

i18n Project
Document d'Architecture du Logiciel
Version <2.1>



i18n Project	Version: <2.1>
	Date: <13/06/2016>

Historique des révisions

Date	Version	Description	Auteur
<28/05/2016>	<1.0>	<détails>	NASSIRI

i18n Project	Version: <2.1>
	Date: <13/06/2016>

Table des matières

1. Introduction	4
2. Objectif du logiciel	4
2.1 Contexte	4
2.2 Besoins fonctionnels	4
2.3 Besoins non fonctionnels	4
3. Structure	5
3.1 Vue des couches	5
3.2 Sous-systèmes et paquetages	6
3.3 Interfaces	8
4. Comportement	Erreur ! Signet non défini.
4.1 Réalisation des cas d'utilisation	8
4.2 Mécanismes	8
5. Autres vues	Erreur ! Signet non défini.
5.1 Vue processus (optionnel)	Erreur ! Signet non défini.
5.2 Vue implémentation (optionnel)	Erreur ! Signet non défini.
5.3 Vue déploiement (optionnel)	Erreur ! Signet non défini.
5.4 Vue données (optionnel)	Erreur ! Signet non défini.
6. Concepts du domaine	9
7. Qualités de l'architecture	9
8. Points ouverts	Erreur ! Signet non défini.

i18n Project	Version: <2.1>
	Date: <13/06/2016>

Document d'Architecture du Logiciel

1. Introduction

Ce document est une description de l'architecture de l'application, il vise à donner une idée claire sur l'architecture globale de l'application (les couches, les relations entre elles).

2. Objectif du logiciel

Le logiciel visé dans ce document, nommé (i18n)Project, a pour objectif de proposer un logiciel de gestion de projets adapté au contexte multinational des organismes tels que les groupes industriels ou les ONG internationaux.

2.1 Contexte

Ce logiciel permet de planifier et suivre l'avancement des activités d'un projet, de leur affecter des ressources humaines et matérielles, en tenant compte des caractéristiques locales et des préférences culturelles des acteurs d'un projet.

2.2 Besoins fonctionnels

Les besoins fonctionnels de l'application sont :

- CU1 : Planifier projet.
- CU2 : Gérer référentiel.
- CU3 : Consulter reporting.

2.3 Besoins non fonctionnels

Des qualités et des contraintes importantes à prendre en compte lors de l'élaboration de l'architecture de l'application :

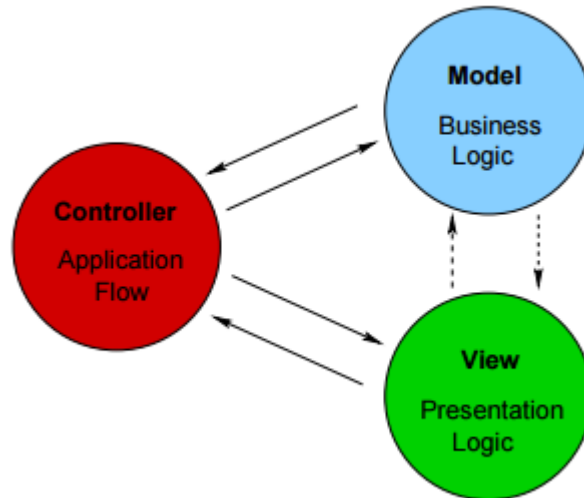
- Fonctionnalité
- Utilisabilité
- Fiabilité
- Disponibilité
- Temps moyen entre les pannes
- Temps moyen de réparation
- Performance
- Temps de réponse
- Capacité

i18n Project	Version: <2.1>
	Date: <13/06/2016>

3. Structure

3.1 Vue des couches

Le diagramme suivant représente les couches du logiciel. Nous expliquerons ensuite le rôle de chaque couche.



3.1.1 La couche MODEL :

- Représente le fond (sujet d'étude).
- S'intéresse à la représentation des données de la couche métier (business logique), i.e. données spécifiques à l'application.

3.1.2 La couche VIEW :

- Concerne la forme (représentation).
- Elle interagit avec le modèle.
- La vue concerne principalement la représentation des données du modèle à l'écran (ou sur tout autre périphérique de sortie).
- Il peut donc exister plusieurs vues.

