur, sa description et une date qui lui est dû.

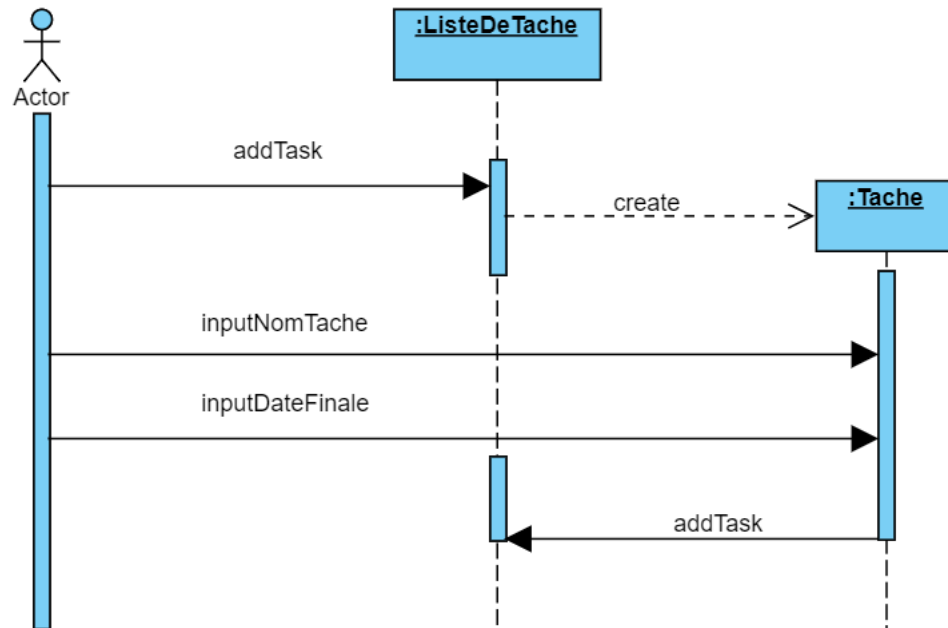
Elle a aussi un Statut(énumération) qui est soit Incomplet, soit complété, soit passé.

L'interface commentaire est ici pour ajouter ou supprimer ou obtenir une note sur une tâche.

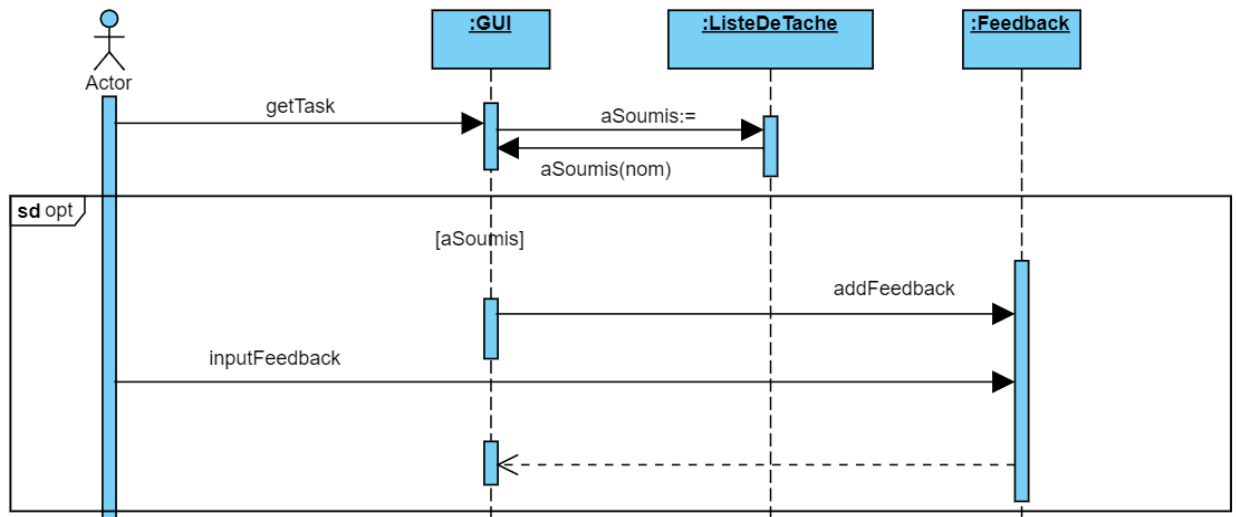


5.3 Diagrammes de séquences UML

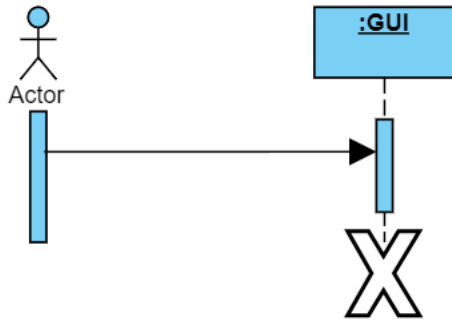
5.3.1 Diagramme de séquence #1 : Séquence pour ajouter une tâche



