

LDNR

Projet Java

FAD Project

Fabrice HOOGE, Ainsley REMY, David Fournier
28/03/2017

Date	Auteur	Commentaire
28/03/2017	Fabrice HOOGE	Initialisation du document
29/03/2017	Ainsley REMY	Ajout des cas d'utilisation
29/03/2017	David FOURNIER	Ajout des maquettes visuelles

I. Cahier des charges

A. Expression du besoin

L'objectif du projet est de fournir une application qui permet de gérer un carnet d'adresses. Les fonctionnalités minimales seront

- Saisie de nouvelles personnes sans limitation de nombre
- Sauvegarde / récupération dans un fichier
- Informations individuelles collectées : nom, prénom, numéro(s) de téléphone, adresse postale, adresse(s) email
- Recherche d'un ensemble de fiches répondant à un nom donné.

Le travail rendu doit être totalement fonctionnel, c'est-à-dire qu'il doit compiler et s'exécuter sans problème pour les fonctionnalités implémentées (gestion complète des cas d'exception).

B. Ajouts au cahier des charges

Plusieurs précisions sur le résultat rendu ont été apportés afin de respecter les délais demandés :

L'application ne sera pas multi-utilisateur, la sauvegarde sera effectuée sur le poste de l'utilisateur et ne sera pas partagée.

La gestion des données et notamment leur sauvegarde se feront par sérialisation d'objet. Ceci permettra une portabilité plus grande et une simplicité de mise en œuvre.

La sauvegarde des contacts sera effectuée uniquement à la fermeture de l'application afin de limiter les accès disques.

Le contrôle des informations saisies sera mise en place dans une 2^{ème} version de l'application. Mais cela est prévu dans le diagramme de classe.

La recherche d'un contact se fera exclusivement sur le nom.

Un contact valide est un contact pour lequel le nom, au moins, a été défini.

Un contact aura 3 numéros de téléphone au maximum : professionnel, domicile et bureau

Un contact aura 2 adresses mail au maximum : professionnelle et personnelle

C. Version de l'application

Voici ce que contiendront les différentes versions de l'application :

Version 1 :

- Ajout de contact
- Recherche par le nom uniquement
- Sauvegarde par sérialisation
- Récupération par désérialisation
- Utilisation par une interface console

Version 2 prévisionnelle :

- Utilisation d'une BDD
- Interface graphique
- Suppression d'un contact
- Validation des informations saisie

II. Planning

Le planning projet prévisionnel suivant a été défini :

Activité	27-mars	Colonne1	28-mars	Colonne2	29-mars	Colonne3	30-mars	Colonne4
	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi	Matin	Après-midi
Conception								
Code								
Test								
Documentation								

