Concepteur Développeur Informatique

Chapitre 06-08 : Les Entrées et Sorties en Java

Ce document a été co-rédigé par les personnes suivantes :

1- BOUHBEL Azzedine
2- DA ROCHA Manuel
3- DESCHAMPS Francis
4- DORE Emryck
5- DURINGER Gaspard
6- EL FATEOUI Najim
7- FLAMAND Kevin
8- GARNIER Thomas
9- LAVIGNON Baptiste
10- MARTIN Gillaume
11- MASSE Auriane
12- MOHAMED Amine
13- PLOUCHARD Cédric
14- RAJAOMAZAVA Dina
15- SALAME Alexandre
16- SILOTIA Gilles
17- VUILLAUME Laureene

Avec l'assistance de DALGALIAN Tcharou

Sommaire

(1.) Cahier des charges

(2.) Analyse des données et analyse fonctionnelle

(3.) Conception

(4.) Réalisation

(5.) Mise en service

(6.) Synthèse

1.)Le cahier des charges

L'objectif de ce document est de concevoir et de réaliser une application console possédant les fonctionnalités suivantes :

(01.) Message de bienvenue :

Afficher un message de bienvenue

(02.) Menu principal :

Afficher un menu principal comportant les entrées suivantes :

- (1) - Enregistrer une personne
- (2) - Rechercher une personne (par son identifiant).
- (3) - Rechercher la liste de toutes les personnes
- (4) - Modifier une personne
- (5) - Supprimer une personne

1.)Le cahier des charges (suite)

Liste des fonctionnalités demandées (suite) :

(03.) Enregistrer une personne :

Effectuer la séquence suivante :

- Proposer à l'utilisateur de saisir les informations relatives à une personne (identifiant exclu).
- Sauvegarder dans les donnée persistantes les informations saisies.
- Afficher les données relatives à la personne saisie (identifiant inclus)

(04.) Rechercher une personne (par son identifiant) :

Effectuer la séquence suivante :

- Proposer à l'utilisateur de saisir l'identifiant de la personne à rechercher.
- Rechercher dans les données persistantes l'identifiant saisi.
- Récupérer et afficher les données personnelles de la personne trouvée (identifiant inclus).

1.)Le cahier des charges (suite)

Liste des fonctionnalités demandées (suite) :

(05.) Modifier une personne :

Effectuer les tâches suivantes :

- Proposer à l'utilisateur de saisir les données personnelles de la personne à modifier (identifiant inclus).
- Rechercher dans les données persistantes l'identifiant saisi.
- Modifier les données personnelles de la personne trouvée. (identifiant exclu)
- Afficher les données personnelles de la personne modifiée. (identifiant inclus)

1.)Le cahier des charges (suite)

Liste des fonctionnalités demandées (suite) :

(06.) Supprimer une personne :

Effectuer les tâches suivantes :

- Proposer à l'utilisateur de saisir l'identifiant de la personne à rechercher.
- Rechercher dans les données persistantes l'identifiant saisi.
- Supprimer les données personnelles de la personne trouvée.
- Afficher les données personnelles de la personne supprimée. (identifiant inclus)

1.)Le cahier des charges (suite)

(07.) Données saisies :

Les informations personnelles saisies par l'étudiant sont définies ci-dessous :

- Le nom : de type chaîne de caractères.
- Le prénom : de type chaîne de caractères.
- L'âge : de type entier.

(08.) Données stockées (dans les données persistantes) :

Les informations personnelles à stocker dans les données persistantes sont définies ci-dessous :

- L'identifiant : de type 'long'.
- Le nom : de type 'String'.
- Le prénom : de type 'String'.
- L'âge : de type 'int'.

2.)Analyse de données et analyse fonctionnelle

La phase d'analyse de données et d'analyse fonctionnelle consiste à produire les documents suivants :

(2.1.) Le dictionnaire des données.

(2.2.) Le Modèle Conceptuel de Données (celui-ci sera omis dans ce projet).

(2.3.) Le diagramme des cas d'utilisation.

(2.4.) Le diagramme de séquence (boite noire).

