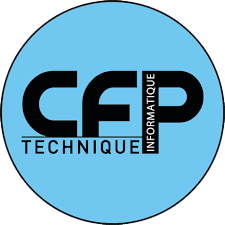


Documentation Technique



Tiago Gerard | Version.1.1 | 04.2018

# Table des matières

[Table des matières 1](#_Toc516816001)

[Introduction 2](#_Toc516816002)

[Etude d’opportunité 2](#_Toc516816003)

[Analyse fonctionnelle 3](#_Toc516816004)

[Poster des offres 3](#_Toc516816005)

[Modifier ses offres 4](#_Toc516816006)

[Rechercher des offres 4](#_Toc516816007)

[Modifier son compte 4](#_Toc516816008)

[Crée un compte 4](#_Toc516816009)

[Description des vues 5](#_Toc516816010)

[Page Création de compte 5](#_Toc516816011)

[Page d’édition du compte 6](#_Toc516816012)

[Page de recherche d’offre 7](#_Toc516816013)

[Page mes offres 8](#_Toc516816014)

[Page Settings 9](#_Toc516816015)

[Page crée une offre 10](#_Toc516816016)

[Page modification d’offre 11](#_Toc516816017)

[introduction 13](#_Toc516816018)

[MCD 14](#_Toc516816019)

[MLD 14](#_Toc516816020)

[Dictionnaire de données 14](#_Toc516816021)

[introduction 15](#_Toc516816022)

[architecture 15](#_Toc516816023)

[description 15](#_Toc516816024)

[Introduction 21](#_Toc516816025)

[architecture 22](#_Toc516816026)

[description 23](#_Toc516816027)

[Butter knife 26](#_Toc516816028)

[Glide 27](#_Toc516816029)

[Plan de tests 30](#_Toc516816030)

[Amélioration possible 34](#_Toc516816031)

[Difficultés rencontrées 34](#_Toc516816032)

[Conclusion 34](#_Toc516816033)

# Introduction

Le gaspillage est le fait de jeté des aliments encore comestible. Le gaspillage peut se retrouver à toute étape de la chaine de nos produits de la fabrication au consommateur. Des milliers de tonnes de nourriture sont jetées par années alors qu’elles pourraient être redistribuées et consommer.

# Étude d’opportunité

Pourquoi ce projet

À Genève, le gaspillage alimentaire est un fléau que l'administration a du mal à combattre. Près d’un tiers, des aliments ne seront pas consommés, mais bien jetés. C’est pourquoi j’ai décidé de créer une application pour aider à mieux utiliser nos ressources.

Description

No Waste est une application d’offre de denrées alimentaires basée sur le partage. Qui a pour but de faire diminuer le gaspillage alimentaire et de favoriser le partage. L’application sur smartphone permet de vérifier si des offres intéressantes sont proposées à tout moment en tout endroit. Il sera possible de proposer une denrée sur l’application et les autres utilisateurs pourront vous contacter pour la récupérer.

Ce que mon application a de plus

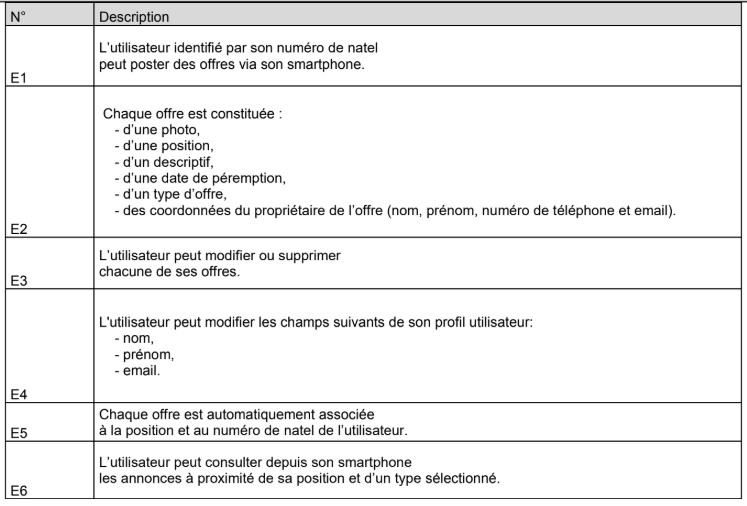
Plusieurs association et mouvement sur le partage des denrées alimentaires proche de péremption existent, mais elle reste souvent méconnue. De plus elles ne peuvent organise des rencontres pour échanger des aliments que rarement et il est difficile pour tout le monde d’adapter son emploi du temps pour y participer. Grâce au smartphone ce service devient permanent et les rencontres se font directement entre particuliers intéressés par le produit. De plus, il permet de pouvoir proposer un aliment à tout moment et donc de faire face aux différentes dates de péremption.

