

# **Deliverable 4: Final report**

SEG2505-Introduction au génie logiciel

Automne 2017

School of Electrical Engineering and Computer Science

**University of Ottawa**

Professeur: Dr. Miguel A. Garzón

## **Groupe CEYAB**

-Ahmed Amine Yacoubi 8341706

-Brahim Kanouche 8879715

-Yassine Bennani 8256148

-Emmanuel Komla Asinyo 8890676

	<b>Introduction</b>	<b>2</b>
<b>Contributions et corrections des erreurs des livrables</b>		<b>2</b>
Contributions		2
Corrections		3
Livrable 1		3
Livrable 2		3
Livrable 3		3
<b>Les exigences de l'application</b>		<b>4</b>
Exigences fonctionnelles		4
Exigences Non-fonctionnelles		5
<b>Diagrammes UML (cas d'utilisation, Classe, Séquence, machine d'état)</b>		<b>6</b>
Cas d'utilisation		6
Cas d'utilisation 1 : Créer un nouvel utilisateur.		6
Cas d'utilisation 2 : Assigner une tâche.		6
Cas d'utilisation 3 : Ajouter un article à la liste d'achats		7
Diagram de Classe		8
Diagrammes de séquence		9
Diagrammes de séquence 1: Ajout d'un item au shopping-List.		9
Diagrammes de séquence 2: Ajout d'une tache.		10
Diagrammes de séquence 3: Register(ajout d'un utilisateur).		11
Diagram d'état de machine		12
<b>Capture d'écran des interfaces usagers</b>		<b>13</b>
		13
		13
		13
		13
		14
		14

## -Introduction

HouseHold Manager est une application simple et efficace qui vas vous aider a gérer et à distribuer toutes vos taches ménagères. Notre application se base sur la simplicité de l'utilisation vue que on s'adresse à des parents et des grands parents.

L'application a trois simples fragments (avec option du swape pour passer de l'un à l'autre), le premier affiche le shopping liste avec deux inputs textes pour le nom de l'item et sa quantité et un bouton pour l'ajouter à la liste .Le deuxième affiche toute les taches avec l'option de ajouter une nouvelle et avec un bouton pour ajouter une nouvelle et un autre pour la supprimer. Le troisième affichent simplement tout les utilisateurs avec quelques informations et un bouton pour les texter.

## -Contribution and Corrections

Les contributions sont comme qui suit :

	Livable 1	Livable 2	Livable 3	Livable4
Ahmed Amine Yacoubi	20%	30%	25%	25%
Yassine Bennani	30%	25%	20%	25%
Ibrahim Kanouche	25%	20%	25%	30%
Emmanuel Komla Asinyo	25%	25%	30%	20%

## Les corrections :

	Liverable 1
Les changements faits	<p>Les exigences fonctionnelles: on a changé les exigences qu'on a pas pu faire et on les a changé par ce qu'on rajouté de plus ( exigences 1,3,4,5,9,13,14,16,18 )</p> <p>Use Case: On a modifié le use case 2 pour les confirmations d'allocation vue qu'on utilisé sqlite database et on aura besoin d'un serveur pour générer des notifications</p>

	Deliverable 2
Les changements faits	<p>Diagram UML : on a ajouté des classes dont on avait besoin et changé quelque type de relations .</p> <p>Sequence diagram: On a ajouté GUI pour toutes les sequences</p> <p>State machine diagram: On a ajouté les actions et on a corrigé une erreur pour une fleche qui était de double sense .</p>

Les changements faits:

On a mis à jour les screen shots de nos interfaces

## I)-Les exigences de l'application

### -Exigences fonctionnelles

1. Le système doit authentifier l'utilisateur en demandant email et son mot de passe
2. Le système doit permettre d'assigner des tâches à chaque utilisateur
3. Le système doit permettre d'ajouter des utilisateurs
4. Le système doit permettre d'ajouter des notes à chaque tâche.
5. Le système peut permettre d'envoyer un message à un utilisateur
6. Le système peut permettre à l'utilisateur de supprimer une tâche
7. Le système doit afficher le nombre total des tâches assignées à chaque utilisateur.
8. Le système doit afficher les équipements nécessaires pour chaque tâche.
9. Le système doit permettre l'utilisateur de se déconnecter
10. Le système doit avoir une liste des articles pour les courses.
11. Le système doit permettre à l'utilisateur d'ajouter un nouvel article s'il n'existe pas sur la liste.
12. Le système doit afficher tous les utilisateurs de l'application.
13. Le système doit permettre à l'utilisateur de choisir ou prendre une photo de profil
14. Le système doit permettre à l'utilisateur de modifier une tâche
15. Le système doit afficher la liste des tâches à accomplir dans la maison.
16. Le système doit permettre à l'utilisateur d'ouvrir message ou messenger à partir de l'application
17. Le système doit donner l'option d'ajouter des tâches.
18. Le système peut donner l'option de consulter le calendrier dès l'application
19. Le système doit donner l'option de consulter le paramètre de l'application.