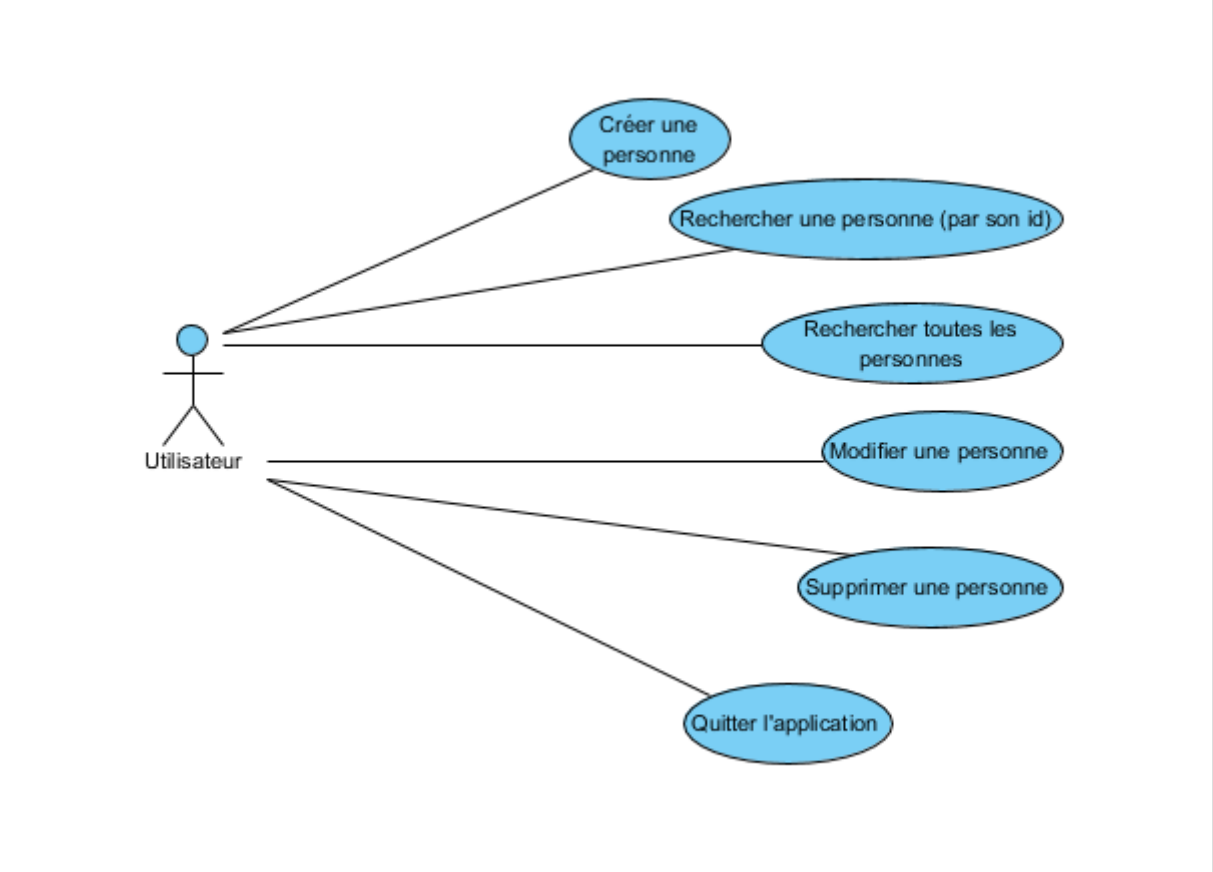
2.1.)Le dictionnaire des données

.

ENTITE	ATTRIBUTS	TYPE
Personne	id	long
	nom	String
	prenom	String
	age	int
Counter	idAvalaible	long