# Analyse fonctionnelle

Description de l’application

No Waste est une application a but non lucratif, permet aux utilisateurs de proposer des denrées alimentaires dont il souhaite se séparer. L’application associe votre numéro de téléphone à votre compte utilisateur. À l’aide d’une interface simple, on peut poster des offres ou en rechercher pour organiser des échanges.

Cahier des charges



Description de la liste des fonctionnalités

Poster des offres

L’application permet de poster des offres de denrées alimentaires. Il vous suffit de rentré une description de l’offre, le type d’aliment, une photo, sa date de péremption et pour donner un indice de distance aux utilisateurs voulant recherche des offres nous vous devrez indiquer l’adresse a la qu’elle vous trouver ou juste laisser l’application récupérer votre position actuelle grâce au GPS du téléphone.

Modifier ses offres

Depuis une liste de vos offres postées, vous pouvez les sélectionner pour en modifier le contenue. L’offre sera alors mise à jour avec les informations changées par vos soins.

Rechercher des offres

Dans l’écran principal de l’application, vous trouverez une liste d’offres classée par proximité avec votre position actuelle. À l’aide du menu filtre, vous pouvez appliquer le filtre de votre choix pour voir uniquement les offres d’un certain type d’aliment.

Modifier son compte

Grâce au menu modifier mon compte il est possible de changer ces derniers à l’exception du numéro de téléphone.

Crée un compte

L’or du lancement de l’application si vous ne disposez pas déjà d’un compte dans l’application NoWaste un écran pour vous permettre de crée un compte apparaitra. Il suffit alors de remplir les informations requises et votre compte est créé.

Attention le numéro de téléphone est automatiquement récupéré pour le bien de votre identification.

# Description des vues

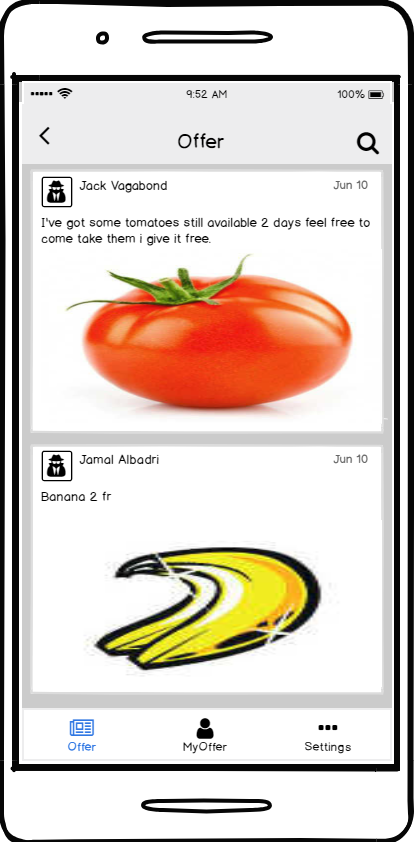
Page Création de compte

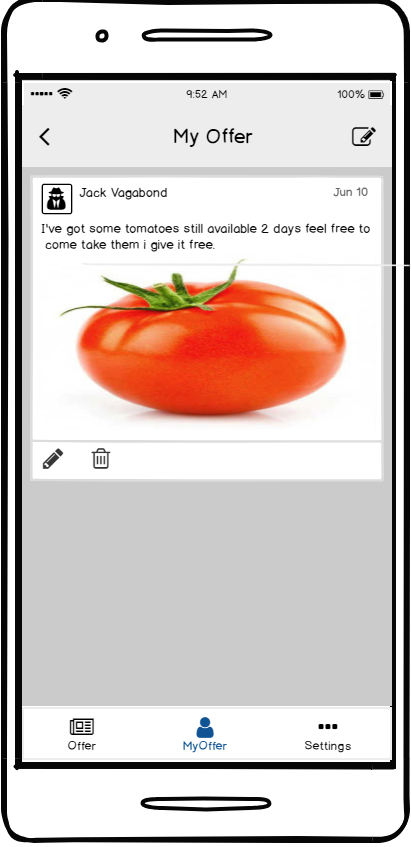
Cette page apparait si aucun compte n’est associé au numéro de téléphone du smartphone. Elle contient des champs pour saisir les données nécessaires à la création du compte et un bouton pour valider la création. Le numéro récupérer sur le téléphone est afficher dans un champ bloquer à l’édition.

Page d’édition du compte

Cette page permet de changer les informations du compte. Elle dispose pour cela des mêmes champs que pour la création et elle est accessible depuis le menu « changer mon compte » lui-même disponible dans l’onglet settings de la page home.

