

Equipe 5 :
Bessou Claire - chef de projet
Figue Michel
Mariette Jean

Projet Java : Carnet d'adresses

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.

Fonctionnalités détaillées :

Fonctionnalités de base :

Saisie de personnes :

Récupération des informations de la personne (nom obligatoire)
Ajout à la liste et enregistrement
Affichage confirmant ou non l'ajout

Affichage des personnes :

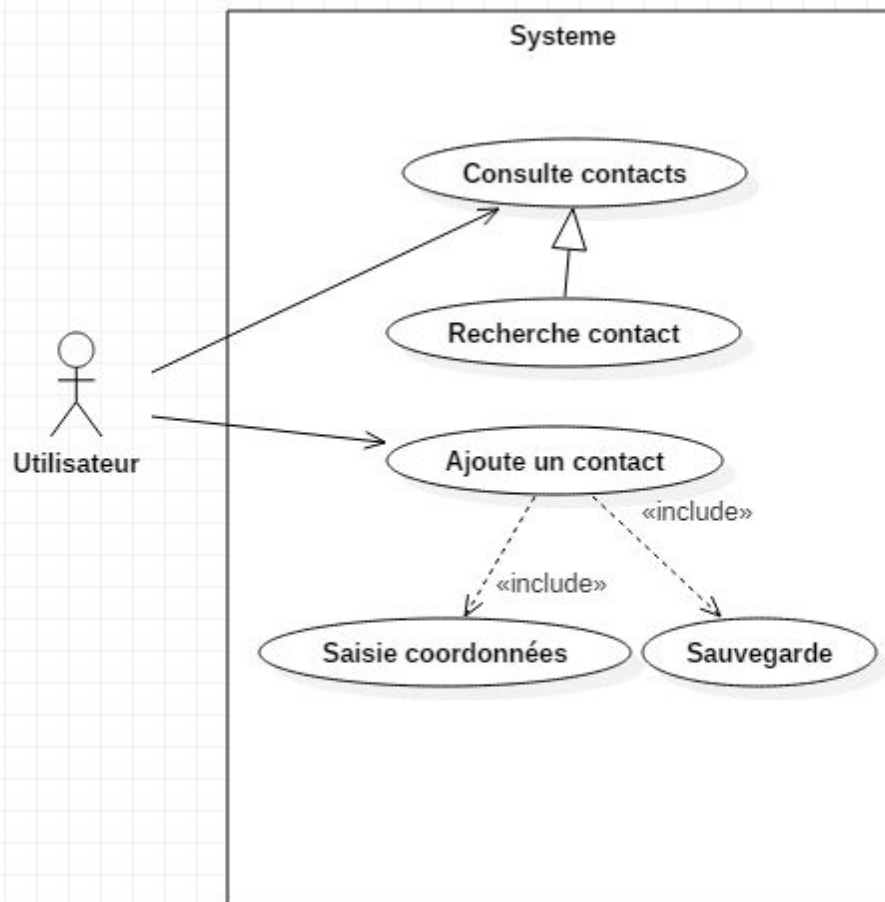
Affichage sous forme de liste alphabétique
Récupération de la chaîne saisie par l'utilisateur
Recherche des utilisateurs dont le nom contient la chaîne
Mise à jour de la liste en fonction de la recherche

Sauvegarde :

automatique quand on ajoute une personne
sauvegarde dans un fichier

Récupération :