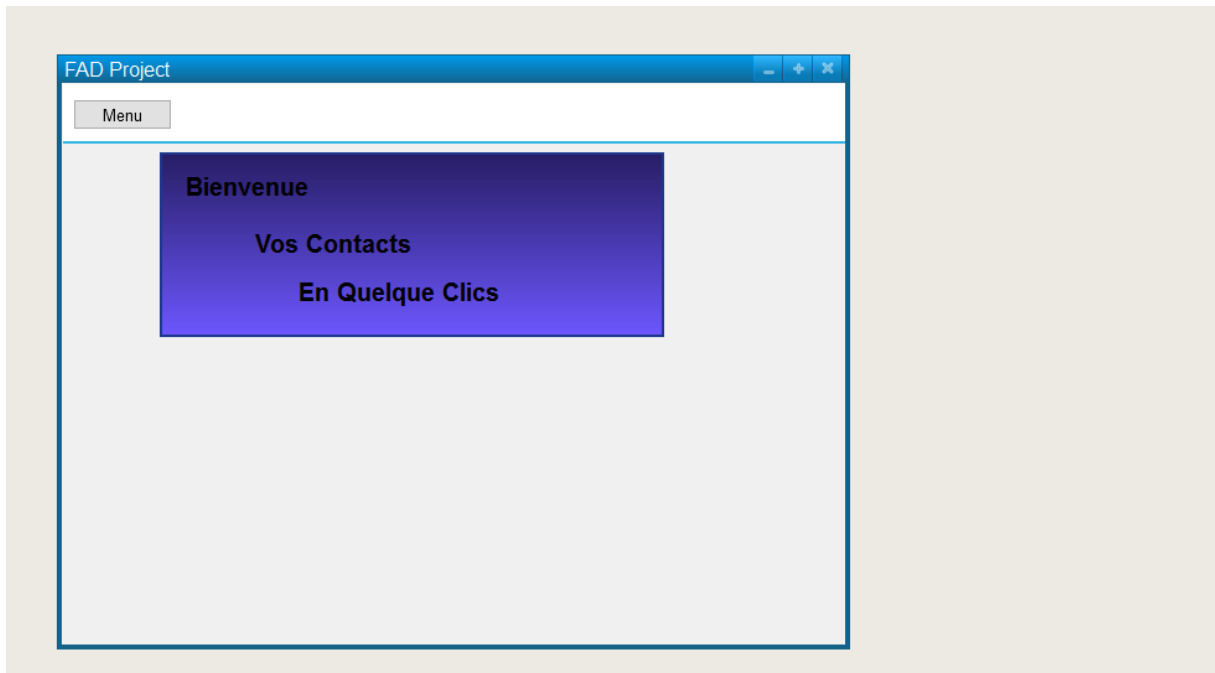
III. Maquette

A. Accueil

Le premier visuel ci-dessous est la page d'accueil lorsqu'un utilisateur démarre l'application.

Une fenêtre s'ouvre aux dimensions prédéfinies :

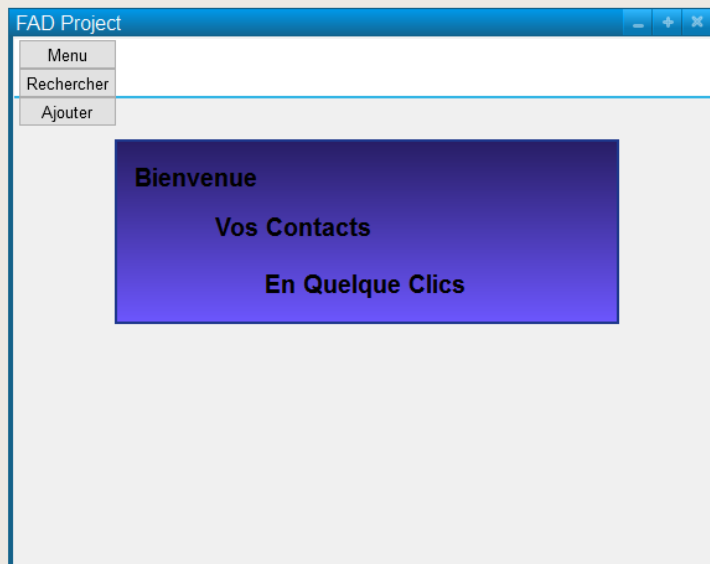
- Il a la possibilité de réduire ou de fermer cette fenêtre.
- Il peut cliquer sur le bouton « Menu »
- Visuellement, on observe également un message de bienvenue.
- Le fond d'écran ici est gris ; il correspond au bureau de l'utilisateur.



B. Menu

Après avoir cliqué sur « menu », l'utilisateur voit apparaître deux nouveaux boutons, à savoir « rechercher » et « ajouter ».

Pour le reste, pas de changements particuliers à observer.