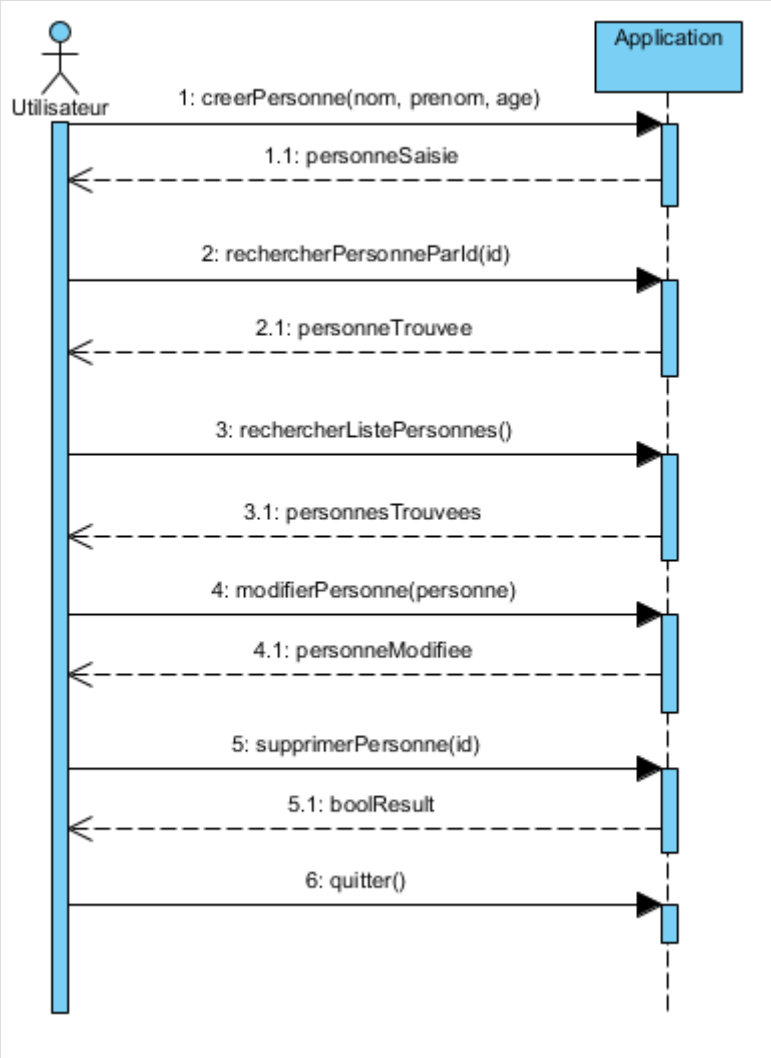
2.3.)Le diagramme des cas d'utilisation

.



2.4.)Le diagramme de séquence (boite noire)

.



3.)Conception

La phase de conception consiste à produire les documents suivants :

(3.1.) Les diagrammes de séquence (boite blanche).

(3.2.) Les algorithmes de méthodes particulières (uniquement des cas particuliers).

(3.3.) Le diagramme des classes.

4.)Réalisation

La phase de réalisation consiste à produire une application dont les fonctionnalités sont conformes aux phases précédentes :

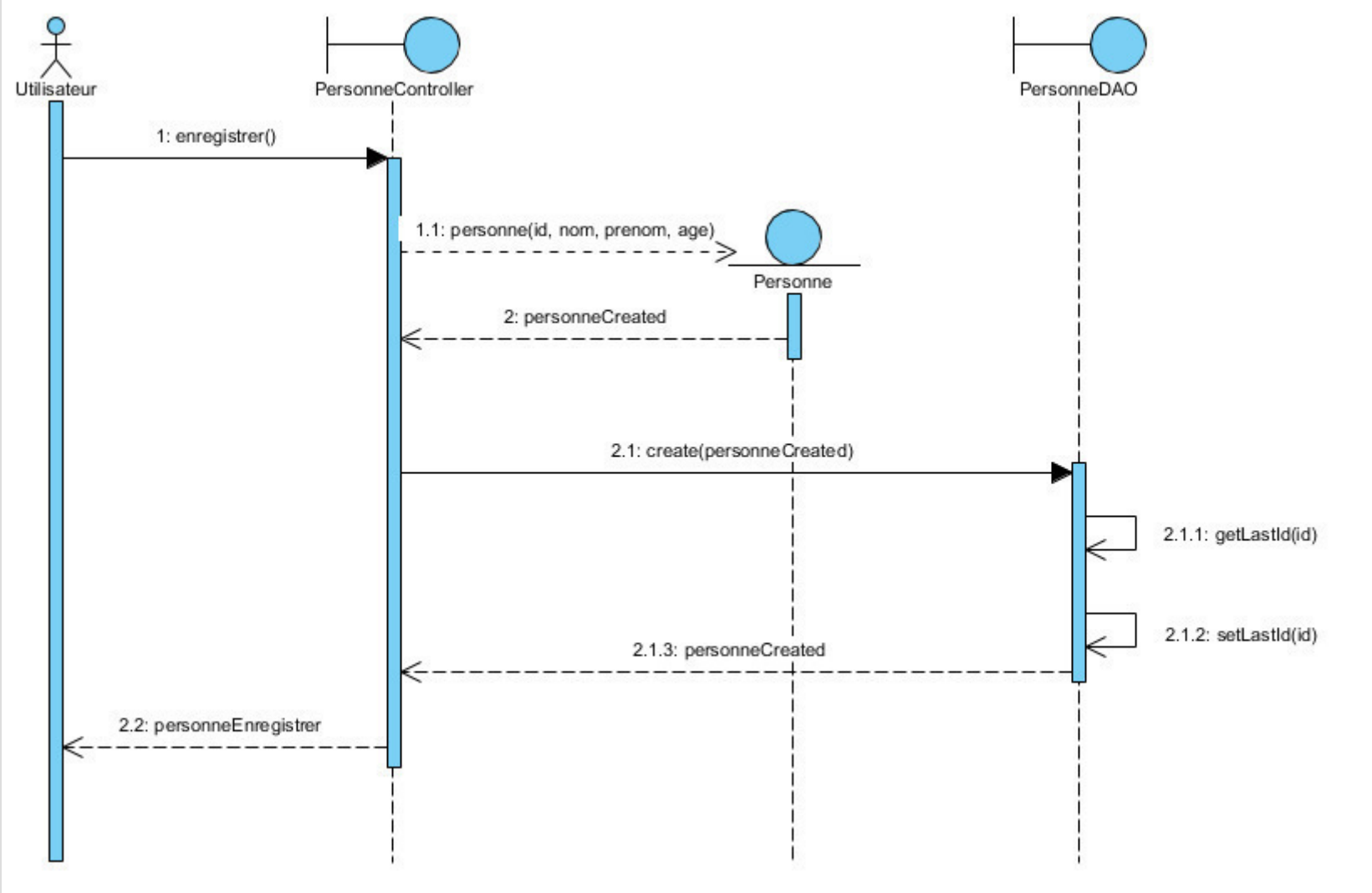
Cette phase de réalisation comportera les parties suivantes :

