



Rédacteurs :		
Kévin LEBRETON		Thomas RAMBALDI
REVISIONS		
Date	Nature de la modification	Version
24/10/16		1.1

Historique des modifications du document

Date	Version	Description	Auteur
23/10/2016	1.0	Création du corps du document	Rambaldi Thomas
24/10/2016	1.1	Rédaction des parties : Introduction / projet à tester / exigences	Rambaldi Thomas
28/10/2016	1.1	Rédaction des parties : Introduction / projet à tester / environnement de test / test à réaliser	Rambaldi Thomas Lebreton Kévin
07/11/2016	1.1	Rédaction des parties : Introduction / document joint	Rambaldi Thomas Lebreton Kévin

Table des matières

1.Introduction.....	5
1.1.Présentation de l'application.....	5
1.2.Contexte de réalisation.....	5
1.3.Choix technologiques.....	5
2.Projet à tester.....	6
2.1.Liste des modules et/ou des lots.....	6
2.2.Date de livraison.....	6
3.Documents joints.....	7
3.1.Documents généraux.....	7
3.2.Documents techniques.....	7
3.3.Documentation.....	7
4.Environnement de test.....	8
4.1.Outils de test.....	8
4.2.Bases de test / données de test.....	8
4.3.Coordonnées de l'administrateurs de la bases de données.....	8
5.Tests à réaliser.....	9
5.1.Listes des modules à tester.....	9
5.2.Objectif des tests.....	9
6.Exigences.....	10
6.1.Exigences fonctionnelles.....	10
6.1.1 Personnes.....	10
6.1.2 Groupes.....	10
6.2.Exigences non fonctionnelles.....	10
6.2.1 Développement du projet.....	10
6.3.Contraintes.....	10
6.3.1 Langage et APIs.....	10
6.4.Analyse du risque.....	11
6.5.Matrice Exigences/Risques pour définir les priorités.....	11
6.6.Jeux d'essais.....	11
6.7.Estimation de la charge.....	11
7.Stratégie de tests.....	12
7.1.Description de l'approche générale.....	12
7.2.Phase de tests.....	12
7.3.Champagne de test.....	12
7.4.Order d'exécution des tests.....	12
8.Condition d'arrêt.....	12
8.1.Critères retenus et pourquoi.....	12
9.Gestion des fiches d'anomalies.....	13
9.1.Modèles des fiches d'anomalie.....	13
9.2.Actions et états.....	14
9.3.Gestion des flux.....	14
9.4.Liste des intervenants.....	14

10.Ressources humaines.....	15
10.1.Nom et responsabilité.....	15
10.2.Informations utiles.....	15
11.Planning des test.....	15
11.1.Planning.....	15
11.2.Acteurs.....	15

1. Introduction

1.1. Présentation de l'application

L'application est un gestionnaire d'annuaire. Cette application permet à un utilisateur d'effectuer plusieurs actions.

Il peut lister des personnes appartenant à un groupe, visualiser les détails d'une personne, pouvoir ajouter ou supprimer des personnes ou modifier des informations.

Elle permet aussi de gérer les groupes (ajout ou suppression).

1.2. Contexte de réalisation

Le projet sera développé avec l'aide de l'IDE Eclipse et du langage Java (JEE). Le système de gestion de base de données qui sera utilisé dans ce développement sera MySQL. Plusieurs documents devront être au préalable rédigés comme par exemple le cahier des charges ou encore le plan de test.

La réalisation du projet se fera sur deux mois avec deux rendus. Une première partie, le 21 novembre 2016 et une deuxième le 12 décembre 2016.