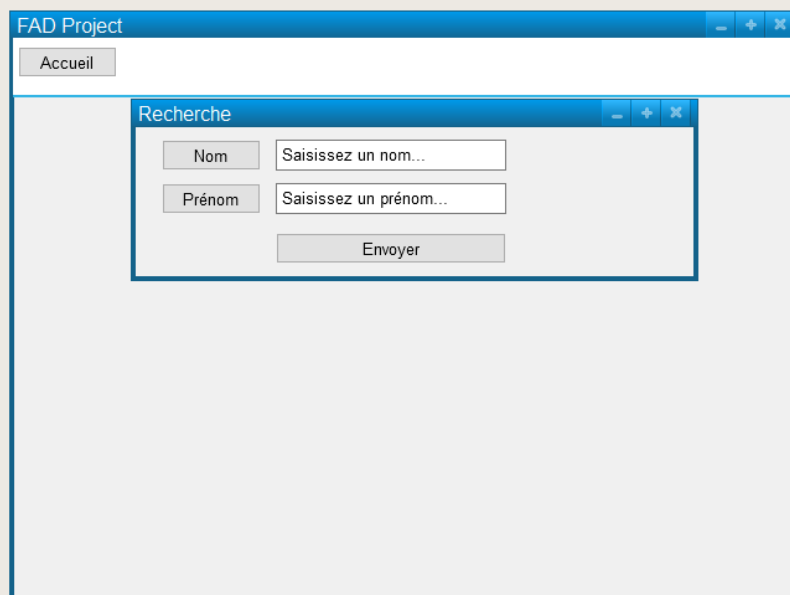
C. Fonction Rechercher

A partir du menu, l'utilisateur souhaite donc faire une recherche ; l'écran suivant va donc lui permettre de saisir le nom et/ou le prénom.

On observe ici un nouveau bouton, « accueil », qui permet donc de revenir à la première page de l'application.

Les champs nom et prénoms indiquent où entrer les informations et l'utilisateur n'a plus qu'à cliquer sur le bouton « envoyer » pour valider sa saisie.

Par ailleurs, nous pouvons observer que le message de bienvenue n'est plus affiché.



D. Fonction Envoyer

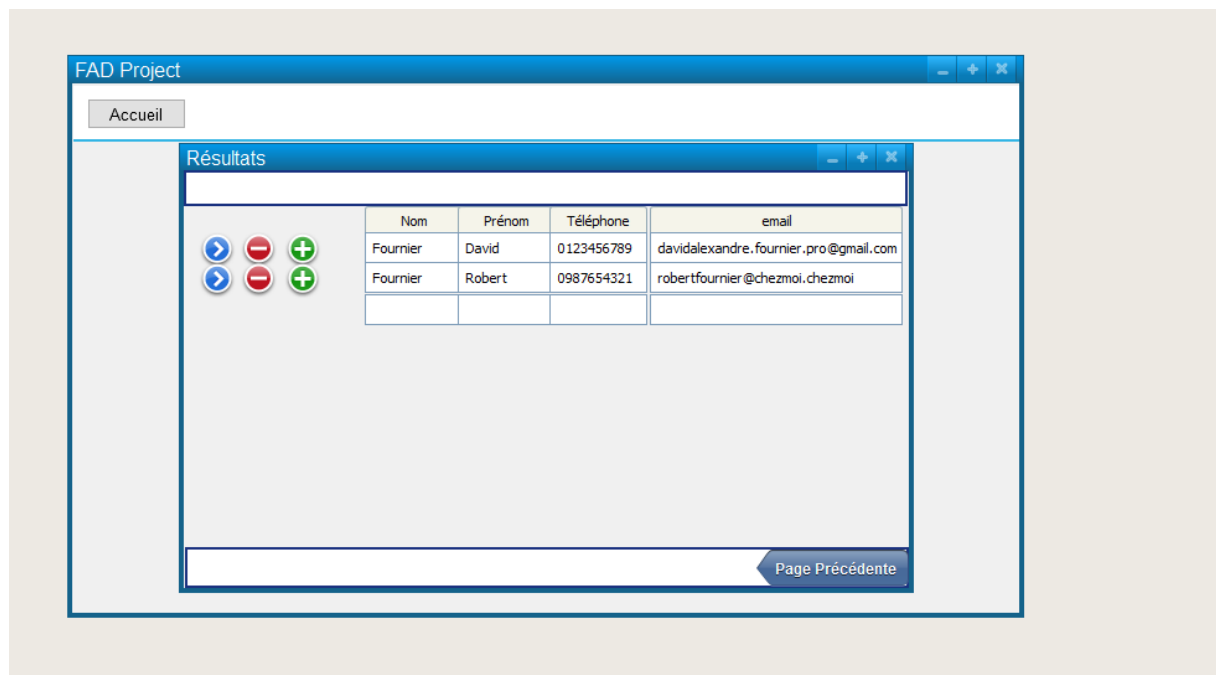
Une fois la saisie validée, l'utilisateur accède aux informations correspondantes, mais dans un souci d'affichage contenu dans la fenêtre de l'application, seuls les champs nom, prénom, téléphone et email sont présentés.

En effet, dans le cas d'adresses prenant beaucoup de places, nous afficherons le résultat dans un écran suivant.