(4.1.) L'architecture applicative (arborescence des packages et couches applicatives)

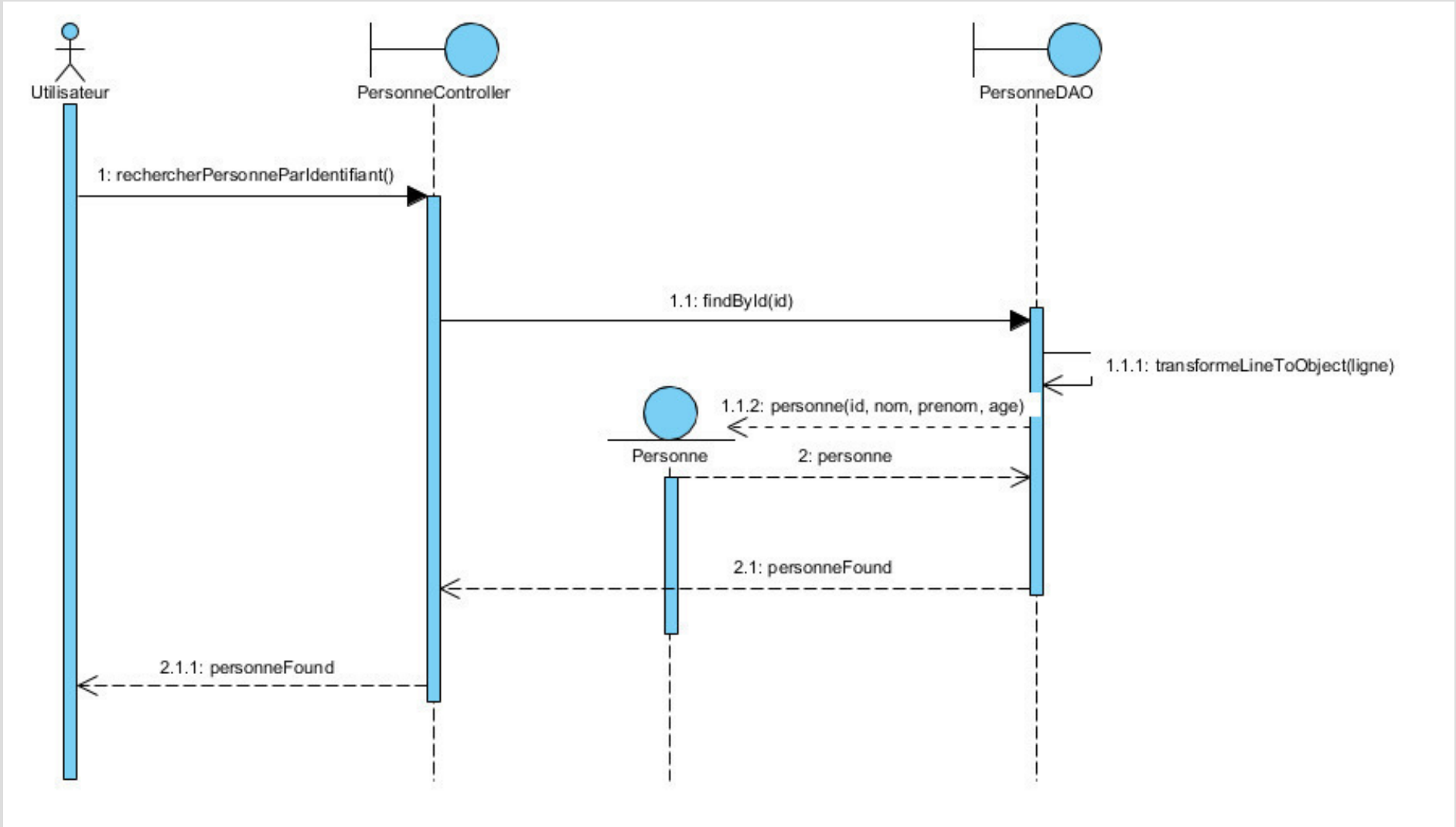
(4.2.) Les composants applicatifs (uniquement des cas particuliers)