3.1.3 La couche CONTROLLER:

Le contrôleur gère les interactions avec l'utilisateur :

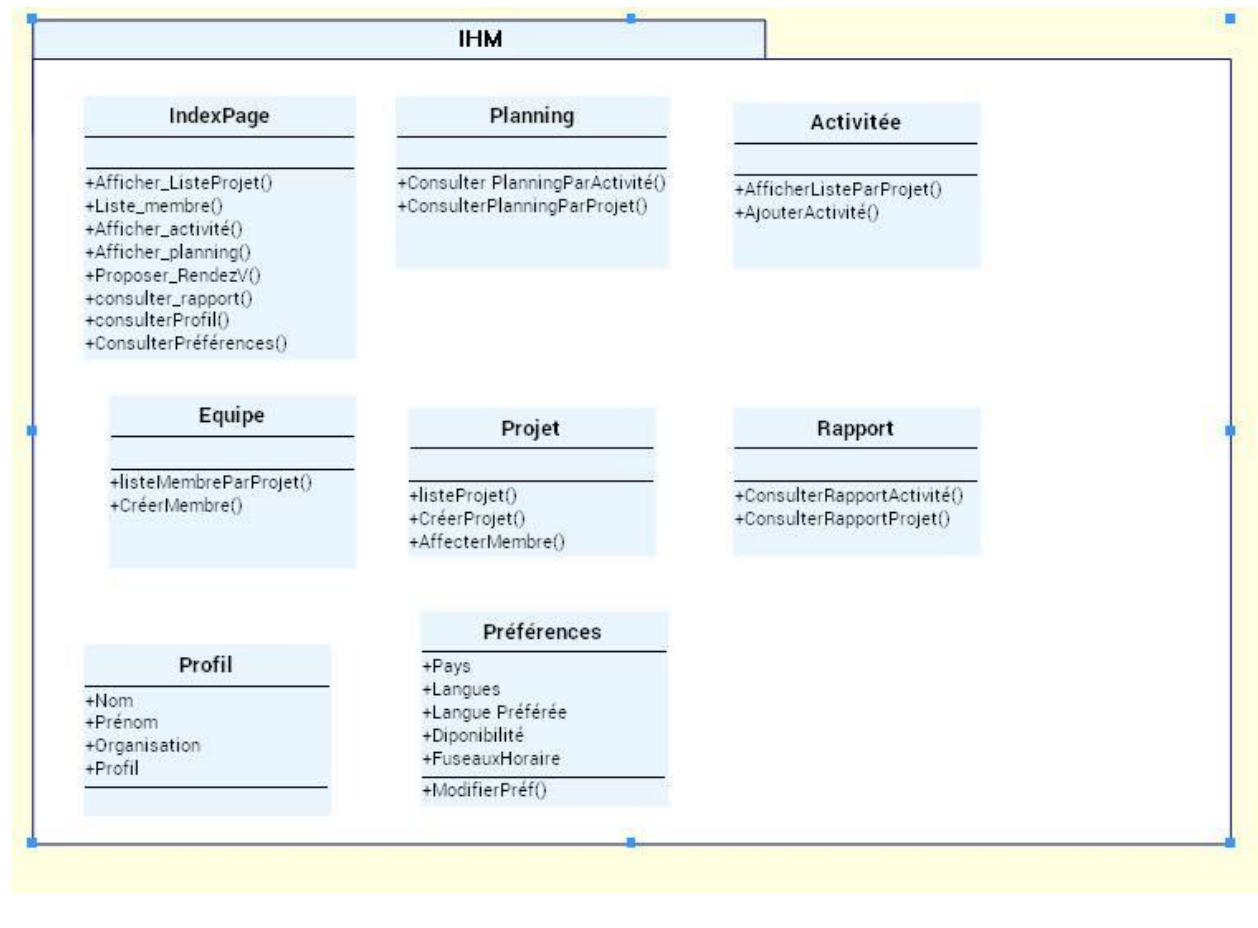
- S'occupe de la réécriture des URL.
- Détermine quels traitements doivent être réalisés.