D'ailleurs, l'utilisateur va pouvoir choisir d'une part s'il souhaite visualiser la fiche contact, la supprimer ou la modifier, grâce aux icônes prévues à cet effet.

On observe également que les résultats présentés sont fonction du nombre de contacts ayant le même nom et/ou prénom.

Un nouveau bouton apparaît (positionné en « south ») ; il permet de revenir à l'écran précédent si la saisie n'était pas celle souhaitée ou tout simplement en faire une nouvelle.



E. Fonction Lire

Une fois la recherche effectuée et la sélection par l'utilisateur du choix en fonction du ou des résultats, ce dernier pourra donc visualiser l'ensemble des données du contact.

Sur l'écran ci-dessous, on observe l'ensemble des champs et données correspondantes, la possibilité de revenir à la page précédente, et également une fonctionnalité de pouvoir supprimer ce contact directement.

The screenshot shows a window titled 'FAD Project' with a sub-window 'Contact'. The 'Contact' form has the following fields and values:

Nom	Fournier
Prénom	David
Téléphone	01 23 45 67 89
Email	davidalexandre.fournier.pro@gmail.com
Adresse	999, WestCoast Side Street Avenue
Code Postal	12345
Ville	LosAngeles

At the bottom right of the 'Contact' window, there is a red circular button with a minus sign and a blue button labeled 'Page Précédente'.

F. Fonction Ajouter

L'utilisateur va pouvoir à présent ajouter son contact avec les différents champs, sachant que seul le champ nom est obligatoire.

Il a toujours la possibilité de retourner à l'accueil, et cette fois-ci de valider son ajout ou encore retourner à l'accueil précédent.

The screenshot shows a window titled 'FAD Project' with a sub-window 'Ajout'. The 'Ajout' form has the following fields and placeholder text:

Nom	Saisissez un nom...
Prénom	Saisissez un prénom...
Mail Perso	Saisissez l'adresse mail personnel...
Mail Pro	Saisissez l'adresse mail professionnel...
Téléphone Bureau	Saisissez le téléphone du bureau...
Téléphone Domicile	Saisissez le téléphone du domicile...
Téléphone Mobile	Saisissez le téléphone portable...
Adresse Postale	Saisissez l'adresse postale...
Code Postal	Saisissez le code postal...
Ville	Saisissez la ville...

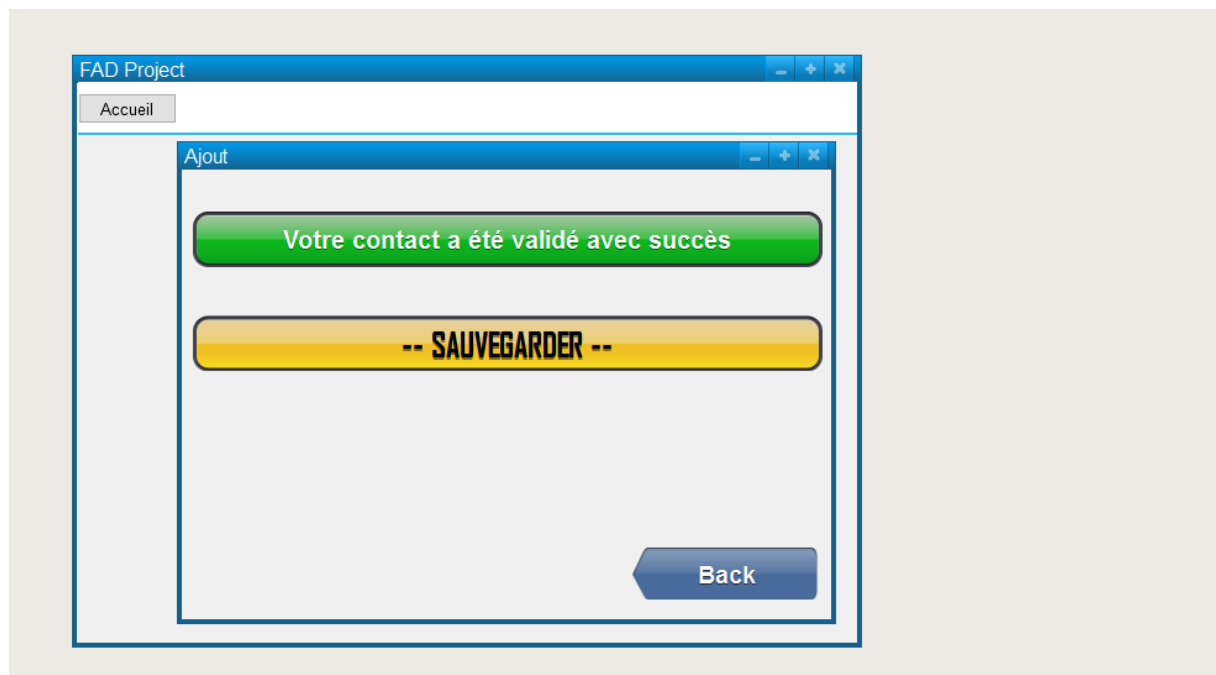
At the bottom of the 'Ajout' window, there is a green button labeled 'Valider' and a blue button labeled 'Back'.