(4.3.) Les tests à effectuer sur l'application

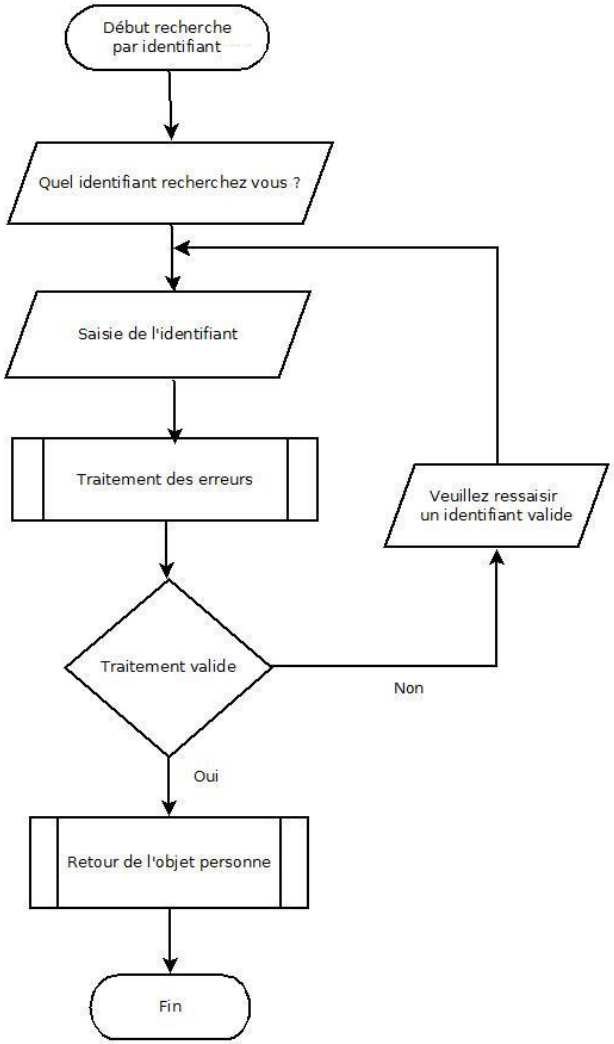
3.1.)Le diagramme de séquence (Créer personne)