automatique quand on ouvre l'application
récupération à partir d'un fichier

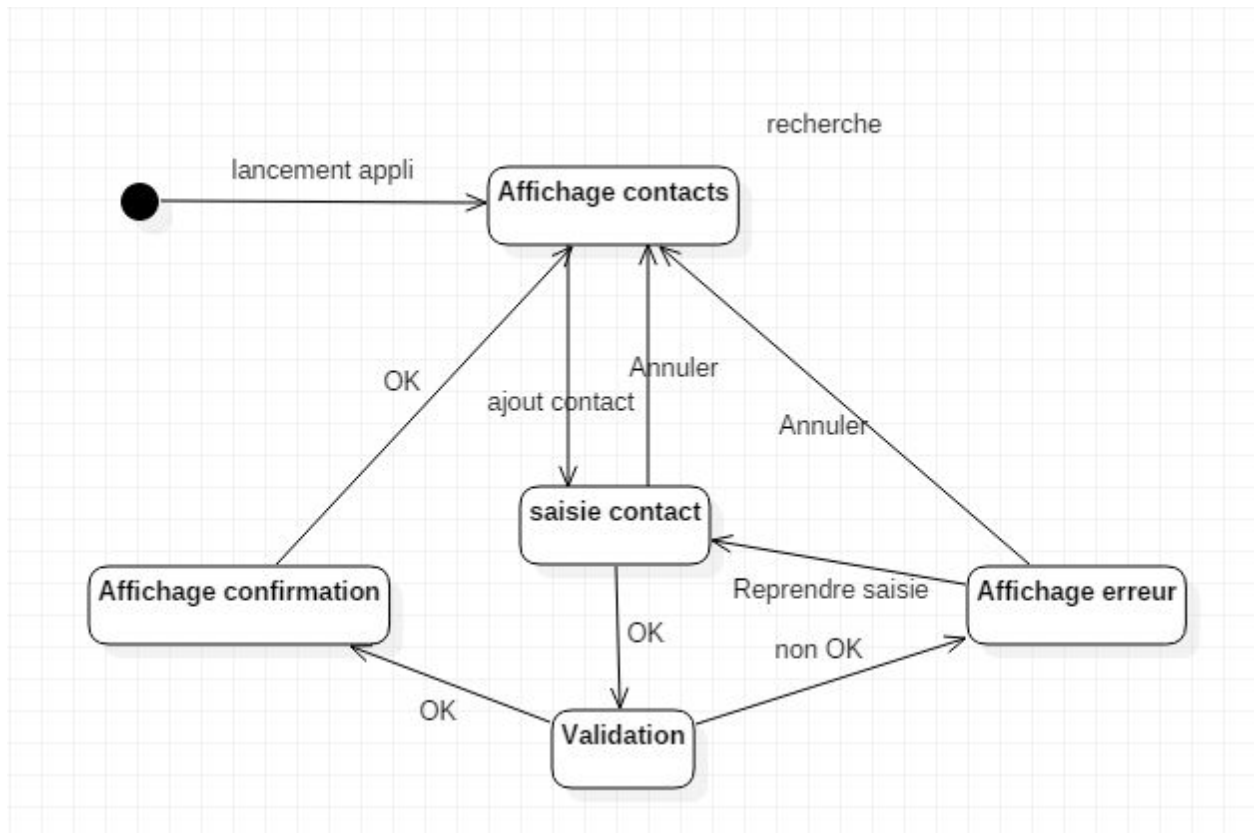


Fonctionnalités supplémentaires :