20. Le système doit donner l'option de modifier la date d'échéance des tâches.

## Exigences Non-fonctionnelles

1. Le système doit être écrit en Java.
2. Le système doit être construit à l'aide d'Android Studio 2.3 .
3. Le système doit être implémenté avec la méthodologie AGILE.
4. Le système doit être implémenté avant le 6 Décembre 2017.
5. le système doit être capable d'authentifier l'utilisateur dans moins de 5 secondes.

## II)-Diagrammes UML

### 1er cas

Nom : Créer un nouvel utilisateur.

Acteurs : Utilisateur.

Pré-condition: L'utilisateur doit avoir accès à l'application.

Post-condition: L'utilisateur est ajouté à la liste d'utilisateurs.

<u>Utilisateur</u>	<u>Systeme</u>
1. L'utilisateur ouvre l'application	2. Donne l'option à l'utilisateur de s'authentifier
3. L'utilisateur clique sur "nouvel utilisateur"	4. Une nouvelle fenêtre de création pour un nouvel utilisateur s'affiche demandant l'ajout d'un nom et de ses identifiants
5. L'utilisateur ajoute un nom au nouvel utilisateur et ses identifiants	6. Un onglet pour prendre ou choisir une photo apparaît.

7. L'utilisateur confirme la création d'un nouvel utilisateur	8.L'utilisateur est créé et s'affiche dans la table d'utilisateurs
---	--

#### Description :

Créer un nouvel utilisateur est un cas d'utilisation où l'utilisateur est l'acteur et a comme but d'avoir accès à l'application en créant un compte avec une pré-condition d'en avoir l'application installée sur son Cell. Premièrement l'application ouvre une fenêtre à l'utilisateur l'option de s'authentifier si il a déjà un compte avec une option de s'enregistrer si c'est pas le cas .Ensuite l'application ouvre une nouvelle fenêtre afin de permettre à l'utilisateur de créer son compte on lui demandant d'entrer des informations comme le nom , l'email et de créer un mot de passe et le confirmer .Une fois terminé , un nouvel onglet de confirmation apparaît qui demande la confirmation de l'utilisateur pour valider la création du compte . Finalement , le compte est bien créée , l'utilisateur a accès à l'application et s'affiche dans la table des utilisateurs.

#### 2eme cas

Nom : Assigner une tâche.

Acteurs : Utilisateur.

Pré-condition: L'utilisateur doit être authentifié.

Post-condition: La tâche sera assignée à l'utilisateur voulu.

Utilisateur	Systeme
1. L'utilisateur ouvre l'application.	2. Affiche la page principale de l'application
3. L'utilisateur clique sur new task.	4.Ouvre une nouvelle fenêtre qui permet de choisir une personne pour cette tâche , ajouter un délai , les

	équipements nécessaires et une note.
5. a- L'utilisateur remplit les cases obligatoires et optionnelles b-confirme la tâche.	6. Tache crée et affiché sur l'interface des taches.

#### Description :

Assigner une tâche est un cas d'utilisation ou l'utilisateur est l'acteur et a comme but d'assigner une tâche à un autre utilisateur (existant) avec une pré-condition d'être authentifié. Premièrement l'application ouvre la page principale (qui sera Task par défaut) .Ensuite l'utilisateur clique sur new task et l'application ouvre une nouvelle fenêtre qui lui permet de donner d'abord un nom pour cette tâche , choisir une personne(existante) qui va recevoir une demande pour faire cette tâche, ajouter un délais , les équipements nécessaires et une note. Une fois confirmée , l'application génère une notification sous forme d'une fenêtre de confirmation pour la personne choisis qui lui donne l'option de confirmer ou rejeter cette tâche.