Page de recherche d’offre

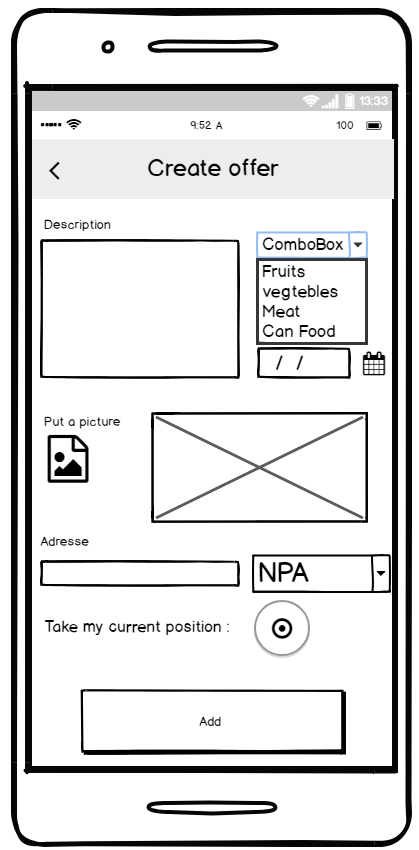
Cette vue contient une liste déroulante d’offre triée par ordre croissant de distance entre l’utilisateur de l’application et l’endroit où a été postée l’offre.

Page mes offres

Cette page liste toutes les offres pas encore arrivées à date de péremption postées par le client. Depuis cette page l’utilisateur peut créer une nouvelle offre ou supprimer/modifier une ancienne.

Page Modofier mon compte

La page modifier mon compte permet a l’utilisateur de visualiser son compte pour s’avoir sous qu’elle nom prénom email il s’est identifié. Elle permet aussi d’accéder a la page changer mon compte via le menu situer en haut.

Page crée une offre

Cette vue permet de créer de nouvelle offre. Avec touts les champs nécessaires a disposition il suffit de remplir les donner et pour créer une offre. Aucun champ ne peut être nul pour qu’une offre soit valide. Elle est accessible depuis le menu créer une offre sur l’onglet mes offres.