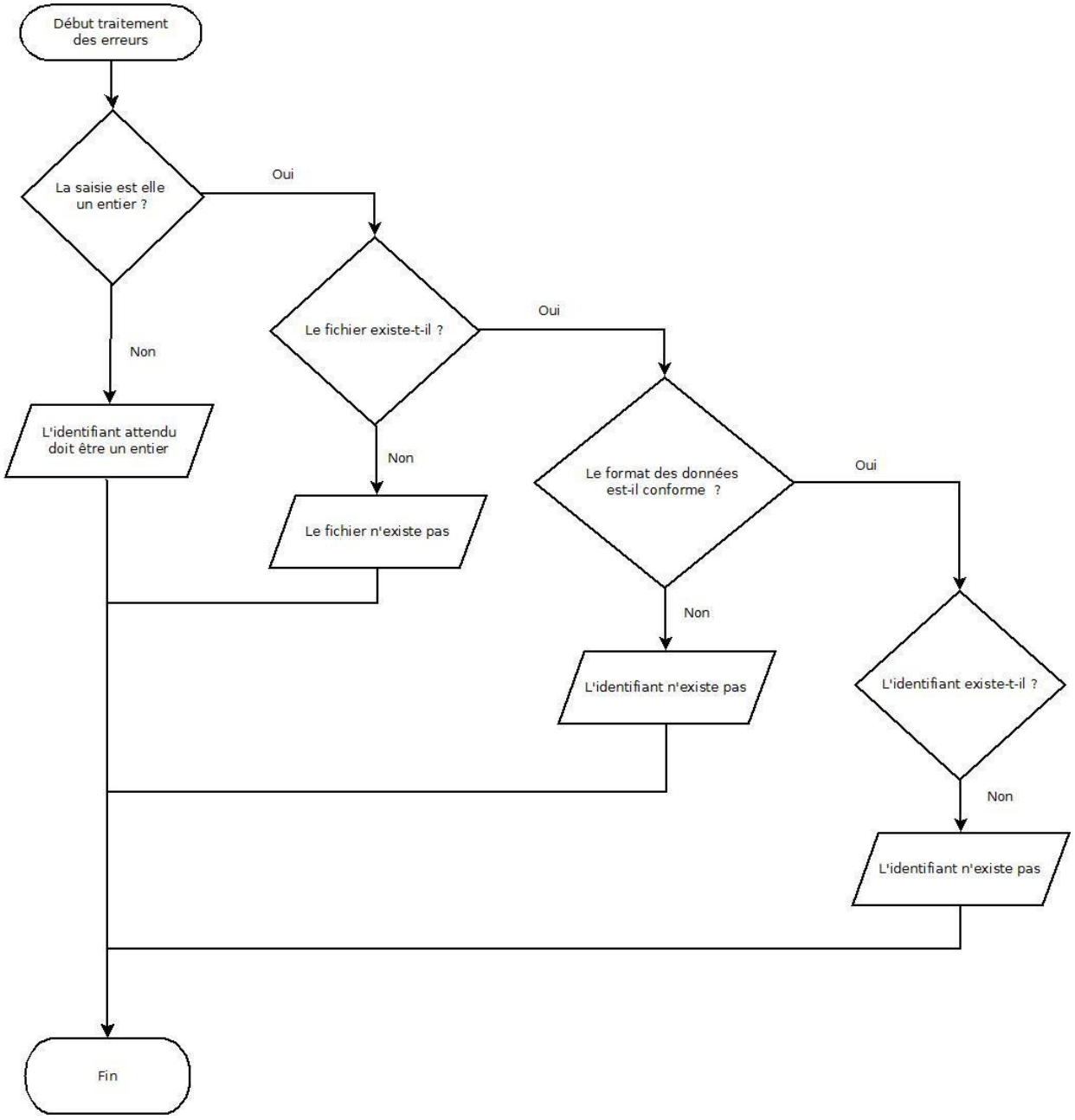


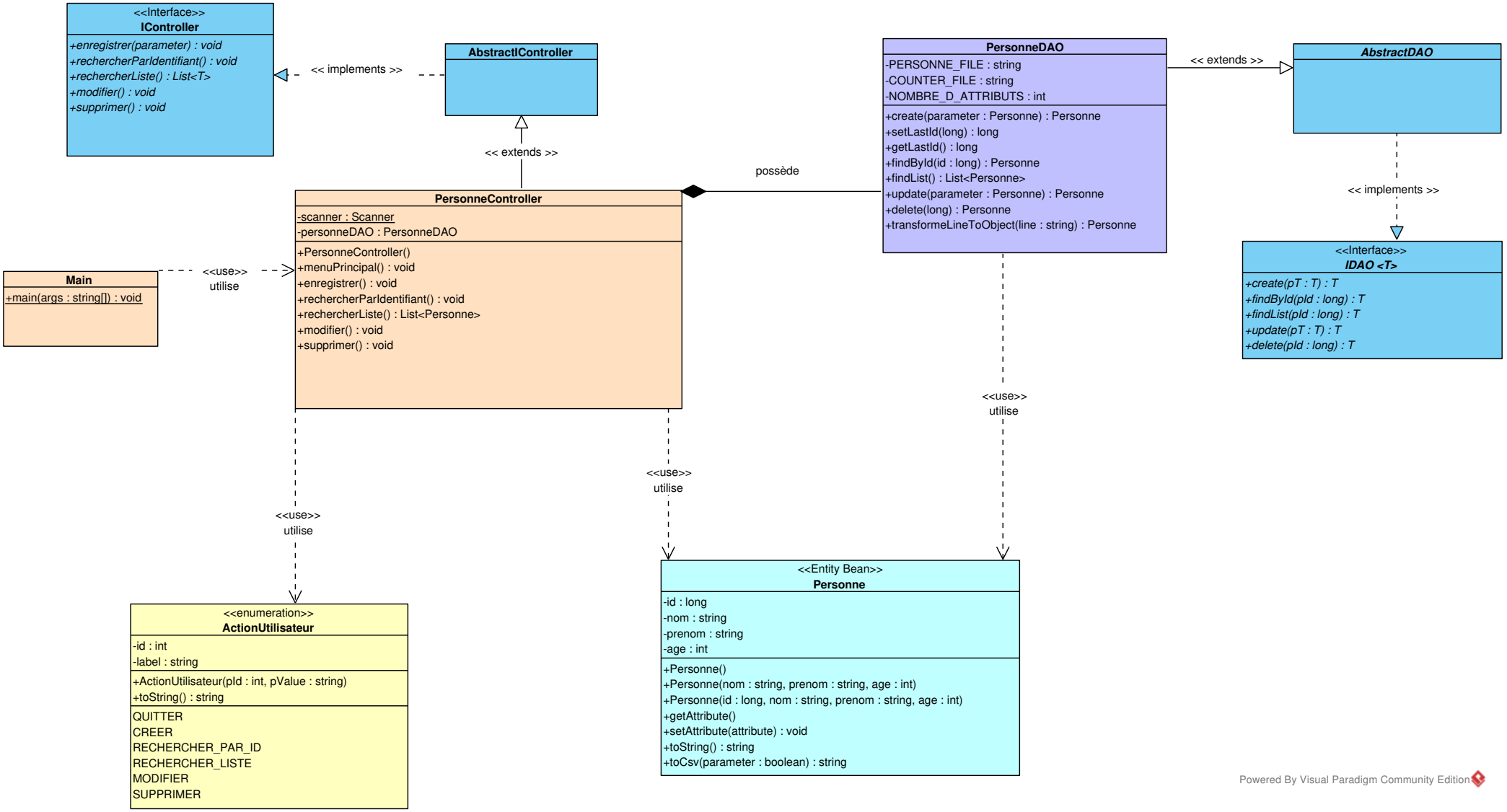
3.1.)Le diagramme de séquence (Rechercher par Id)