#### 3eme cas

Nom : Ajouter un article à la liste d'achats

Acteurs : Utilisateur

Precondition: L'utilisateur doit être authentifié

Postcondition: L'article est ajouté à la liste des achats

<u>Utilisateur</u>	<u>Systeme</u>
1. L'utilisateur ouvre l'application	2. Affiche la page principale de l'application
3. L'utilisateur clique sur shopping	4. Une nouvelle fenêtre de shopping est affichée
5. L'utilisateur ajoute le nom d'un nouvel article sur la case "new item" et la quantité sur «quantity »	6. Le système prend les données saisis

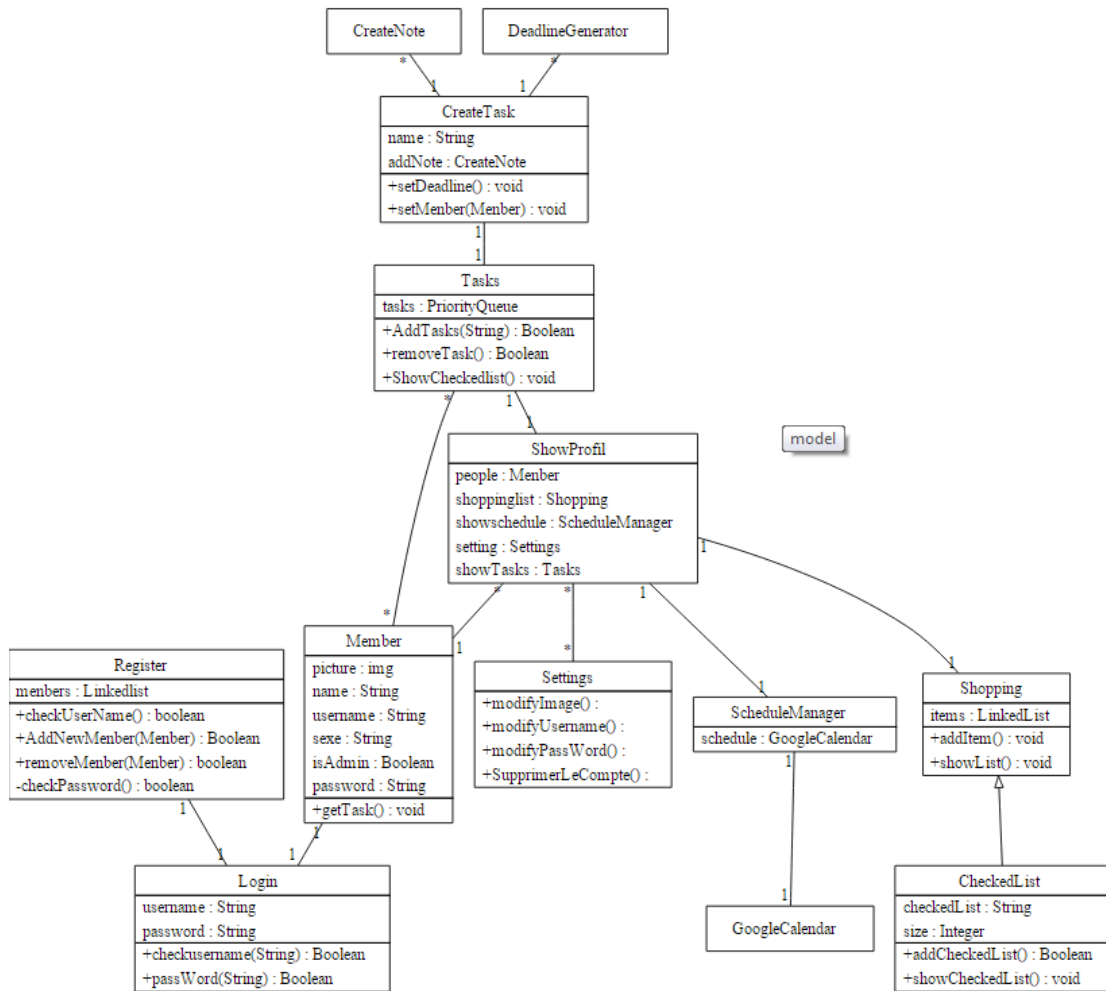


7. L'utilisateur confirme l'ajout de l'article	8. Ajoute l'article à la liste des achats
--	---

#### Description :

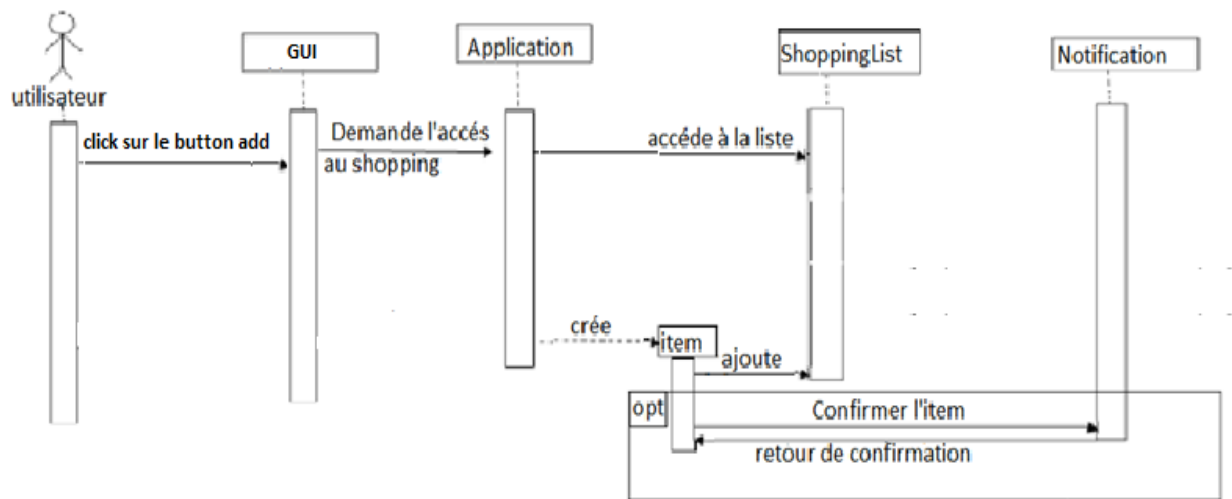
Ajouter un article à la liste d'achats est un cas d'utilisation où l'utilisateur est l'acteur et a comme but d'ajouter un article à la liste d'achats avec une pré-condition d'être authentifié. Premièrement l'application ouvre la page principale (qui sera Task par défaut). Ensuite l'utilisateur clique sur shopping et l'application le prend à la nouvelle fenêtre de shopping qui lui permet de voir tous les articles déjà ajoutés. Et après, l'utilisateur clique sur new item puis il peut taper un nom d'un article, une fois fini l'application lui donne l'option de confirmer son article. L'utilisateur doit appuyer sur confirmer l'ajout de l'article pour que l'application finalement l'ajoute sur la liste des achats.