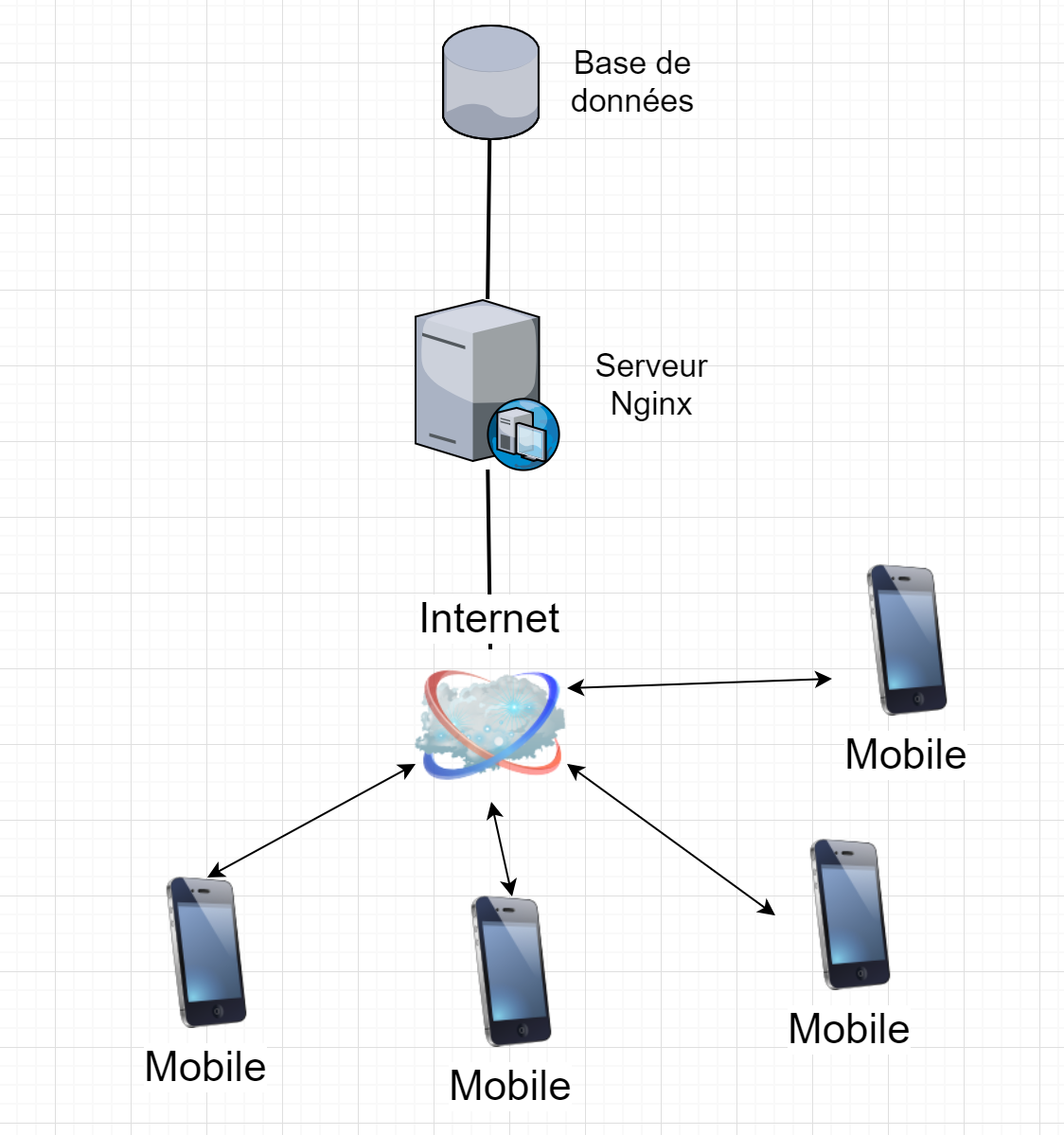
Page modification d’offre

Cette page permet de changer les informations d’une offre. Elle dispose pour cela des mêmes champs que pour la création et elle est accessible en appuyant sur le bouton modifier en dessous de chaque offre dans l’onglet mes offres de la page home.

Analyse Organique

L’architecture généraliste du projet

Le projet comporte deux parties distinctes la partie serveur et la partie applicative. L’applicatif consiste en une App Android programmée en JAVA qui sert d’interface utilisateur. Et d’u service web qui administre la base de données et affiche des données pour le que le smartphone les récupère et les affiche.



Base de données

introduction

PhpMyAdmin : version 4.5.4.1

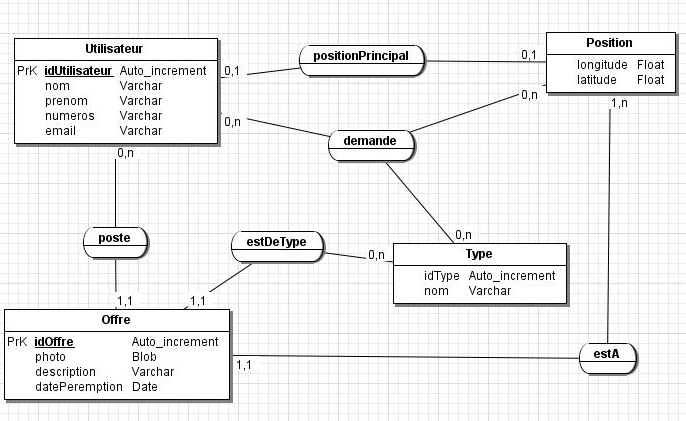
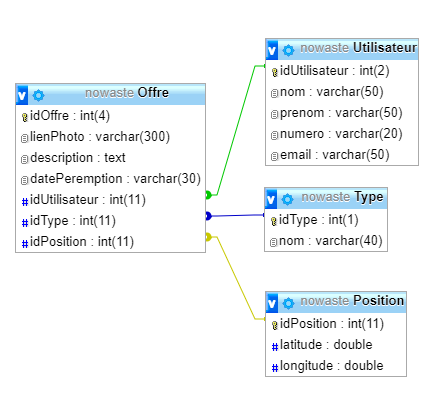
Moteur de base : InnoDB

Interclassement : utf8mb4\_general\_ci

MySQL : version 10.0.33 (Ubuntu 16.04.1)

Nginx : version 1.0.3 (Ubuntu 16.04.1)

MCD

MLD

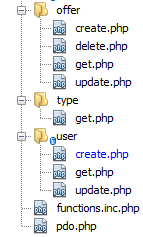
Dictionnaire de données

Service Web

introduction

Le service web est une série de scripts web avec une architecture définie qui va permettre la transition de donnée à l'aide de requête HTTP. Le service web sert notamment comme couche intermédiaire entre le client et la base de données. Grâce à cette architecture, il n’existe aucune interaction directe entre le client et la base de données ce qui augmente la sécurité. Les performances sont également un point important en effet grâce au service web les échanges de données se font via des requêtes HTTP et ne nécessite pas de maintenir une connexion longue et couteuse avec la base donc réduit l’utilisation réseaux sur les clients ici des smartphone Android. Le service effectuera aussi les traitements des données et des requêtes ce qui permet au smartphone client de juste envoyer et récupérer des donner sans avoir de travail supplémentaire.

architecture

Le service Web suit une architecture REST ce qui signifie que les noms de chemin sont explicites. Et que chaque page retourne des données qui sont dans ce cas encodées en JSON.

description

**10.134.97.230/nowaste/service/functions.inc.php**

La page functions.php est la librairie de fonctions contenant toutes les fonctions de gestion comme de vérification pour la base, mais aussi les fonctions d’administration de la base

**10.134.97.230/nowaste/service/pdo.php**

La page pdo.php permet de récupérer un objet PDO avec la connexion à la base de données avec le client Service avec les identifiants suivant :

Hote = localhost  
Utilisateur = serviceClient  
Mot De Passe = service4264  
nom de la base = nowaste

Elle est utilisée par function.inc.php qui va effectuer les requêtes sur la base de donnée.