Ces fonctionnalités seront ajoutées dans l'ordre suivant l'avancement du projet

1 Ajouter une interface graphique

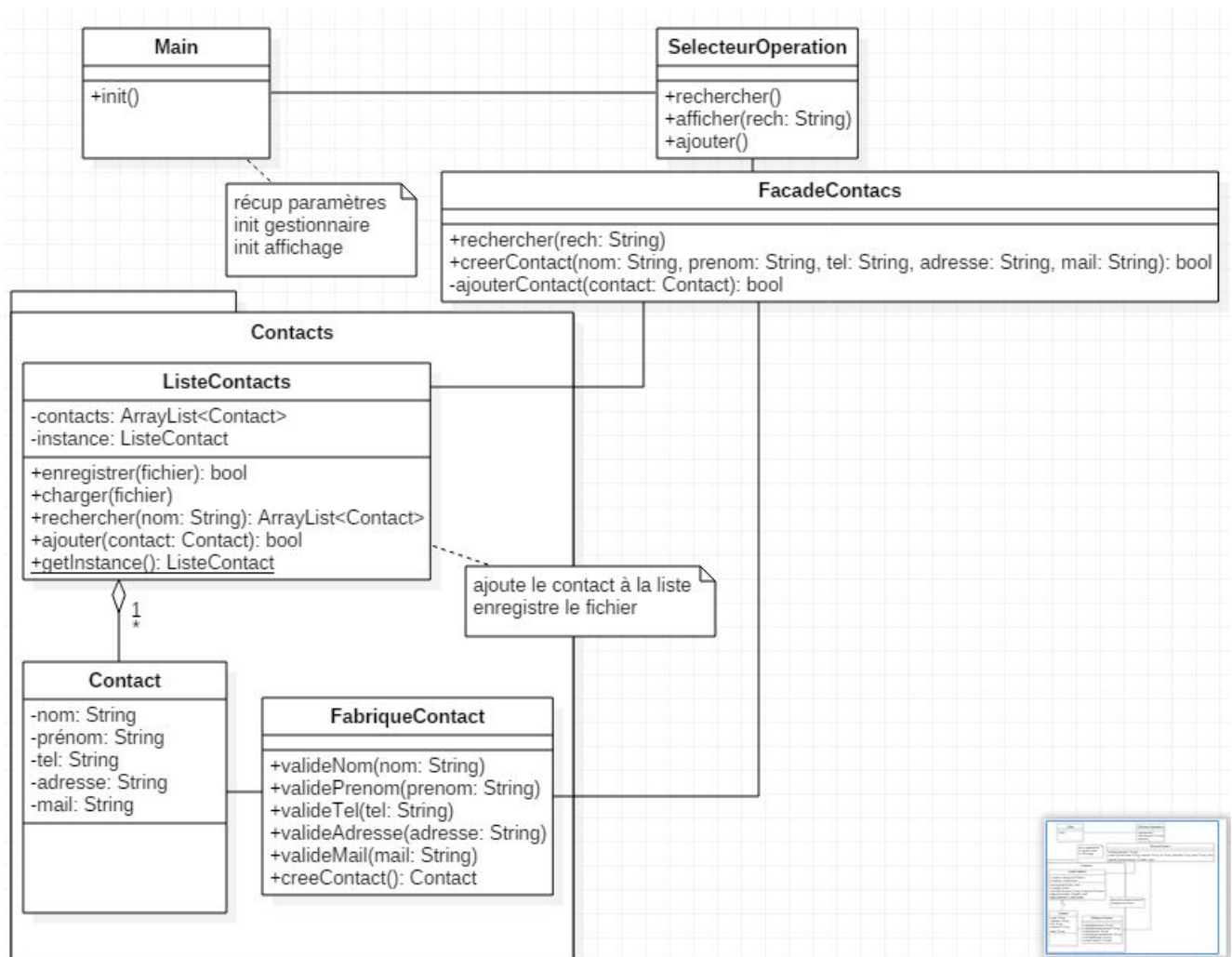
- Une fenêtre d'affichage principale contenant :
 - Un champs de recherche
 - Un bouton "recherche contact"
 - Un bouton "ajouter un contact"
 - La liste des contacts (avec scroll déroulant)
- Une fenêtre d'ajout de contact contenant :
 - Les différents champs renseignant les coordonnées du contact
 - Un bouton "Effacer" pour réinitialiser les champs
 - Un bouton "Annuler" pour revenir à la fenêtre principale
 - Un bouton "Ajouter" pour ajouter le nouveau contact
 - Une fenêtre popup "Confirmation" lorsque le contact a bien été ajouté
 - Une fenêtre popup "Erreur" si l'ajout n'as pas fonctionné



- 2 Modifier/supprimer une personne
- 3 Alléger l'affichage en liste (seulement nom, prénom, tel) et aller sur la page d'une personne pour afficher ses données complètes
- 4 Recherche sur tous les critères/choix de l'ordre d'affichage
- 5 Autocomplétion
- 6 Groupes de contacts (créer et attribuer des groupes, affichage par groupes)
- 7 Séparation des contacts (par lettre, groupe,...)
- 8 Gestion de comptes

Conception :