## Diagrammes de classe

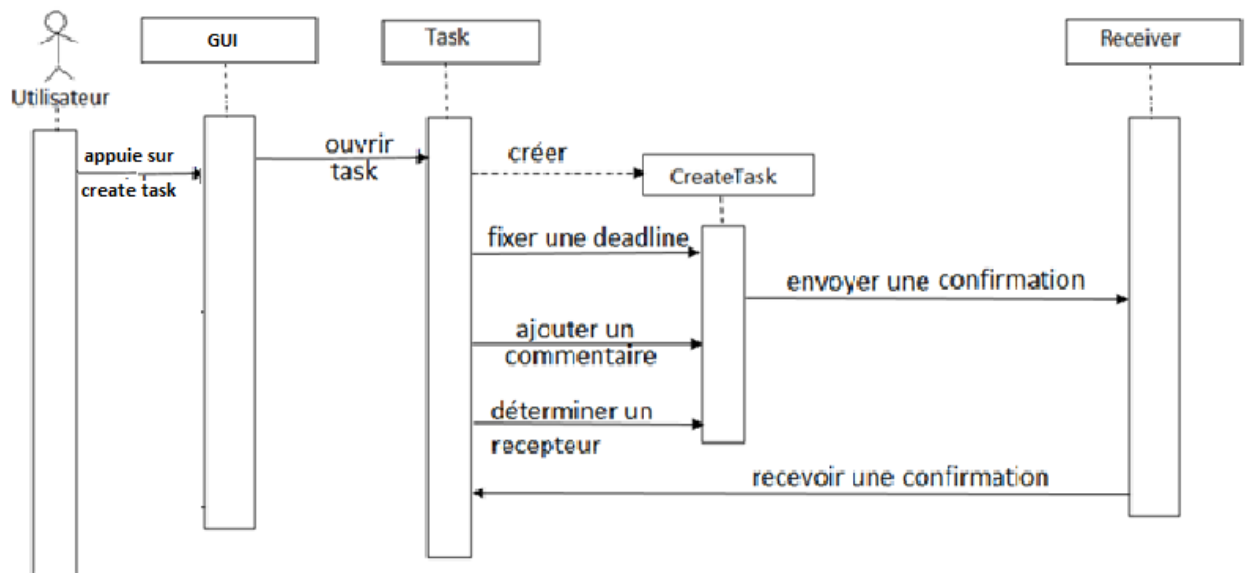


## Diagrammes de séquence

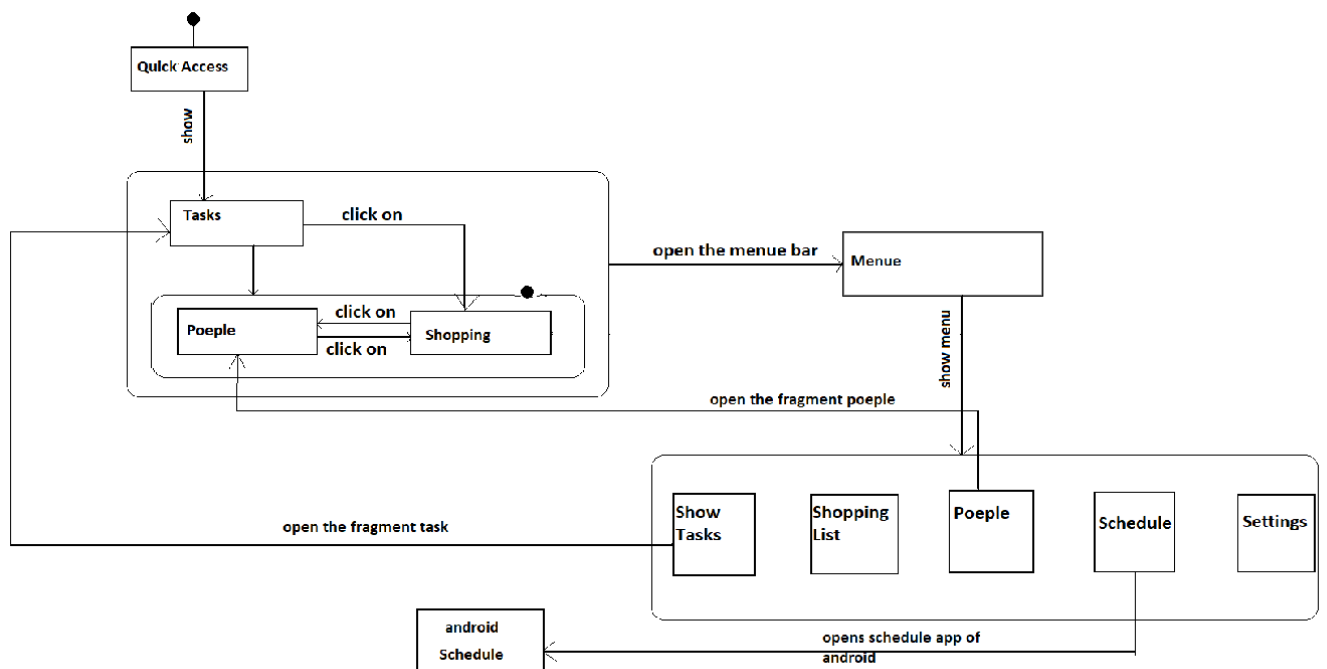