**10.134.97.230/nowaste/service/offer/create.php**

Le corps de la requette HTTP

|  |  |
| --- | --- |
| Clefs | Valeurs |
| Image | L’image envoyée |
| Name | Le nom de l’image |
| descscription | La description de l’offre |
| latitude | La latitude de l’offre |
| longitude | La longitude de l’offre |
| idUtilisateur | Identifiant de l’utilisateur voulant créer cette offre |
| idType | Identifiant du type de l’offre |

La page create.php du répertoire offer va récupérer les différentes données envoyer par un requête HTTP en GET sauf l’image et son nom qui sont envoyés a l’aide du PART puis il va d’abord à l’aide des fonctions présente dans functions.inc.php vérifier si un enregistrement de la position envoyer existe dans la base si tel est le cas il récupère l’identifiant de la position existe sinon il crée un nouvel enregistrement et récupère son id. Ensuite, il va avec des vérifications d’extension et de taille enregistrez l’image envoyée par la requête HTTP et crée un lien unique pour cette image qui sera stocké en base. Une fois ces, étapes remplies le script va créer un nouvel enregistrement offre avec tous les champs préalablement récupérer.

10.134.97.230/nowaste/service/offer/get.php

Le corps de la requette HTTP

|  |  |
| --- | --- |
| Clefs | Valeurs |
| latitude | La latitude de l’offre |
| longitude | La longitude de l’offre |

La page get.php du répertoire offer a deux fonctionnalités si la requête HTTP GET contient les coordonner de l’utilisateur alors la page va rechercher les 20 offres les plus proches de la position et les renvoyer sous le format JSON ci-dessous

{

"is": "100",

"prenom": "tiago",

"lien": "0.00129900-152881288920180521-031142.jpg",

"description": "ceci est une description",

"date": "2018-05-08",

"idType": "6",

"distance": "51796.5747815184",

"contact": "+41791931253"

},

Le corps de la requette HTTP

|  |  |
| --- | --- |
| Clefs | Valeurs |
| idUtilisateur | L’is de l’utilisateur voulant récupérer ses offres |

Si elle ne contient qu’un is utilisateur la page va retourner une liste des offres postées par ce dernier fonctionnalisé servant à afficher les offres de l’utilisateur pour qu’ils puissent les modifier ou les supprimer. La liste d’offre s’affiche avec le format suivant :

{

"is": "100",

"prenom": "tiago",

"lien": "0.00129900-152881288920180521-031142.jpg",

"description": "hdhd",

"date": "2018-05-08",

"idType": "6",

"latitude": "46.195882365447154",

"longitude": "6.1101684447243745",

"contact": "+41791931253"

}

**10.134.97.230/nowaste/service/offer/update.php**

Le corps de la requête HTTP

|  |  |
| --- | --- |
| Clefs | Valeurs |
| descscription | La description de l’offre |
| latitude | La latitude de l’offre |
| longitude | La longitude de l’offre |
| idUtilisateur | Identifiant de l’utilisateur voulant créer cette offre |
| idType | Identifiant du type de l’offre |

L’update d’une offre reprend les mêmes mécanismes de vérification que la création puis va éditer l’enregistrement de la base de données.

**10.134.97.230/nowaste/service/offer/delete.php**

Le corps de la requête HTTP

|  |  |
| --- | --- |
| Clefs | Valeurs |
| idOffre | L’identifiant de l’offre à supprimer |

La page delete.php du répertoire offer sert à supprimer une offre de la base grâce à son identifiant.

**10.134.97.230/nowaste/service/type/get.php**

La page get.php du répertoire type n’a besoin d’aucun paramètre de requête HTTP elle va toujours retourner la liste des types disponible sur la base servant notamment a la sélection du type durant la création d’une offre sur l’appareil client.