G. Fonction Sauvegarder

A présent, l'utilisateur va pouvoir sauvegarder, bien entendu après avoir validé son ajout.

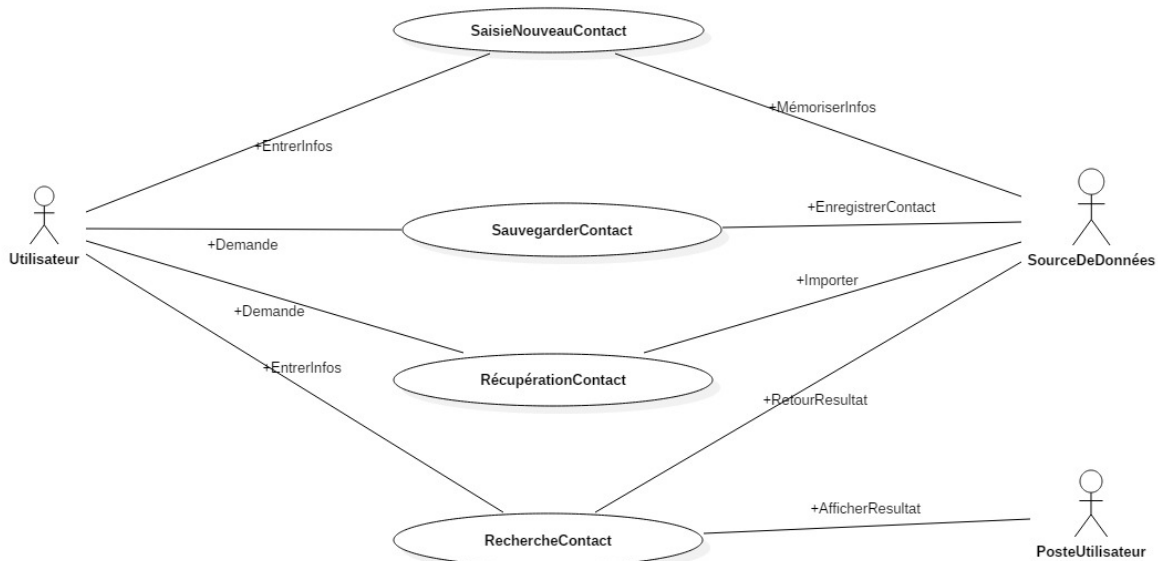
Pour ce faire, il doit donc cliquer sur le bouton sauvegarder.

Il peut toujours retourner à l'écran précédent ou vers l'accueil.



IV. Diagrammes

A. Diagramme de cas d'utilisation



Cas n°1 : SaisieNouveauContact

L'**utilisateur** *entre* les informations concernant le contact (Nom, prénom, téléphone, adresse...). Ces informations sont alors *mémorisées* dans le système (**sourceDeDonnées**) mais pas encore sauvegardées dans le carnet d'adresses.

Cas n°2 : SauvegarderContact

L'**utilisateur** *demande* à sauvegarder les informations. Le contact est enregistré dans le fichier qui va stocker les informations (**SourcedeDonnées**) et une nouvelle entrée va être créée dans le carnet.

Cas n°3 : RécupérationContact

L'**utilisateur** *demande* à récupérer une liste de contacts. Celle-ci est *importée* à partir de la **source de données**.