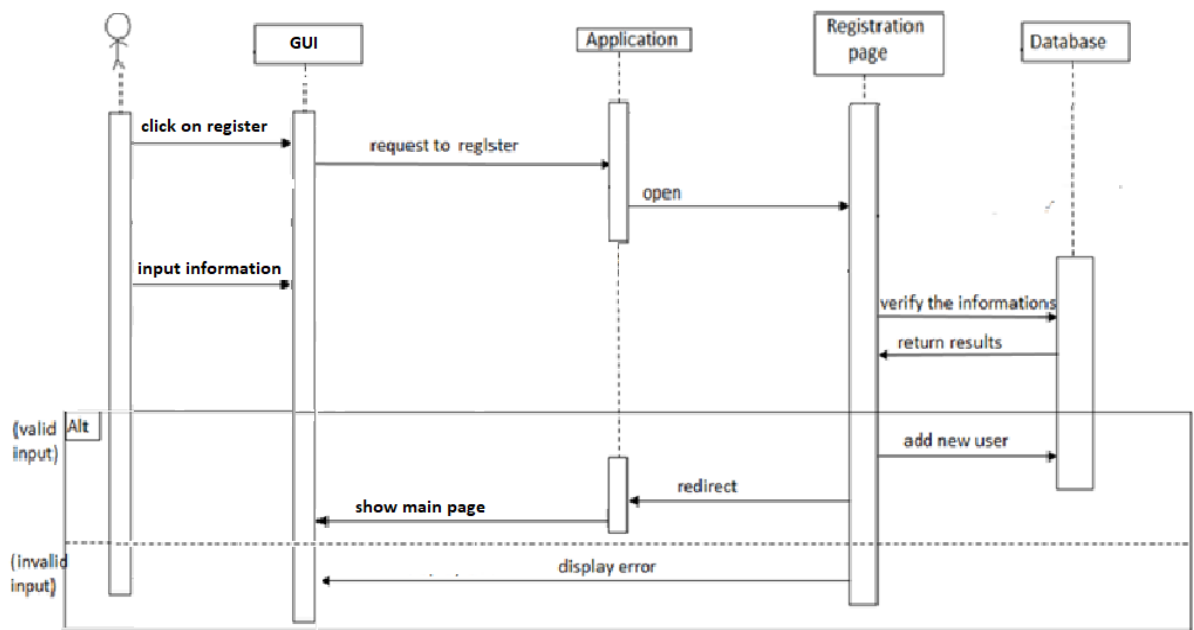
Diagrammes de séquence 1: Ajout d'un item au shopping-List.