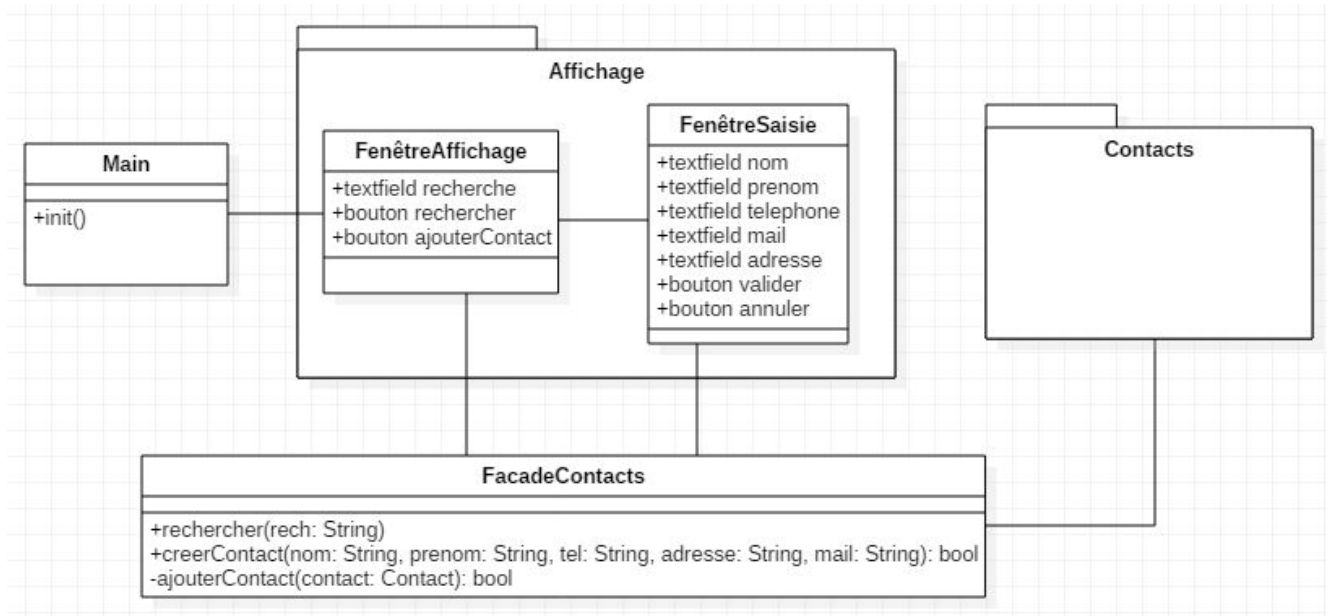
Version 0.1 :



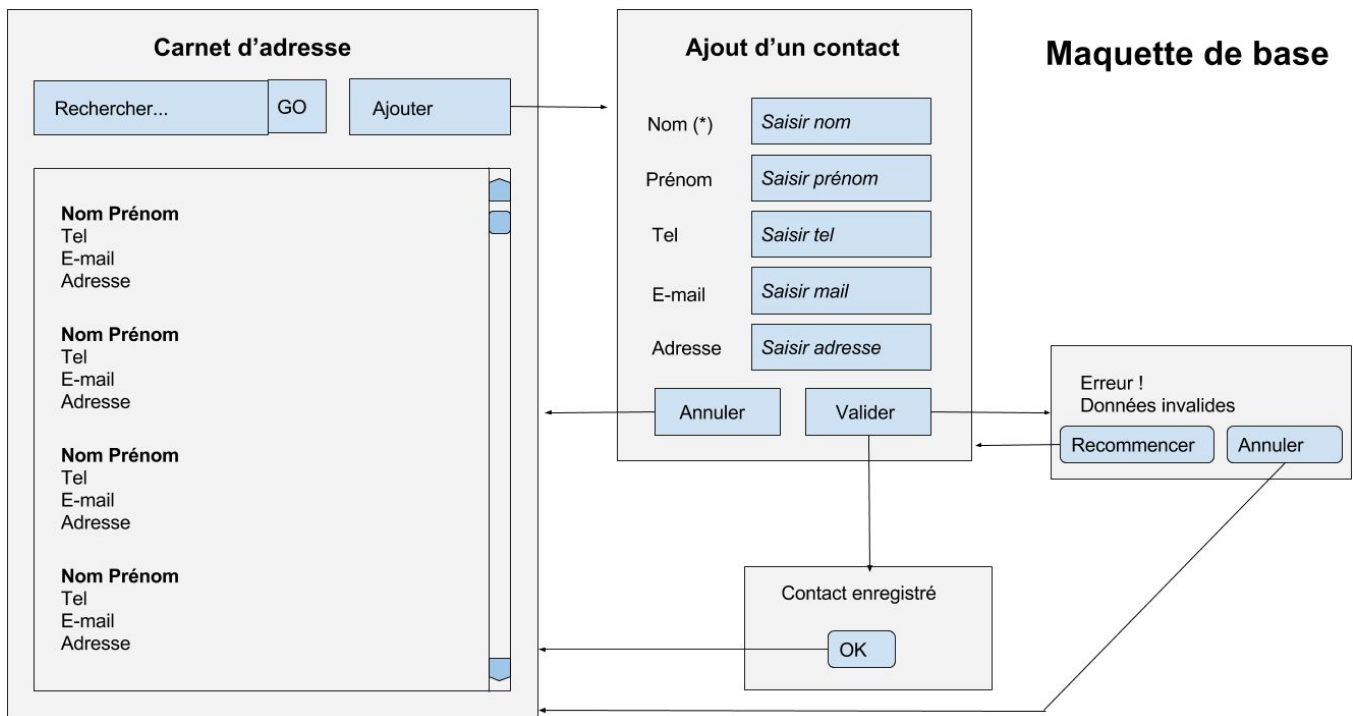
Données en entrée :