5.3.2 Diagramme de séquence #2 : Séquence pour ajouter du feedback(commentaire) à une tâche

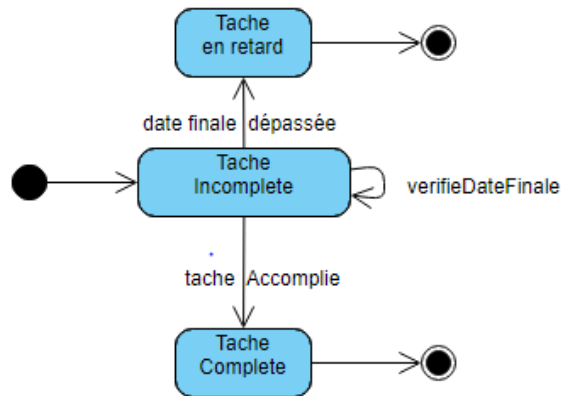


5.3.3 Diagramme de séquence #3 : Séquence pour supprimer un utilisateur.

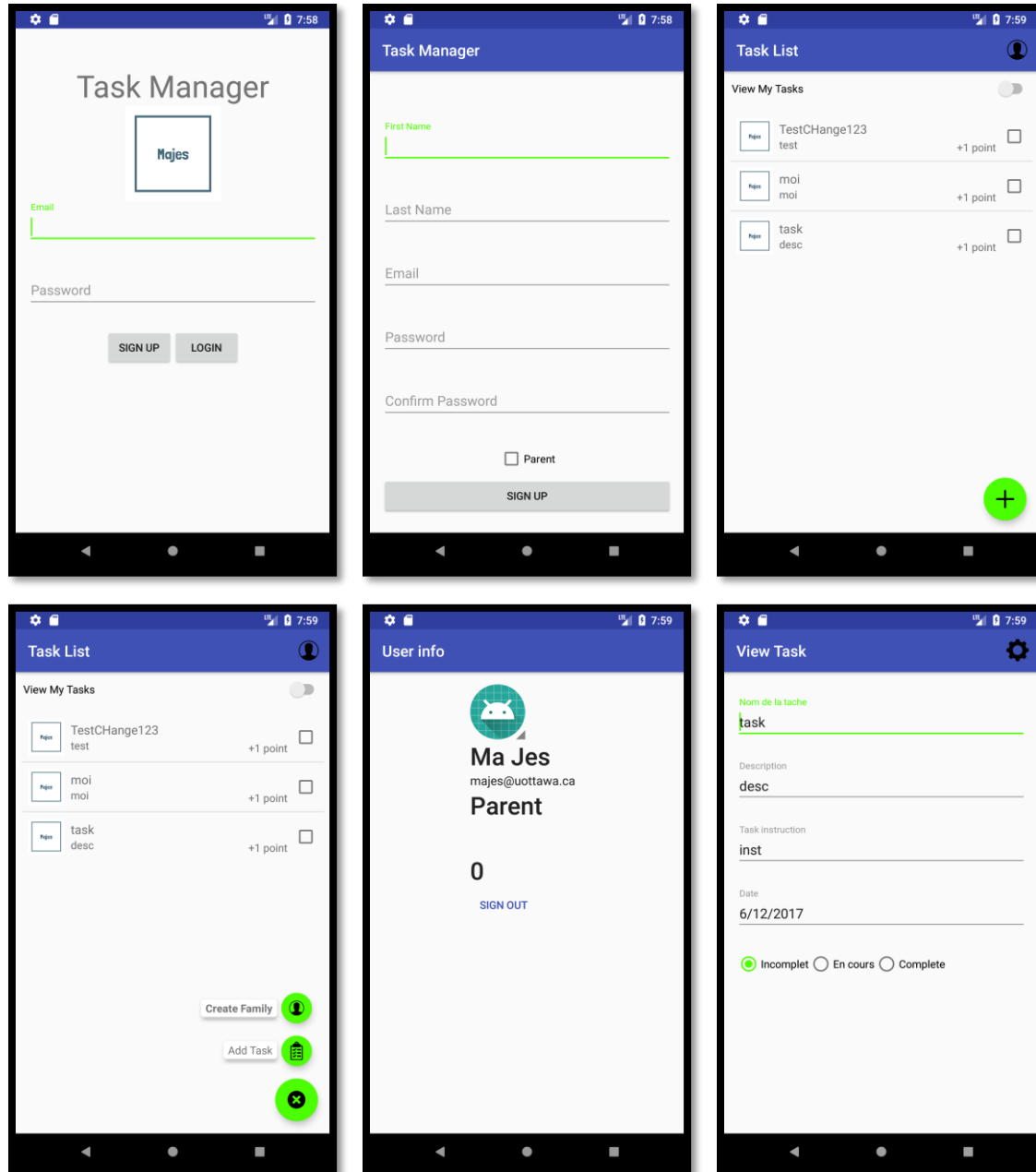


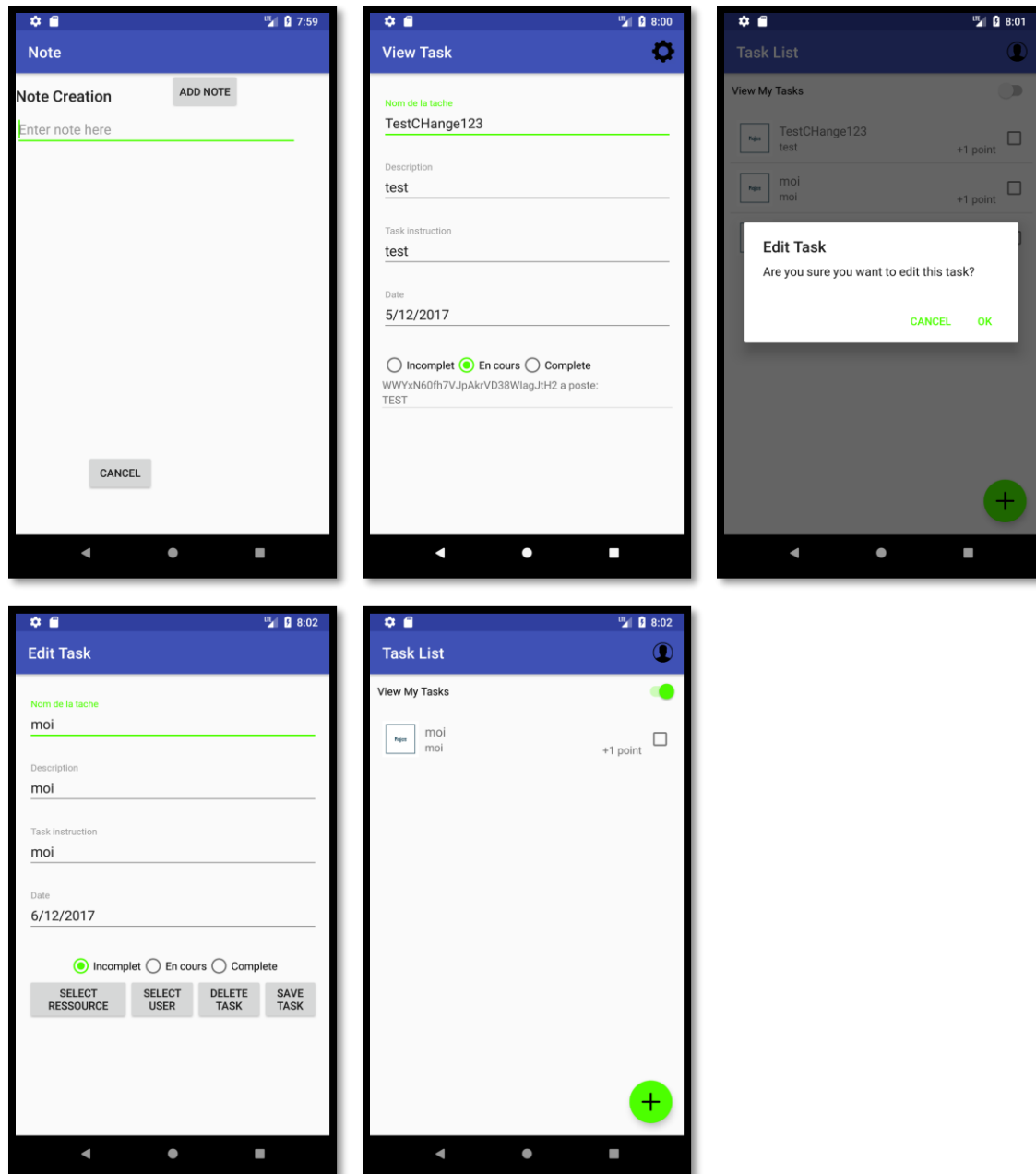
5.4 Diagramme de machine d'état UML

Diagramme de machine d'état pour une tâche.



6. Captures d'écran de l'interface (du troisième livrable)(Mise à jour)





7. Leçons apprises et suggestions

Tâche

- La conception de database grâce aux laboratoires.
- La création d'une application Android.
- La réalisation d'interface usager.
- Les heures 1h30 de laboratoire n'était pas assez.