Diagrammes de séquence 2: Ajout d'une tâche.



Diagrammes de séquence 3: Register(ajout d'un utilisateur).



Les Interface usager :



100%



9:03 PM

# HouseHold Manager



Email

Password

LOGIN

REGISTER



4G 54% 23:16

## AddImageActivity

### Add Image

Camera

Gallery

Cancel

Click to add an image





100% 9:02 PM

# Contact us

## Email



Support:

[support@householdmanager.ca](mailto:support@householdmanager.ca)

Services:

[services@householdmanager.ca](mailto:services@householdmanager.ca)

Suggestions:

[suggestion@householdmanager.ca](mailto:suggestion@householdmanager.ca)

## Phone



support: +1-514-717-9210

services: +1-647-225-0088

suggestions: +1-613-981-5022

**Always glad to help you ! Toujours content de vous aider**





100%  9:02 PM



**NEED HELP ?**



[help@householdmanager.ca](mailto:help@householdmanager.ca)



+1-514-717-9210 EXT:122



<https://householdmanager.ca/help>



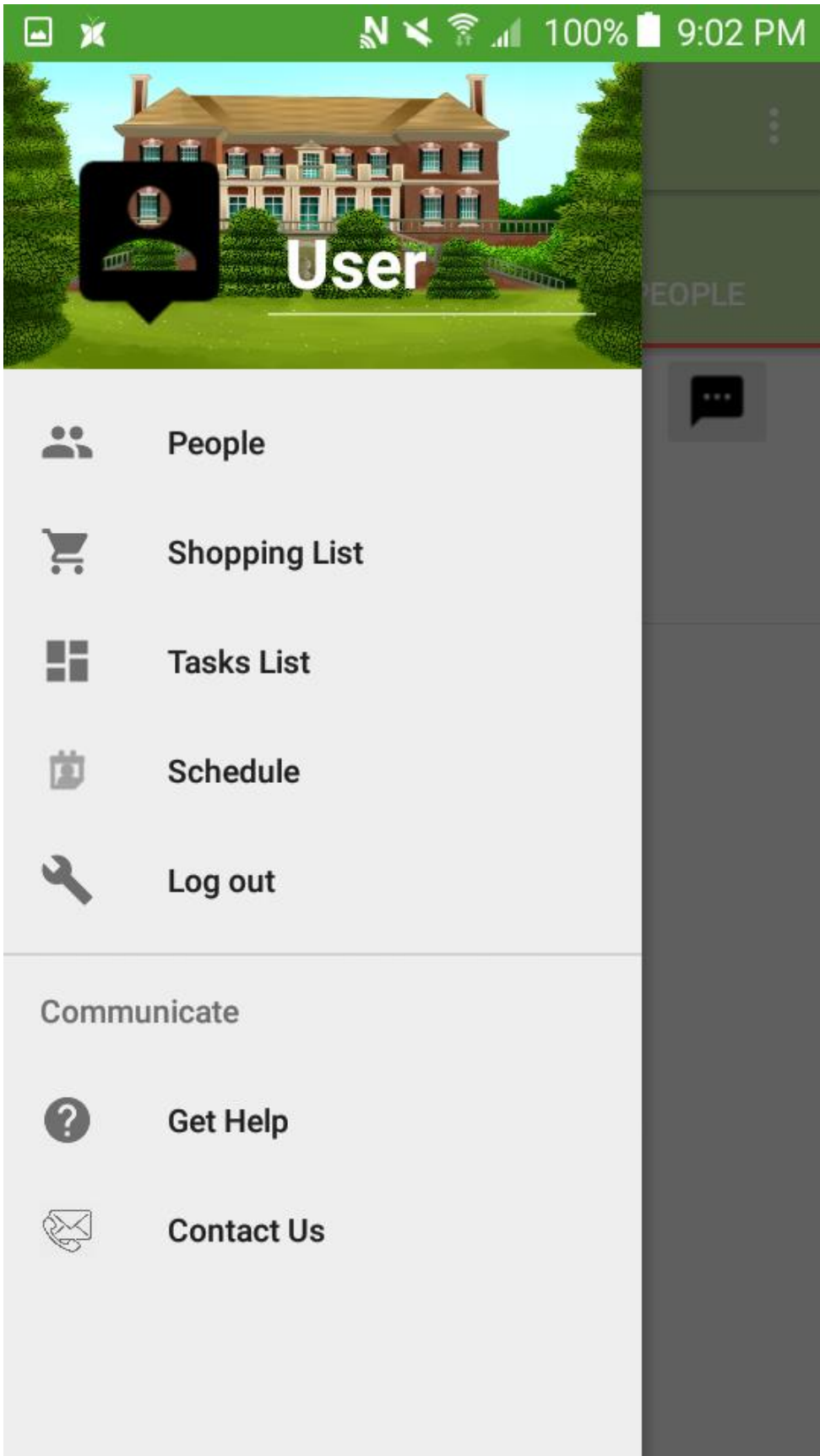
100%  9:02 PM

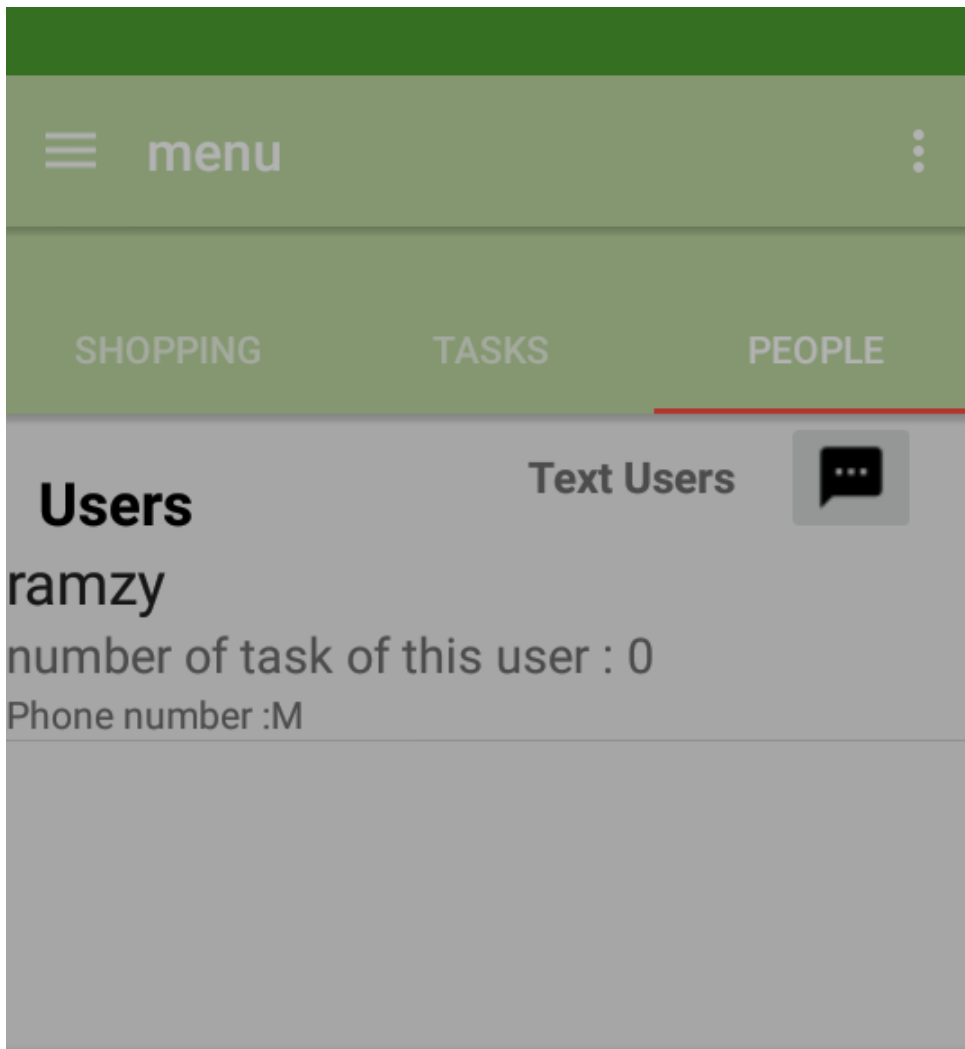


**Thank you for using HouseHold  
Manager**

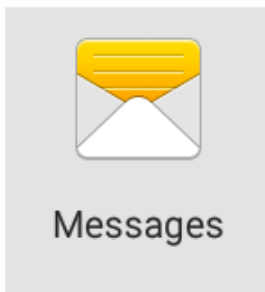
**Click the Buttom Bellow To Logout**

**LOGOUT**





## Ouvrir avec



Messages



Messenger



Hangouts

**CETTE FOIS SEULEMENT**

**TOUJOURS**

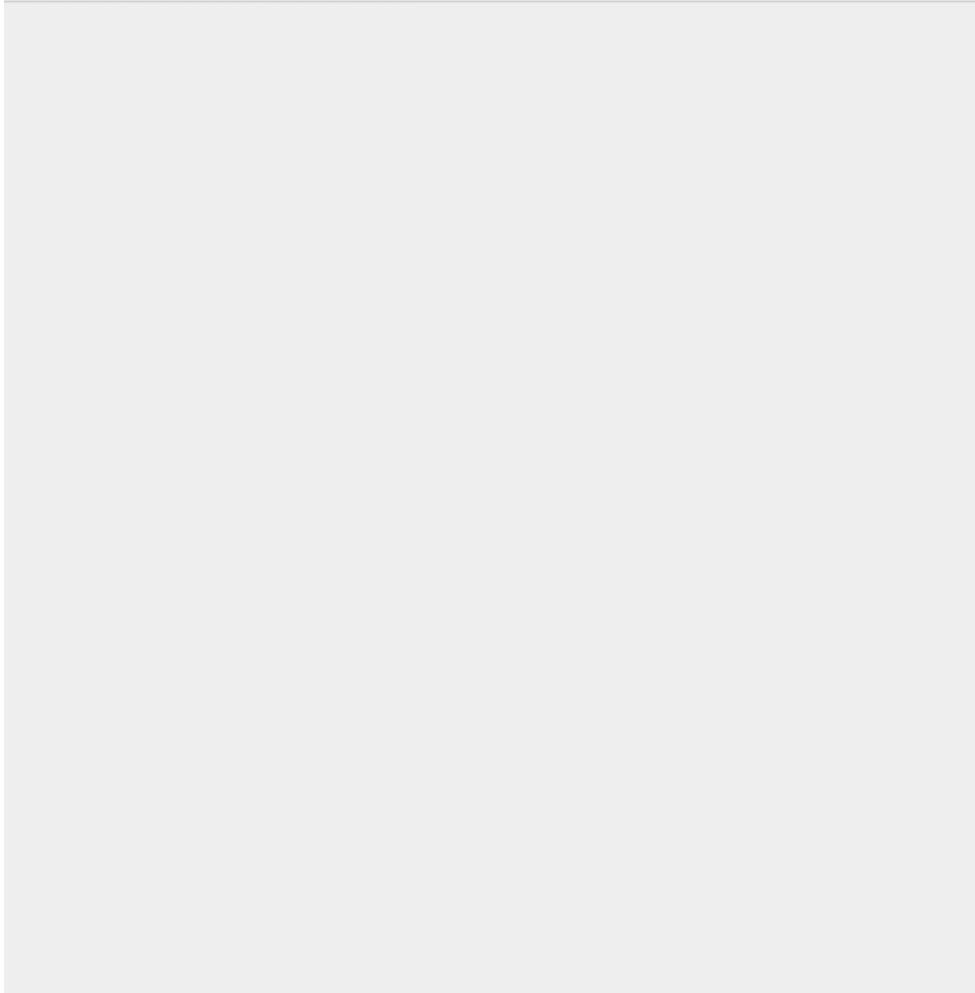
Users

Text Users

ramzy

number of task of this user : 0

Phone number :M



☰ menu



SHOPPING

TASKS

PEOPLE



New Task



delete



edit


# clean living room

Task assigned to : mart

Task due to :2017-12-08 18:30

USE :garbage bags

notes :make sure to clean corners

add item      add quantity      



100%



9:02 PM

## ADD NEW TASK

**Name**

taskname

**Person**

username

**Deadline**

yyymmddhhminmin

**Required equipments**

add equipments

**notes**

add notes

CANCEL

CONFIRM





4G 54% 23:17

☰ menu



SHOPPING

TASKS

PEOPLE

## Users

Text Users



Martin

number of task of this user : 0

Phone number :+1-613-123-45678

## Notes :

- On a eu un problème grave avec notre database qui nuit aux autres méthodes comme affichage des items dans la liste de shopping , on a essayé de résoudre le problème avec le TA (Philippe) mais sans aucun résultat pour qu'on finit par voir le prof . Après plusieurs tentatives pour déboguer notre problème , il a finit par nous suggérer d'utiliser le concept des classes singleton . Finalement on a deux versions , la première avec la database avec le problème cité ci-haut et l'autre avec singleton qui fonctionne bien . Merci pour la compréhension
- A noter que la class MainActivity est la classe Menu dans notre projet.