i18n Project	Version: <2.1>
	Date: <13/06/2016>

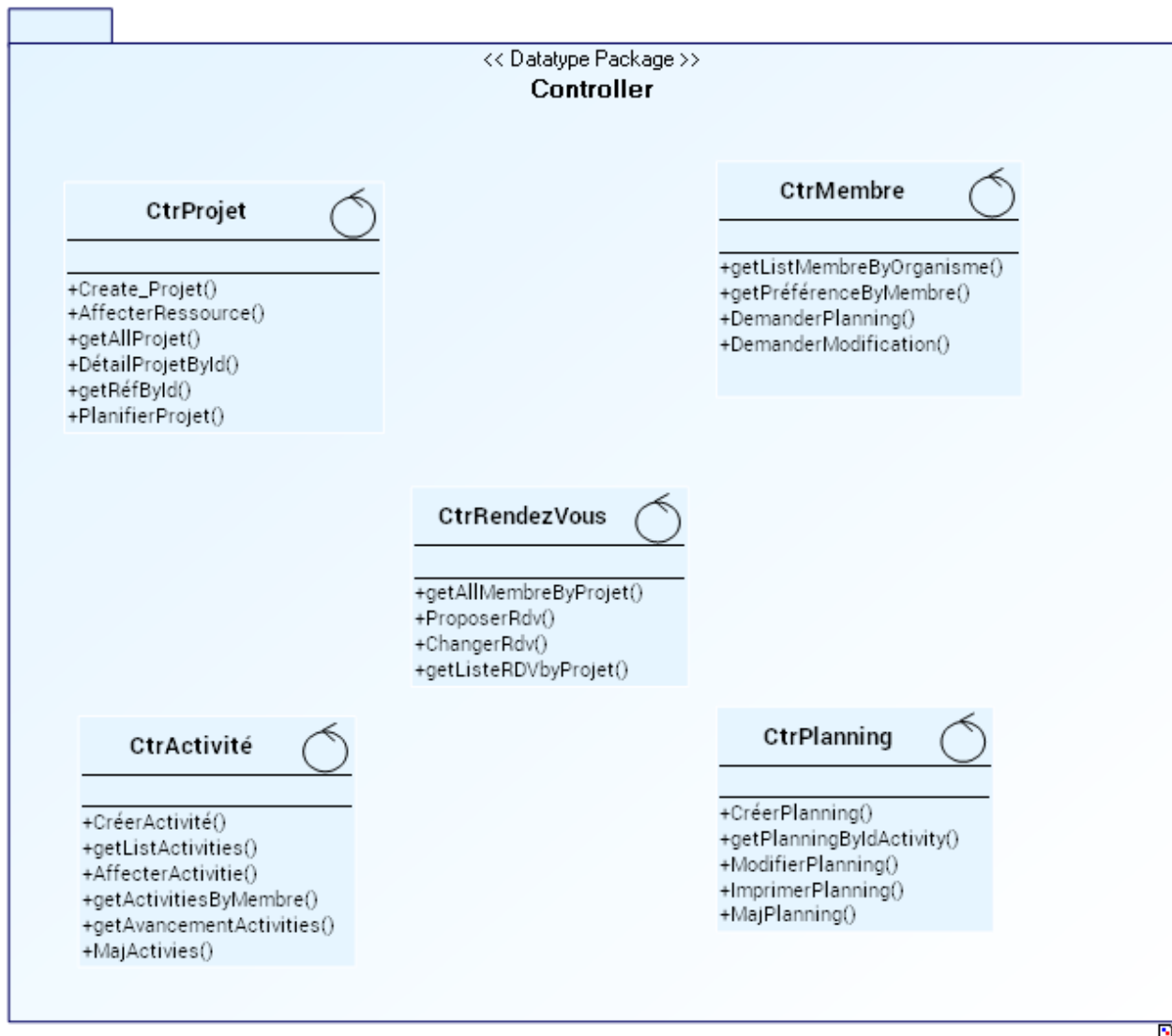
3.2 Sous-systèmes et paquetages

Le diagramme suivant établit la représentation en couches de l'architecture du logiciel i18n Project. Chaque couche de l'architecture se décompose en différents paquetages gérant chacun un aspect du rôle de la couche à laquelle il appartient, et chaque paquetage regroupe plusieurs classes significatives. Pour chaque paquetage, nous en ferons une brève description et nous citerons les classes importantes.