Nom : 2 à 30 caractères alphabétiques, tirets, apostrophes, espaces
 Prénom : 2 à 30 caractères alphabétiques, tirets, apostrophes, espaces
 Téléphone : 10 à 14 caractères numériques, tirets, espaces, points
 Mail : 254 caractères
 Adresse : 300 caractères alphanumériques, tirets, apostrophes, espaces, points
 CP : 4 à 5 caractères numériques
 Ville : 45 caractères alphabétiques, tirets, apostrophes, espaces, points

Version 1.0 :



Story board



Planning Prévisionnel :

Ce planning prévisionnel décrit de manière succincte les étapes de développement du projet.

Ainsi il sera séparé en demi journée de travail.

Jour 1 (27/03) :

Première demi journée:

Tutorat

Deuxième demi-journée:

Travail en commun.

Consacrée à l'étude du projet et de ses complexités. On en profite pour faire le point sur ce que l'on aura besoin comme connaissances sur le sujet et qui sait faire quoi.

On définit également les outils qui seront nécessaire pour la réalisation de notre projet (communication, échange de fichiers, gestion de version, etc...).

Jour 2 (28/03)

Première demi journée.

Travail en commun.

Consacrée à la conception de notre application.

Discussion sur la manière dont celle-ci va s'exécuter pour un client.

Création d'un diagramme de cas d'utilisation.

Création de la maquette afin d'avoir un visuel.

Définition des différentes classes nécessaires.

Deuxième demi journée.

Consacrée à la définition des classes, et création du diagramme de classe.

Puis Création des différentes classes et de leur méthodes définies dans le diagramme.

Tâches:

Jean : absent.

Réalisation de la classe Contact : Michel

Réalisation de la classe SelecteurOperation + FacadeContacts : Claire

Jour 3 (29/03)

Première demi journée.

Réalisation de la classe FabriqueContacts + tests unitaires : Jean

Réalisation de la classe ListeContacts : Michel, Jean

Réalisation de la classe SelecteurOperation + FacadeContacts : Claire

Deuxième demi journée

Consacrée à l'implémentation de l'interface graphique.

Tâches:

Jean : Création de la fenêtre d'affichage.

Michel, Claire : Création de la fenêtre de saisie.

Jour 4 (30/03)

Première demi journée.

Ajout de fonctionnalités

Tâches:

Jean : Modification de la ListeContacts pour ajouter la suppression de contacts

Michel : Modification de l'interface graphique

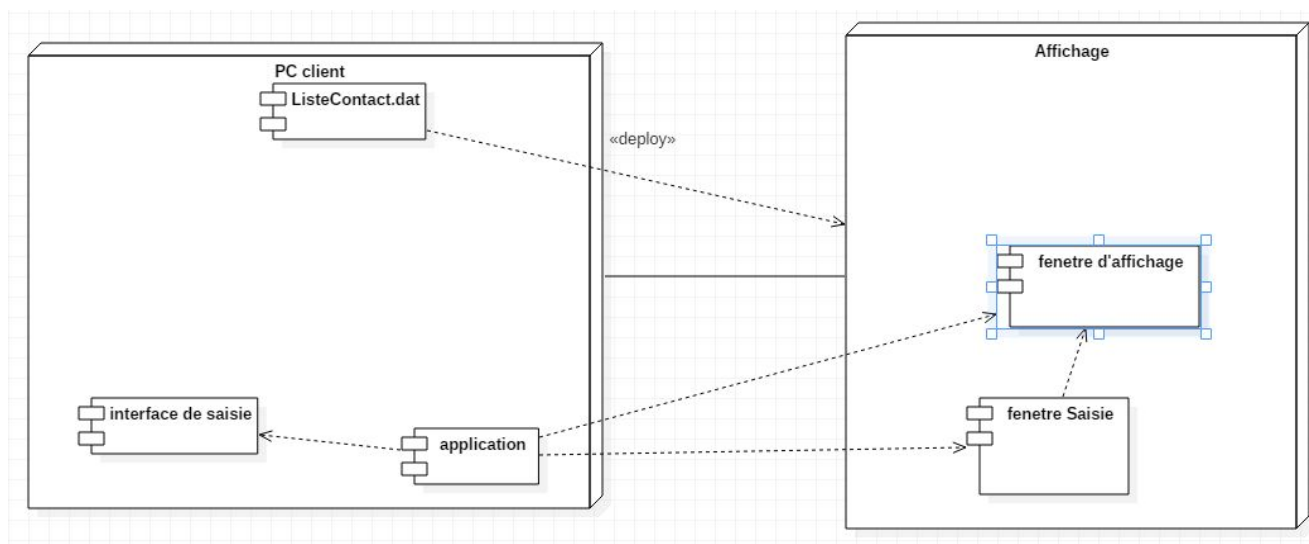
Claire : Modification de la façade

Deuxième demi journée.

Fin de la création des classes + tests unitaires et tests complet de l'application.

Fin de la documentation.

Diagramme de déploiement



Résultat des tests :

Tests unitaires de FabriqueContact

The screenshot shows the Test Results window for the 'CarnetAdresses' project. The 'contacts.FabriqueContactTest' suite is expanded, showing 8 tests passed. A tooltip is visible over the 'testValideVille' test, listing its sub-tests: 'valideVille', 'Regex de la ville validée', 'valideMail', 'Regex du mail validée', 'valideCP', 'Regex du code postal validée', 'valideNom', 'Regex du Nom Validée', 'valideTel', 'Regex du numéro de téléphone validée', 'validePrenom', 'Regex du prénom réussie', 'creeContact', 'Contact créé!', 'nom : testNom', 'prénom : testPrénom', 'Tel : 00-00.00-00.00', 'adresse : test de l'adresse', 'Code Postal : 31000', 'Ville : Test ville', 'Mail : test@exemple.test', 'valideAdresse', and 'Regex de l'adresse validée'.

Test Results X
CarnetAdresses X

Tests passed: 100,00 %

All 8 tests passed. (0,075 s)

- contacts.FabriqueContactTest passed
 - testValideVille passed (0,002 s)
 - valideVille
 - Regex de la ville validée
 - valideMail
 - Regex du mail validée
 - valideCP
 - Regex du code postal validée
 - valideNom
 - Regex du Nom Validée
 - valideTel
 - Regex du numéro de téléphone validée
 - validePrenom
 - Regex du prénom réussie
 - creeContact
 - Contact créé!
 - nom : testNom
 - prénom : testPrénom
 - Tel : 00-00.00-00.00
 - adresse : test de l'adresse
 - Code Postal : 31000
 - Ville : Test ville
 - Mail : test@exemple.test
 - valideAdresse
 - Regex de l'adresse validée

Tests unitaires de ListeContacts

The screenshot shows the Test Results window for the 'CarnetAdresses' project. The 'contacts.ListeContactsTest' suite is expanded, showing 6 tests passed. The tests are: 'testEnregistrer' (0,059 s), 'testAjouter' (0,002 s), 'testToString' (0,001 s), 'testCharger' (0,002 s), 'testGetInstance' (0,0 s), and 'testRechercher' (0,0 s). The right pane shows the output for each test: 'enregistrer' (Enregistrement réussi), 'ajouter' (Test ajout Réussi.), 'toString' (Méthode toString fonctionnelle.), 'charger' (Chargement fichier réussi.), 'getInstance' (Instance trouvée), and 'rechercher' (Test recherche Réussi.).

Test Results X
CarnetAdresses X

Tests passed: 100,00 %

All 6 tests passed. (0,129 s)

- contacts.ListeContactsTest passed
 - testEnregistrer passed (0,059 s)
 - testAjouter passed (0,002 s)
 - testToString passed (0,001 s)
 - testCharger passed (0,002 s)
 - testGetInstance passed (0,0 s)
 - testRechercher passed (0,0 s)

enregistrer
Enregistrement réussi

ajouter
Test ajout Réussi.

toString
Méthode toString fonctionnelle.

charger
Chargement fichier réussi.

getInstance
Instance trouvée

rechercher
Test recherche Réussi.