ALGORIGRAMME DE LA RECHERCHE PAR IDENTIFIANT D'UN ETUDIANT






<<Interface>>
IController
+enregistrer(parameter) : void
+rechercherParIdentifiant() : void
+rechercherListe() : List<T>
+modifier() : void
+supprimer() : void
<< implements >>
AbstractIController
<< extends >>
PersonneDAO
-PERSONNE_FILE : string
-COUNTER_FILE : string
-NOMBRE_D_ATTRIBUTS : int
+create(parameter : Personne) : Personne
+setLastId(long) : long
+getLastId() : long
+findById(id : long) : Personne
+findList() : List<Personne>
+update(parameter : Personne) : Personne
+delete(long) : Personne
+transformeLineToObject(line : string) : Personne
<< extends >>
AbstractDAO
<< implements >>
<<Interface>>
IDAO <T>
+create(pT : T) : T
+findById(pId : long) : T
+findList(pId : long) : T
+update(pT : T) : T
+delete(pId : long) : T
possède
PersonneController
-scanner : Scanner
-personneDAO : PersonneDAO
+PersonneController()
+menuPrincipal() : void
+enregistrer() : void
+rechercherParIdentifiant() : void
+rechercherListe() : List<Personne>
+modifier() : void
+supprimer() : void
Main
+main(args : string[]) : void
<<use>>
utilise
<<use>>
utilise
<<use>>
utilise
<<use>>
utilise
<<enumeration>>
ActionUtilisateur
-id : int
-label : string
+ActionUtilisateur(pId : int, pValue : string)
+toString() : string
QUITTER
CREER
RECHERCHER_PAR_ID
RECHERCHER_LISTE
MODIFIER
SUPPRIMER
<<Entity Bean>>
Personne
-id : long
-nom : string
-prenom : string
-age : int
+Personne()
+Personne(nom : string, prenom : string, age : int)
+Personne(id : long, nom : string, prenom : string, age : int)
+getAttribute()
+setAttribute(attribute) : void
+toString() : string
+toCsv(parameter : boolean) : string
Powered By Visual Paradigm Community Edition

4.1.)L'architecture applicative

- L'arborescence des répertoires du projet sera indiquera les noms et emplacement de ces répertoires.
- Cette arborescence est soumise à des conventions de découpage et de nommage.

Elle se présentera sous la forme suivante :

```
[Répertoire racine du projet]
          |
          +-->src
          |    |
          |    +-->business
          |    |    |
          |    |    +-->Exception
          |    |
          |    +-->persistence
          |    |    |
          |    |    +-->dao
          |    |    |
          |    |    +-->entity
          |    |
          |    +-->ui
          |    |    |
          |    |    +-->controller
          |
          +-->data
          |    |
          |    +-->counter.csv
          |    |
          |    +-->personne.csv
```

4.2.)Les composants applicatifs

La réalisation des composants applicatifs sera illustrée par quelques exemples (choisis par le développeur) :

```
public void rechercherParIdentifiant() {

    Personne personneFound = null;

    try {
       System.out.println("Quel identifiant recherchez vous ?");

       long pId = scanner.nextInt();

       personneFound = this.personneDAO.findById(pId);
       System.out.println(personneFound.toString());

    } catch (InputMismatchException e) {
       System.out.println("L'identifiant doit être un chiffre");
       scanner.nextLine();
    } catch (DAOException e) {
       System.out.println("Rechercher une personne par identifiant -- " + e.getMessage());

    } catch (EntityNotFoundException e) {
       System.out.println("Rechercher une personne par identifiant -- " + e.getMessage());
    }
 }
```