Couche INTERFACE :

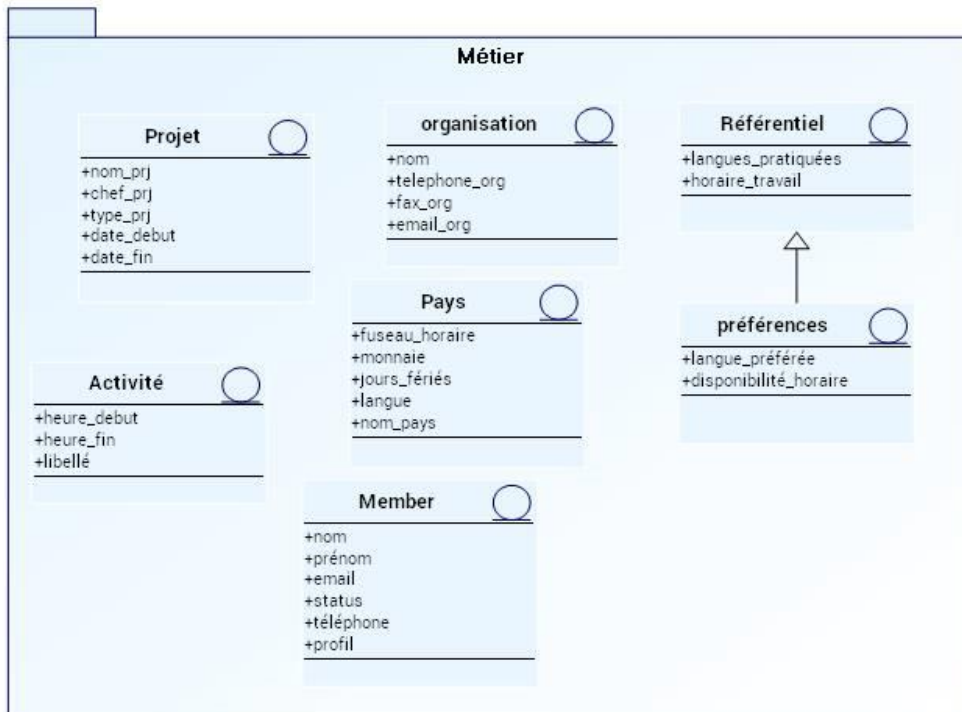


Couche CONTROLE :



Couche Métier

i18n Project	Version: <2.1>
	Date: <13/06/2016>



3.3 Interfaces

Le diagramme représentant les interfaces de notre application est représenté dans l'axe précédent (3.2 Sous-système et Paquetages).

3.4 Réalisation des cas d'utilisation

[pour chaque cas d'utilisation identifié comme important pour l'architecture, présenter sa réalisation. On prendra un ou plusieurs scénarios du cas d'utilisation et on le décrira par un diagramme d'interaction. Les objets apparaissant dans les diagrammes doivent être clairement identifiés dans la partie structure. Ajouter si nécessaire des explications pour qu'on comprenne bien comment les éléments structurels contribuent au scénario.]