1.3. Choix technologiques

Les choix des technologies sont les suivantes :

- Le Java concernant la programmation
- Le JEE pour du multi-niveaux
- Spring pour l'infrastructure de l'application
- JUnit pour les tests
- Emma pour la couverture de code
- L'IDE Eclipse pour le développement
- MySQL pour la partie gestion de la base de données
- Le SQL pour les requêtes

Prénom	Nom	Formation	Mail
Rambaldi	Thomas	M2 – FSIL – ISL	t.rambaldi13@gmail.com
Lebreton	Kévin	M2 – FSIL – ISL	K.kev542@laposte.net

2. Projet à tester

2.1. Liste des modules et/ou des lots

Concernant la couche donnée nous disposons des modules suivants :

- PersonDao
- GroupDao
- JDBC
- JavaBeans (Person, Group)

Pour ce qui est de la couche application nous avons les modules suivants :

- DAO

Pour la couche présentation les modules seront :

- Les pages JSP
- CSS

2.2. Date de livraison

Le projet sera composé de deux livrables :

- Le lundi 21 novembre 2016 :
 - La partie DAO
 - Lot 1 :
 - ♦ Une personne est représentée par un ensemble d'informations : identifiant, nom, prénom, adresse électronique, site WEB, date de naissance et mot de passe.
 - ♦ Chaque personne est placée dans un groupe. Un groupe est composé de quelques dizaines de personnes (par exemple les étudiants du M2 FSIL 2015/2016). Un groupe a donc un nom et un identifiant.
 - ♦ L'application doit présenter une liste de groupes, une liste de personnes de chaque groupe et une vue détaillée de chaque personne (sauf adresse électronique et date de naissance). Une fonction de recherche doit être offerte.
 - ♦ L'application doit être fonctionnelle si nous avons plusieurs milliers de personnes et plusieurs centaines de groupes.
 - ♦ L'application doit permettre à chaque personne de modifier sa description.
 - ♦ Les personnes présentes dans l'annuaire peuvent avoir accès à toutes les informations.
 - Plan de test
- Lundi 12 décembre 2016 :

→ Toutes les parties (pour plus de précision sur le contenu voir : <http://jean-luc.massat.perso.luminy.univ-amu.fr/ens/jee/devoir.html>)

3. Documents joints

3.1. Documents généraux

Le cahier des charges joint est le document qui représente les besoins du clients.

3.2. Documents techniques

Vous trouverez joint au projet deux autres plan de tests :

- Plan de test unitaires : représente les tests des petites parties de notre logiciel
- Plan de test d'intégrations : représente les tests des modules de notre application

3.3. Documentation

Un document qui présente comment utiliser le logiciel et un document qui inclus tous la javadoc nécessaire à la compréhension du code.

4. Environnement de test

4.1. Outils de test

Pour mener à bien le projet et afin que ce dernier soit le plus fiable et répondre aux besoins demandé, plusieurs outils seront utilisés :

- Junit 4 : pour les tests unitaires
- Emma : pour la couverture de code
- FindBugs pour l'analyse statique du code

4.2. Bases de test / données de test

Les données minimales qui devront être dans la table personne :

IdPers	IdGroup	NomPers	prenomPers	MailPers	WebPers	Naissance Pers	MdpPers
Int	Int	Vachar	Vachar	vachar	vachar	vachar	vachar

Les données minimales qui devront être dans la table groupe

IdGroup	NomGroup
Int	vachar

4.3. Coordonnées de l'administrateurs de la bases de données

Prénom	Nom	Formation	Mail
Rambaldi	Thomas	M2 – FSIL – ISL	t.rambaldi13@gmail.com

5. Tests à réaliser

5.1. Listes des modules à tester

Les modules à tester seront :

- Person
- Group
- JDBC
- DAO

5.2. Objectif des tests

Les tests ont pour but d'augmenter la fiabilité de l'application et de vérifier un nombre maximum de comportement. Ce qui permet de valider que le système agit comme demandé par ses développeur et répond ainsi aux besoins du client.

6. Exigences

6.1. Exigences fonctionnelles

6.1.1 Personnes

Une personne appartient à un groupe et un seul. L'application doit être en mesure de délivrer les services suivant : Lister / ajouter / supprimer / rechercher / visualiser les détail d'une personne.