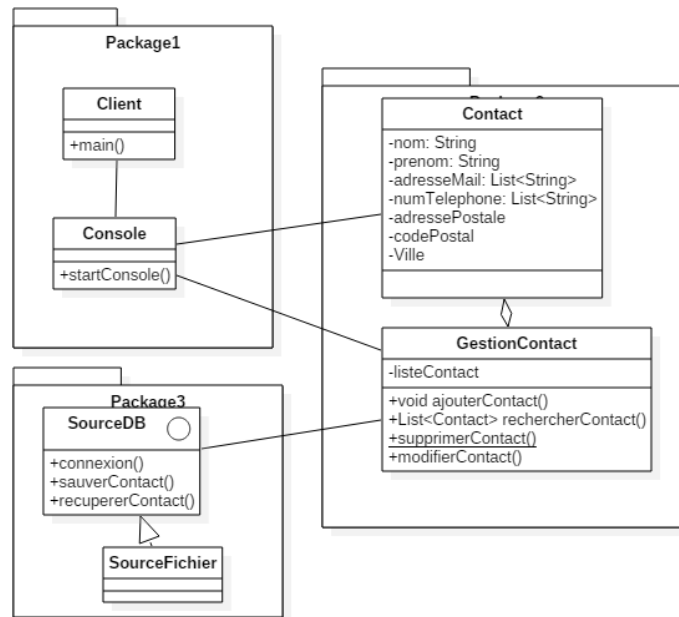
Cas n°4 : RechercheContact

L'**utilisateur** *entre* des informations dans les champs : « nom » et/ou « prénom ». La **source de données** est parcourue. Si les informations demandées sont trouvées, le/les contacts sont *affichés* sur l'écran (**poste utilisateur**).

Une fois les informations retournées, un nouveau cas d'utilisation serait alors possible : Modifier/Supprimer Contact. Mais cette fonctionnalité n'a pas encore été implémentée.

B. Diagramme de classes

Le diagramme de classe suivant montre l'application en version 1



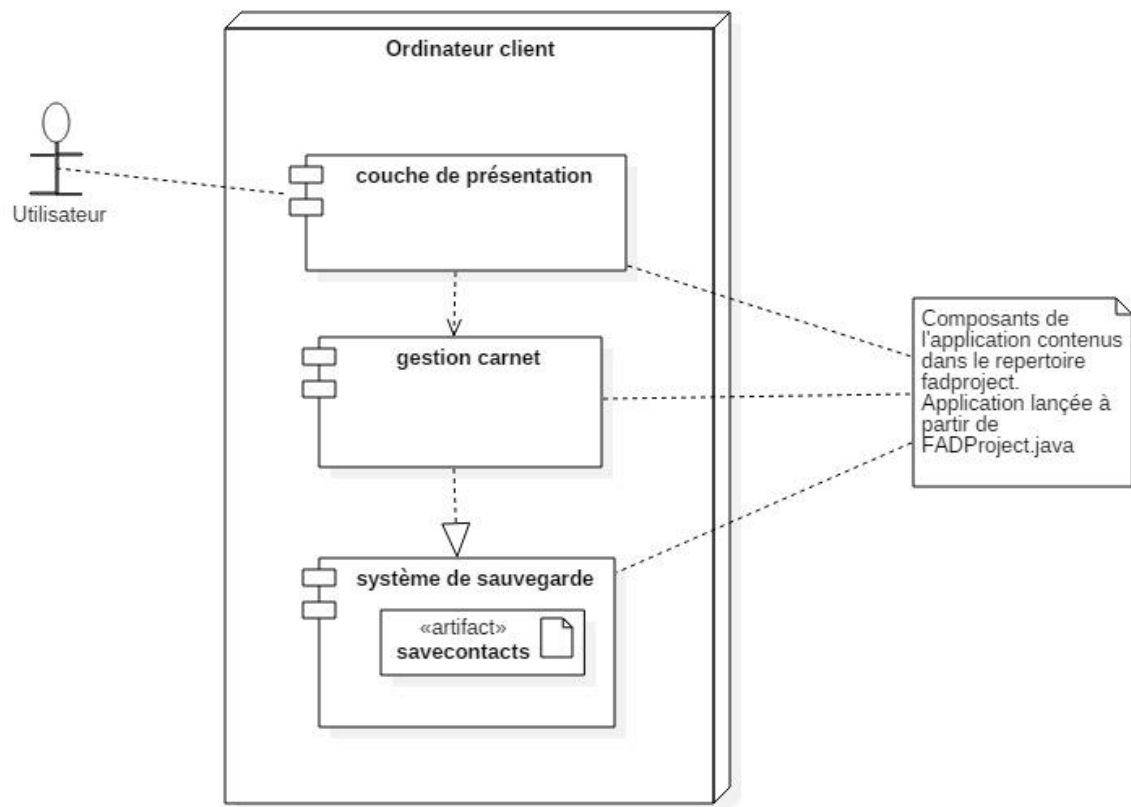
La classe contact permet de définir un objet contact. La classe GestionContact permet les opérations sur les contacts.