3.5 Mécanismes

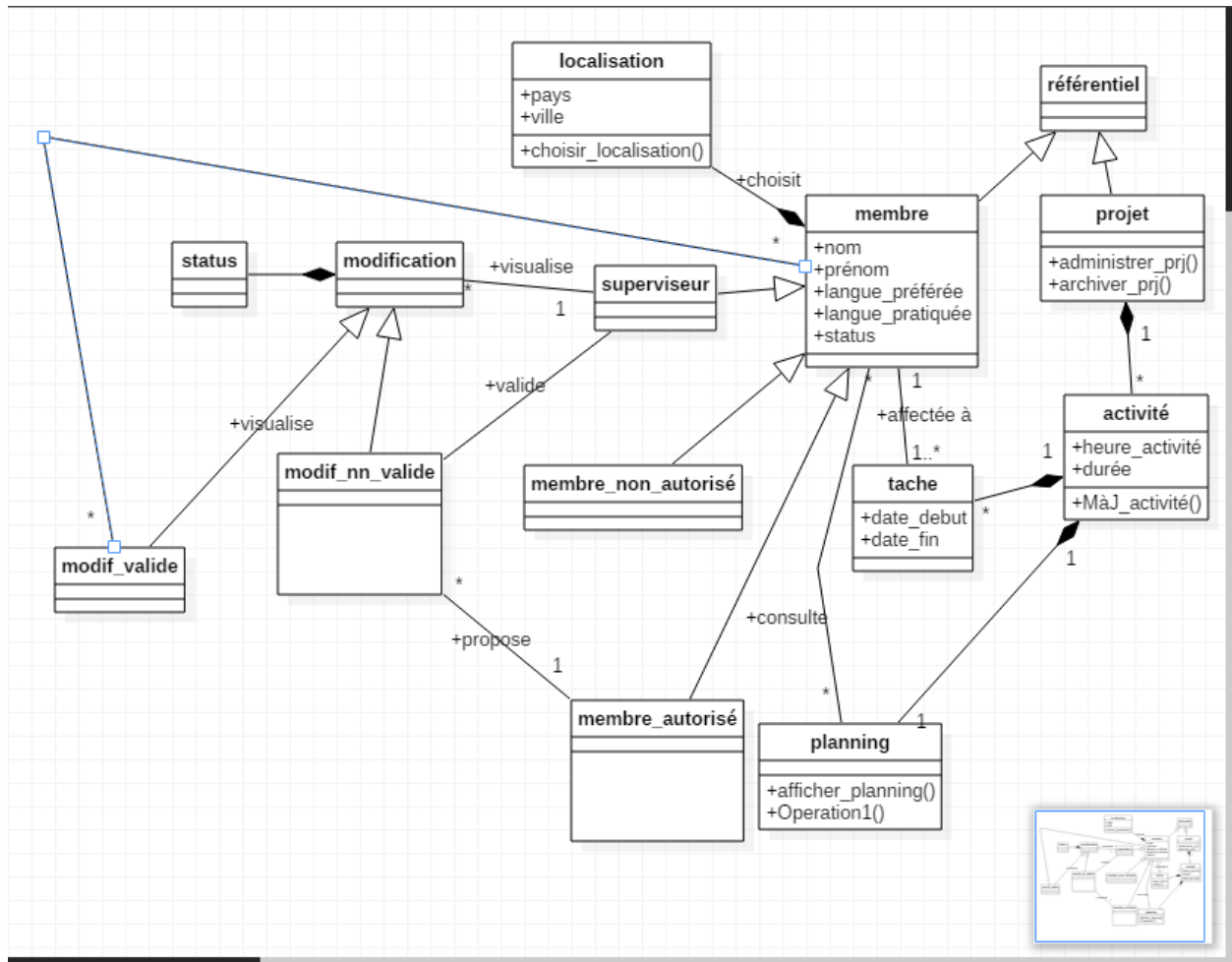
Technologie

i18n Project	Version: <2.1>
	Date: <13/06/2016>

Plusieurs possibilités de développement, ci-dessous : le listing des avantages et inconvénients de chacun :
Développement Avantages Inconvénients
Java Connaissance de la technologie Nécessite de développer TOUS les scénarios de l'application
Bases de données MySQL Connaissance de la technologie

4. Concepts du domaine

Cette section décrit les concepts spécifiques du domaine et leurs relations.



5. Qualités de l'architecture

L'utilisation d'une architecture en couche de type Interface/Contrôle/Domaine permet de décomposer de façon claire la réalisation d'un cas d'utilisation. Ce système permet de déléguer à chaque couche une responsabilité lors de l'utilisation d'une fonctionnalité du logiciel. Chacune d'elles communiquera avec une ou plusieurs couches sous-jacentes.