**10.134.97.230/nowaste/service/user/get.php**

Le corps de la requête HTTP

|  |  |
| --- | --- |
| Clefs | Valeurs |
| PHP\_AUTH\_USER | Le numéro de téléphone de l’utilisateur |

La page get.php du répertoire user va récupérer s’il existe l’utilisateur qui a ce numéro de téléphone enregistrer à son compte et le retourner en JSON. Cette page sert pour l'authentification, mais aussi à récupérer les données d’un utilisateur en JSON avec le format qui suit :

{

"idUtilisateur": "30",

"nom": "gerard",

"prenom": "tiago",

"numero": "+41791931253",

"email": "tiago"

}

**10.134.97.230/nowaste/service/user/create.php**

Le corps de la requête HTTP

|  |  |
| --- | --- |
| Clefs | Valeurs |
| email | L’email de l’utilisateur |
| nom | Le nom de l’utilisateur |
| prenom | Le prénom de l’utilisateur |
| numero | Le numéro de téléphone de l’utilisateur |

La page create.php du répertoire user sert à créer un utilisateur avec tous les champs envoyer.

**10.134.97.230/nowaste/service/user/update.php**

Le corps de la requête HTTP

|  |  |
| --- | --- |
| Clefs | Valeurs |
| email | L’email de l’utilisateur |
| nom | Le nom de l’utilisateur |
| prenom | Le prénom de l’utilisateur |
| numero | Le numéro de téléphone de l’utilisateur |
| idUtilisateur | L’identifiant de l’utilisateur |

La page update.php du répertoire user permet de modifier les données de l’utilisateur possédant l’identifiant envoyé.

Application Android

Introduction

L’application est programmée à l’aide d’Android studio, l’IDE de Google pour programmer pour de l’Android. C’est une ide très souvent mis a jour qui prend en charge de nombreux outils

Android Studio : version 3.1

Gradle : version 3.1.2

SDK Minimal :version 21

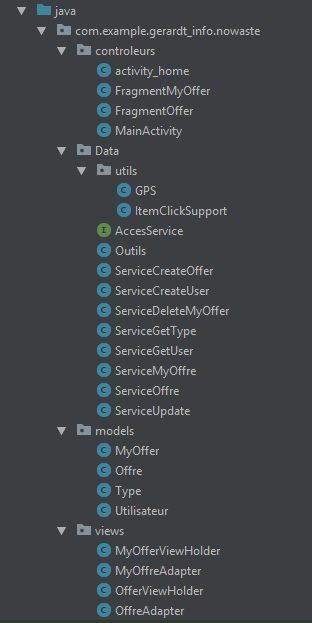
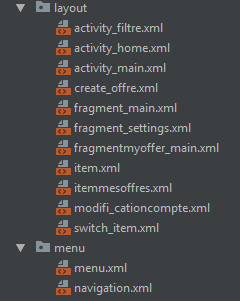
Nginx : version 1.10.3 (Ubuntu 16.04.1)

Librairie Butterknife : version 8.8.1

Librairie Retrofit 2 : version 2.3.0

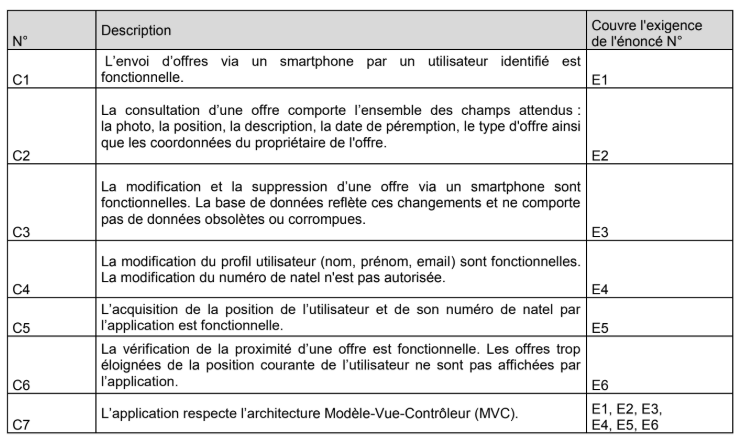
Librairie Glide : version 4.7.1

architecture

L’application Android suit le paterne MVC. Avec un package data qui gère tous les accès au service web.

Capture d’écran des vues XML et de leurs

Capture d’écran des packages Java

description

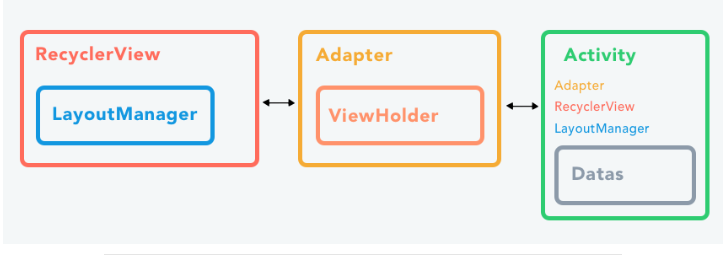
Description global**e**

Pour toutes les vues mise à part la création de comptes, j’ai décidé d’utiliser un seul contrôleur « Activity\_Home ». Le contrôleur va gérer l’ouverture des vues et toutes leurs gestions. Pour les requêtes HTTP l’application utilise la Librairie Retrofit2 qui permet empaqueter les requêtes dans des interfaces et donc d’implémenté des gestionnaires de retour d'appel dans les classes les utilisant (les contrôleurs).