4.2.)Les composants applicatifs

La réalisation des composants applicatifs sera illustrée par quelques exemples (choisis par le développeur) :

```
public Personne findById long pId) throws EntityNotFoundException, DAOException {

    Path path = Paths.get(PERSONNE_FILE);
    String myLine = null;
    try (Stream<String> lines = Files.lines(path)) {

       myLine = lines.filter(line -> (line.split(";")[0]).equals("" + pId)).findFirst()
             .orElse(null);

    } catch (NoSuchFileException e) {
       System.out.println("Le fichier n'a pas été trouvé -- " + e.getMessage());

    } catch (Exception e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
    }

    if(myLine == null) {
       throw new EntityNotFoundException("Personne non trouvée");
    }

    Personne personne = this.tranformLineToObject(myLine);

    return personne;
  }
```

4.2.)Les composants applicatifs

La réalisation des composants applicatifs sera illustrée par quelques exemples (choisis par le développeur) :

```
public Personne tranformLineToObject String line) throws DAOException {

    String[] elements = line.split("\\;");

    if((elements == null) || (elements.length < PersonneDAO.NOMBRE_D_ATTRIBUTS)) {
       throw new DAOException("Erreur d'accès au fichier (lecture / écriture");
    }

    long l = Long.valueOf(elements[0]);
    int i = Integer.valueOf(elements[3]);
    Personne personne = new Personne(l, elements[1], elements[2], i);

    return personne;
  }

 }
```

5.)Mise en service :

.

```
Veuillez choisir une option :
[0] -- [Quitter l'application]
[1] -- [Créer une personne]
[2] -- [Rechercher une personne par id]
[3] -- [Afficher toutes les personnes]
[4] -- [Modifier une personne]
[5] -- [Supprimer une personne]
fff
la saisie doit etre numérique
Veuillez choisir une option :
[0] -- [Quitter l'application]
[1] -- [Créer une personne]
[2] -- [Rechercher une personne par id]
[3] -- [Afficher toutes les personnes]
[4] -- [Modifier une personne]
[5] -- [Supprimer une personne]
1
+-------------------------------------------------------------+
| VEUILLEZ SAISIR LES INFORMATIONS RELATIVES A LA PERSONNE : |
+-------------------------------------------------------------+
Veuillez saisir un nom :
toto
Veuillez saisir un prenom :
titi
Veuillez saisir un age :
fgf
Erreur de saisie
```

```
Veuillez choisir une option :
[0] -- [Quitter l'application]
[1] -- [Créer une personne]
[2] -- [Rechercher une personne par id]
[3] -- [Afficher toutes les personnes]
[4] -- [Modifier une personne]
[5] -- [Supprimer une personne]
2
Quel identifiant recherchez vous ?
dd
L'identifiant doit être un chiffre
Veuillez choisir une option :
[0] -- [Quitter l'application]
[1] -- [Créer une personne]
[2] -- [Rechercher une personne par id]
[3] -- [Afficher toutes les personnes]
[4] -- [Modifier une personne]
[5] -- [Supprimer une personne]
2
Quel identifiant recherchez vous ?
56
Rechercher une personne par identifiant -- Personne non trouvée
Veuillez choisir une option :
[0] -- [Quitter l'application]
[1] -- [Créer une personne]
[2] -- [Rechercher une personne par id]
[3] -- [Afficher toutes les personnes]
[4] -- [Modifier une personne]
[5] -- [Supprimer une personne]
2
Quel identifiant recherchez vous ?
5
Personne [id=5, nom=duringer, prenom=gaspard, age=30]
Veuillez choisir une option :
[0] -- [Quitter l'application]
[1] -- [Créer une personne]
[2] -- [Rechercher une personne par id]
[3] -- [Afficher toutes les personnes]
[4] -- [Modifier une personne]
[5] -- [Supprimer une personne]
```

5.)Conclusion et synthèse :

Ce projet nous a permis d'approfondir nos notions en programmation orienté objet. Notamment les classes, l'héritage, les collections, les interfaces ainsi que l'UML (cf. diagramme de séquence, diagramme de classe).

Nous avons eu pour objectif dans ce projet d'inscrire un ou plusieurs étudiants dans un fichier CSV et dans faire la recherche par rapport à l'identifiant.

Concernant les contraintes (saisi, lecture, enregistrement…), nous avons utilisé les exceptions en Java pour pallier aux différentes erreurs de compilation. Nous avons vu que l'utilisation d'un fichier CSV n'est pas adapter à une base de données. Cette partie nous a permis de voir le rôle du DAO avec les base de données.