6.1.2 Groupes

Un groupe comporte plusieurs personnes. L'application doit être en mesure de délivrer les services suivant : Lister / ajouter / supprimer un groupe

6.2. Exigences non fonctionnelles

6.2.1 Développement Du Projet

Le développement de ce projet devra se faire sur Windows ou Linux, à l'aide d'eclipse et de la technologie JEE avec Spring. Le système doit être fiable pour cela des outils seront nécessaires (JUnit, Emma, FindBug ou autre). L'application doit être en mesure de gérer plusieurs centaines de personnes.

6.3. Contraintes

6.3.1 Langage Et APIs

Le choix du langage de programmation est imposée, ainsi que les APIs.

- Java / JEE
- Spring
- JUnit
- Les composants DAO doivent être connectés et paramétrés par une couche Spring
- Préparation de la base de données pour insérer des données cohérentes.

6.4. Analyse du risque

Dans cette application les risques ne sont pas aussi nombreux qu'un projet de grande envergure, mais les risques sont quand même présent . Pour cette analyse les types de menaces seront extraits parmi les dix-sept menaces répertoriées par la méthode Marion

Les différents risques sont les suivants :

- Accidents physiques
- Erreur lorsqu'un utilisateur fait une saisie
- Erreur de transmission de données
- Erreur de conception / développement
- Erreur d'exploitation

6.5. Matrice Exigences/Risques pour définir les priorités

Nom tâche	Exigences	Risques	Priorité
			5

6.6. Jeux d'essais

6.7. Estimation de la charge

7. Stratégie de tests

On décrira dans ce paragraphe la démarche mise en œuvre pour réaliser les tests

7.1. Description de l'approche générale

7.2. Phase de tests

7.3. Champagne de test

7.4. Ordre d'exécution des tests

8. Condition d'arrêt

8.1. Critères retenus et pourquoi

9. Gestion des fiches d'anomalies

Les anomalies constatées doivent être formalisées pour être transmises à l'entité responsable.

Des outils spécialisés peuvent être utilisés pour déclarer et suivre les anomalies (bugtracker) du projet. Il est également possible de déclarer et suivre les anomalies avec des simples outils de bureautique...

Le contenu de la fiche d'anomalie peut varier en fonction de la phase projet (recette, production...)

Exemples d'informations que l'on peut indiquer sur fiche d'anomalie :

- Identification de l'anomalie
 - Identifiant de l'anomalie
 - Personne ayant déclarée l'anomalie. Permet notamment de prendre contact avec l'émetteur en cas d'incompréhension ou manque de précision
 - Gravité de l'anomalie
 - Date de création
 - Date à laquelle l'anomalie a été clôturée (Après validation du testeur)
 - Statut de l'anomalie (En cours, corrigée, livrée...)
 - Version de l'application sur laquelle l'anomalie a été constatée
 - Environnement
- Description de l'anomalie
 - Scénario de test sur lequel l'anomalie a été constatée
 - Cas de test sur une anomalie détectée pendant une phase de recette
 - Comportement attendu
 - Comportement constaté
- Réponse
 - Analyse
 - Date de livraison
 - Version

9.1. Modèles des fiches d'anomalie

Identification de l'anomalie

Identifiant		Emetteur	
Type		Gravité	
Date de création		Date de clôture	
Environnement		Version	

Description

Scénario		Cas de test	
Comportement			
Comportement constaté			

Réponse

Analyse			
Date de livraison		Version	

9.2. Actions et états

9.3. Gestion des flux

9.4. Liste des intervenants

Prénom	Nom	Formation	Mail
Thomas	Rambaldi	M2 – FSIL – ISL	t.rambaldi13@gmail.com
Kévin	Lebreton	M2 – FSIL – ISL	K.kev542@laposte.net

10. Ressources humaines

10.1. Nom et responsabilité

10.2. Informations utiles

11. Planning des test

11.1. Planning

11.2. Acteurs