Consulter les offres

Avec un service web proposant des pages de récupération d’offre, il a suffi d’implémenter l’interface retrofit que j’ai écrite pour récupérer la liste des offres préalablement triée par le service. Retrofit permet aussi de sérialiser le JSON envoyer par le service en objet java pour cela on crée des class modèle srializable sur le format JSON envoyer pour cela j’utilise un site «http://www.jsonschema2pojo.org » qui créer une class java en fonction d’un format JSON. L’application récupère la position de l’utilisateur et l’envoie dans la requête HTTP pour que le service puisse retourner les offres proch

e.

Pour l’affichage j’ai utilisé les « RecyclerView » un élément d’Android qui permet d’afficher des « Cards » qui grâce a un contrôleur appeler « Adapter » va uniquement charger les items visibles sur l’écran du smartphone et va réutiliser les mêmes conteneurs en chargeant les donner avec celle de l’élément d’après si l’utilisateur descend dans la « ScrollView ».  
Un contrôleur appelé fragment va créer cette ScrollView et utiliser l’adapter pour implémenter les « cards » dans la vue ensuite l’activité home qui gère la bonne navigation va implémenter ces fragments pour qu’il rentre dans l’espace disponible de la page home.

Consulter mes offres

Pour le menu de navigation Mes Offres j’ai utilisé le même fonctionnement avec de petites modifications. D’abord, l’interface d’accès au service ne retourne pas les mêmes objets il retourne une liste d’objet « MyOffre » plutôt qu’une liste d’« Offre » ensuite dans l’adapter et la « RecyclerView » en plus des champs nécessaires à l’affichage j’ai implémenté un Bouton et son  « listener » ainsi qu’un « OnClick listener » sur la « cards » pour pouvoir sélectionner une offre juste en appuyant sur son « cards ».  
Une fois l’offre sélectionner la vue avec les champs pour update l’offre s’affiche et les champs sont remplis avec les donner de l’offre sélectionner.

Création d’une offre

* La création consiste en une vue avec les champs suivants
* Description
* Date
* Photo
* Type
* Adresse/npa

J’ai utilisé un sélectionneur de dates qui ouvre un pop up pour récupérer la date et pour pouvoir gérer le format de date  
La photo ouvre la galerie et permet de sélectionner une image   
Puis quand l’utilisateur je vérifie si tous les champs sont renseignés et j’envoie en requête HTTP les données pour créer une offre.  
L’image est alors enregistrée sur le serveur et son lien est enregistré dans la base

Modification d’une offre

Pour la modification grâce à l’instanciation du « onclick listener » dans la « recyclerView » je peux afficher l’écran de création d’offres avec les données de l’offre sélectionnée. Puis pour la modification, j'envoie les données aux serveurs en requête HTTP le serveur va donc s’occuper de faire les modifications.

Suppression d’une offre

Pour la suppression grâce au bouton instancier sur chaque offre de l’utilisateur. On récupère l’offre sélectionnée et on peut l’effacer à l’aide d’une requête serveur.

Création d’un utilisateur

Si aucun compte n’est lié au numéro de l’utilisateur, l’application permet à l’utilisateur de créer son compte en récupérant son numéro et le laissant renseigner les champs restants. Ensuite, on envoie au serveur puis on le connecte pour qu’il puisse voir les offres.

Modification de l’utilisateur

Une fois la page de modification ouverte on récupère les données de l’utilisateur et on les affiche pour qu’il puisse les modifier. Ensuite, on les renvoie et le service s’occupe de modifier les données.

Librairie

Butter knife



Butter Knife est une librairie écrite par Jake Wharton qui permet d’associer des éléments de la vue aux éléments du code.

Import de la librairie :

implementation 'com.jakewharton:butter knife:8.8.1'

annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'

Glide

Glide est une librairie que j’ai utilisée pour afficher une image grâce à une URL

Import de la librairie :

implementation 'com.github.bumptech.glide:glide:4.7.1'

annotationProcessor 'com.github.bumptech.glide:compiler:4.7.1'