L'interface sourceDB permet d'avoir n'importe quel type de source tant que les méthodes sont implémentées.

Erratum :

La méthode connexion n'est pas implémentée dans l'interface sourceDB.

C. Diagramme de déploiement



Le diagramme de déploiement montre que, comme souhaité lors du cahier des charges, l'application possède peu de composants et ils sont sur le même nœud.

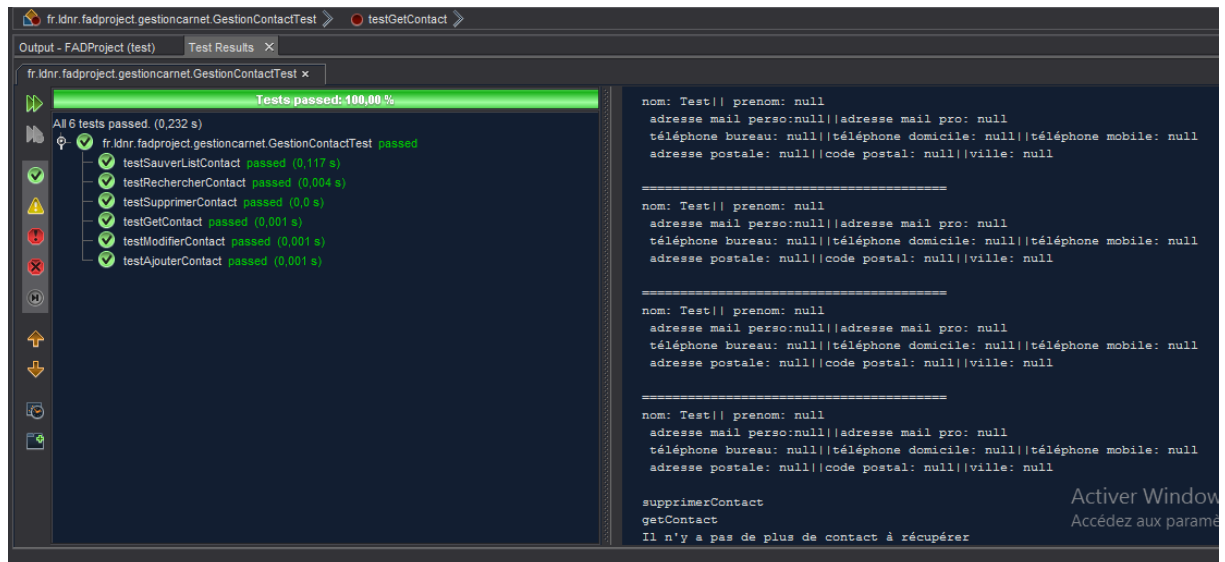
V. Annexe

A. Test Unitaires

Des tests unitaires ont été effectuées sur les classe les plus significative de l'applications :

- Contacts
- GestionContact

Voici une capture d'écran du résultat de test unitaire de GestionContact.



B. Difficultés rencontrées

Durant le projet nous avons rencontrés plusieurs difficultés :

Disponibilités des membres du projet :

La période de recherche de stage ainsi que les problèmes de connexion n'ont pas permis à ce que tous les membres du projet soient disponibles durant tout le projet. Cela a engendré une perte de 2 jours/hommes. Surtout en phase de conception

Les groupes de 3 personnes ne favorisent pas la rapidité d'exécution des tâches surtout lors des transferts de connaissances, cela monopolise 2 tiers des ressources pour un même travail.