Plan de tests

Test unitaire du service web

Les tests unitaires du service web sont effectués avec Postman un utilitaire de requête HTTP pour vérifier si le service réagit correctement.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Date | Fichier testé | Scénario | Valeurs de test | Résultat attendu | Résultat |
| 01 | 07.06.2018 | User/create.php | Vérifier l’ajout d’un utilisateur | Paramètre POST  email= «tiago.gerardet@hotmail.ch»  nom=  «gerard»  numero=  «+410791931253»  prenom= « tiago» | La page retourne «true» en JSON et la base a été modifiée | OK |
| 02 | 07.06.2018 | User/update.php | Vérifier la modification d’un utilisateur | Paramètre POST  idUtilisateur= «30»  email= «tiago.gerardet@hotmail.ch »  nom=  «gerard»  numero=  «+410791931253 »  prenom= « tiago» | La page retourne «true» en JSON et la base a été modifier | OK |
| 03 | 07.06.2018 | User/get.php | Vérifier la récupération des données d’un utilisateur | Paramètre Header  Username= «+410791931253» | La page retourne un utilisateur | OK |
| 04 | 07.06.2018 | Offer/get.php | Vérifier la récupération des offres | Paramètre GET  latitude= «46.5 »  longitude= «5.4 »  idUtilisateur= «30» | La page retourne des offres en JSON | OK |
| 05 | 07.06.2018 | Offer/get.php | Vérifier la récupération des offres d’un utilisateur | Paramètre GET  idUtilisateur= «30» | La page retourne des offres en JSON | OK |
| 07 | 07.06.2018 | Offer/get.php | Vérifier la récupération des offres trier par type | Paramètre GET  latitude= «46.5 »  longitude= «5.4 »  idUtilisateur= «30»  idType= « 5 » | La page retourne des offres en JSON | OK |
| 08 | 07.06.2018 | Offer/create.php | Vérifier la création d’une offre | Paramètre GET  <file>téléchargement.jpg= «  latitude= «46.5 »  longitude= «5.4 »  idUtilisateur= «30»  idType= « 5 » | La page retourne «true» en JSON et la base a été modifiée | OK |
| 09 | 07.06.2018 | Offer/update.php | Vérifier la modification d’une offre | idOffre= « 85 »  description= « test »  datePeremption =  « 2018-05-19 »  latitude= «46.5 »  longitude= «5.4 »  idType= « 4 » | La page retourne «true» en JSON et la base a été modifiée | OK |
| 10 | 07.06.2018 | Offer/delete.php | Vérifier la suppression d’une offre | idOffre= « 85 » | La page retourne «true» en JSON et la base a été modifiée | OK |

Tests fonctionnels

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Date | Scénario | Valeur de test | Résultat attendu | Résultat |
| 01 | 13.06.2018 | 1. Ouvrez l’application 2. Donner les droits s’ils sont demandés 3. Crée un compte |  | L’application vous a créé un compte et montre la page home | OK |
| 02 | 13.06.2018 | 1. Effectuez l’étape 01 2. Attendez que les offres chargent |  | Une liste d’offre s’affiche dans une « scrollView » | OK |
| 03 | 13.06.2018 | 1. Effectuez le test 2 2. Appuyer sur le menu du bas « MyOffer » 3. Attendez que l’écran se charge |  | Une liste avec vos Offres afficher | OK |
| 04 | 13.06.2018 | 1. Effectuez le test 2 2. Appuyer sur le menu en haut 3. Sélectionner le menu filtrer Offre 4. Sélectionnez un type |  | Les offres sont uniquement du type sélectionné | OK |
| 05 | 13.06.2018 | 1. Effectuez le test 2 2. Appuyer sur le menu en haut 3. Sélectionner le menu créer une offre 4. Remplir tous les champs 5. Appuyer sur Ajouter |  | La page mes offres doit contenir une offre supplémentaire | OK |
| 06 | 13.06.2018 | 1. Effectuez le test 2 2. Sélectionner le menu modifier mon compte 3. Modifier le prénom 4. Appuyer sur valider |  | Le prénom des offres sur la page mes offres à changer | OK |

# Amélioration possible

Dans l’optique d’un projet visant à être mis en production, plusieurs améliorations seraient à envisager. D’abord pour la sécurité un système de tokens entre le client et le service web serait une bonne pratique. Pour éviter de transiter les numéros à chaque requête HTTP.

Une autre amélioration possible serait de maintenir des connexions avec le service web. Pour optimiser la connexion.

Difficultés rencontrées

J’ai rencontré des soucis dans l’architecture de mon projet notamment de l’application Android l’architecture n’étant pas parfaite il ma fallut plus de vérification de données et d’état de l’application.

# Conclusion

Pour conclure NoWaste propose une solution technique à un besoin de la vie réelle. Ce projet m’a donnée une vision de ce qu’est un projet informatique et m'a énormément appris sur les réflexions à avoir. Malgré des difficultés rencontrées par moment j’ai su m’adapter à la situation.

Je remercie le centre de formation professionnelle technique pour m’avoir permis de faire mon travail dans des conditions optimales.   
Je remercie aussi tous particulièrement mon maitre d’apprentissage qui a suivi mon TPI M. François.

# Bibliographie

<https://github.com/bumptech/glide>

<http://jakewharton.github.io/butterknife/>