

# Rapport de conception

Adonis NAJIMI,  
Valentin STERN,  
Vincent ALBERT,  
Théo GERRIET

29 décembre 2013

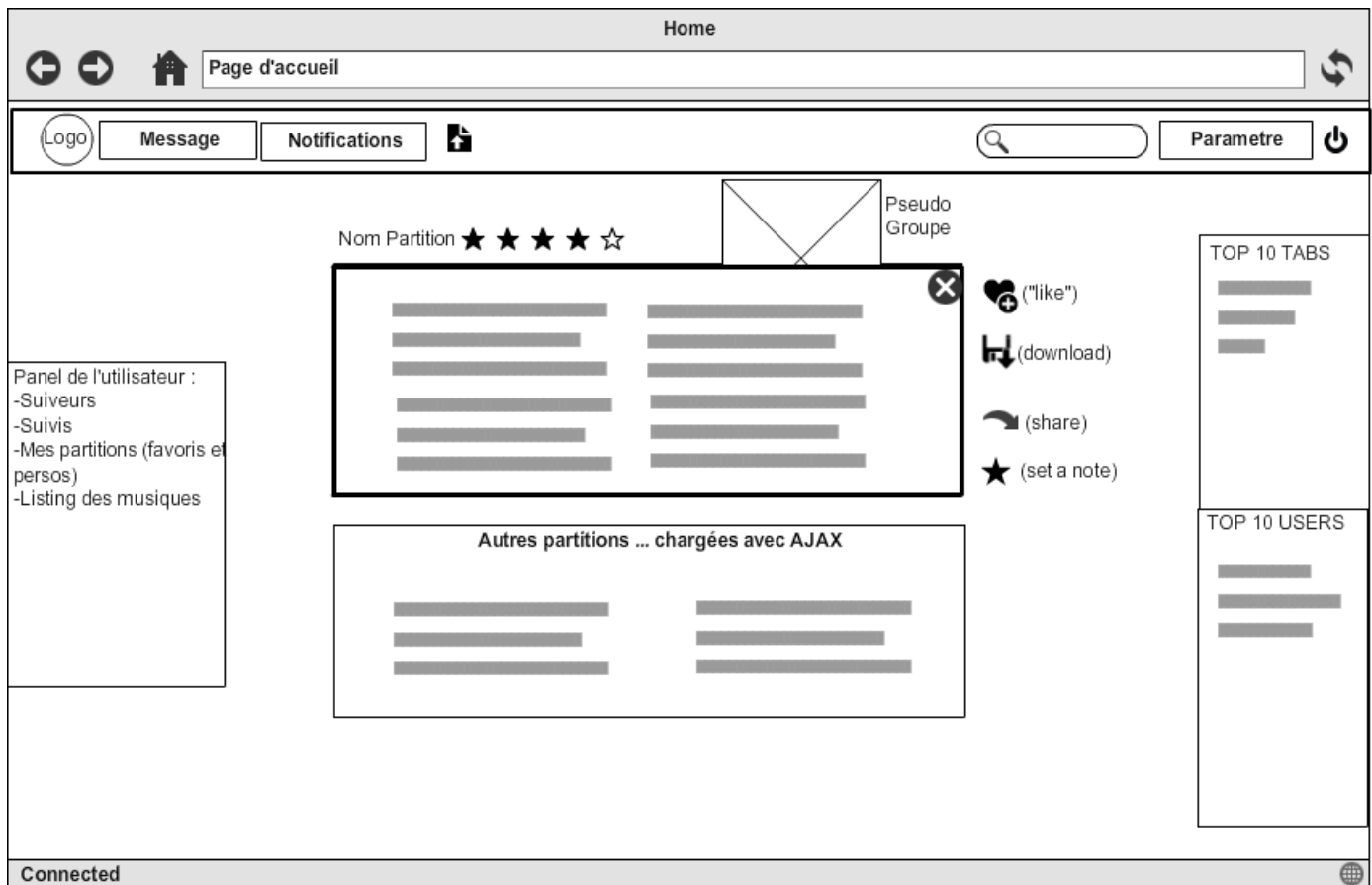
# 1 Présentation du projet

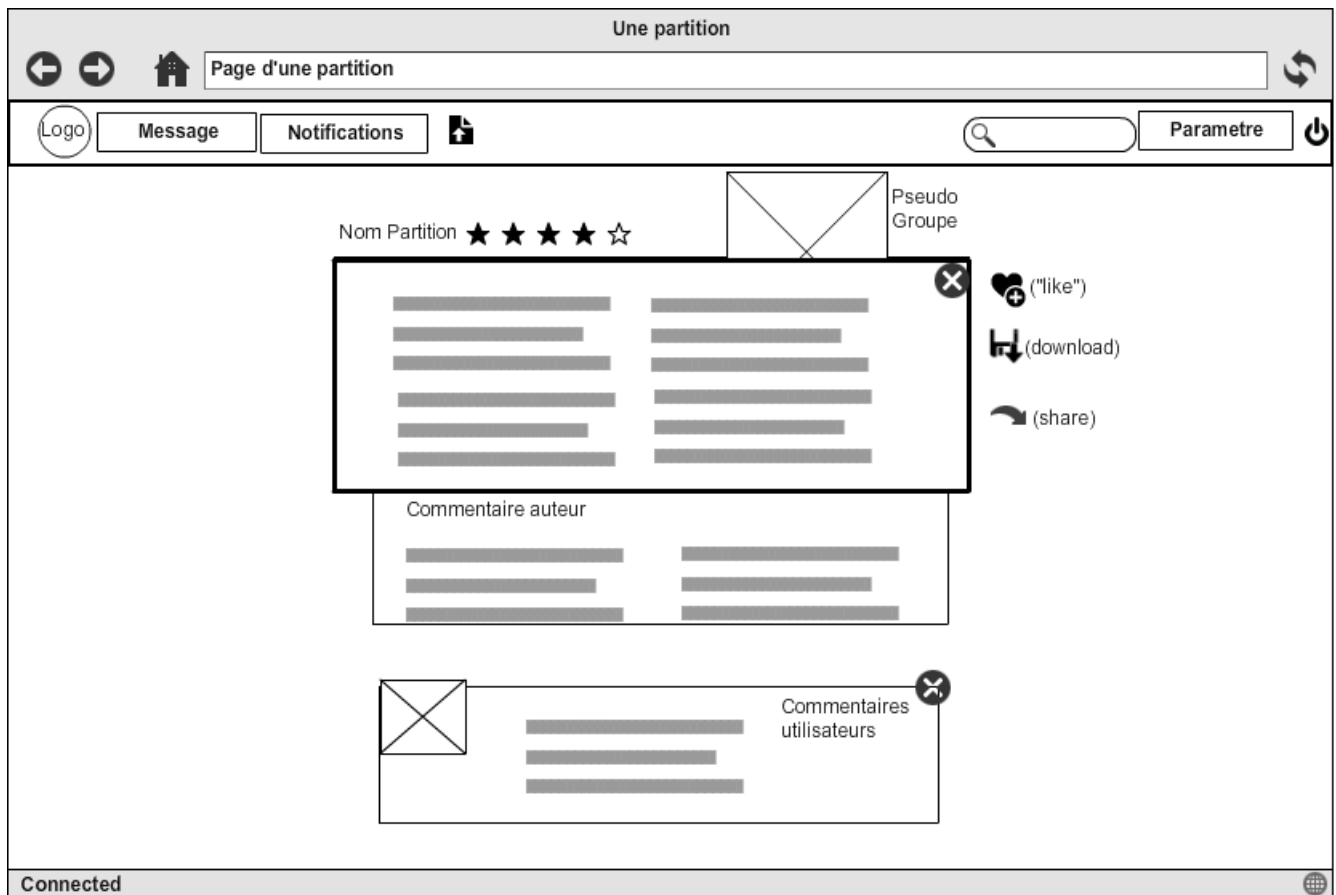
## 2 Client C++

### 2.1 UML

## 3 Application Web

### 3.1 Maquettes d'écran





Utilisateur

Page profil utilisateur

Image de profil

Infos de base

Nom, Prénom, Pseudo, Groupes,

nb suiveurs suivis et favoris

Editer

Édition possible quand on est l'utilisateur en question

Info

Infos détaillées,

description, age, sexe

Tabs

Déroulement Javascript lors du clic sur le bouton

Connected

Groupe

Groupe

logo

Infos groupe

ÉDITER

Si membre du groupe autorisé

Description

Un clic sur le bouton contenant le titre de la section génère un effet de

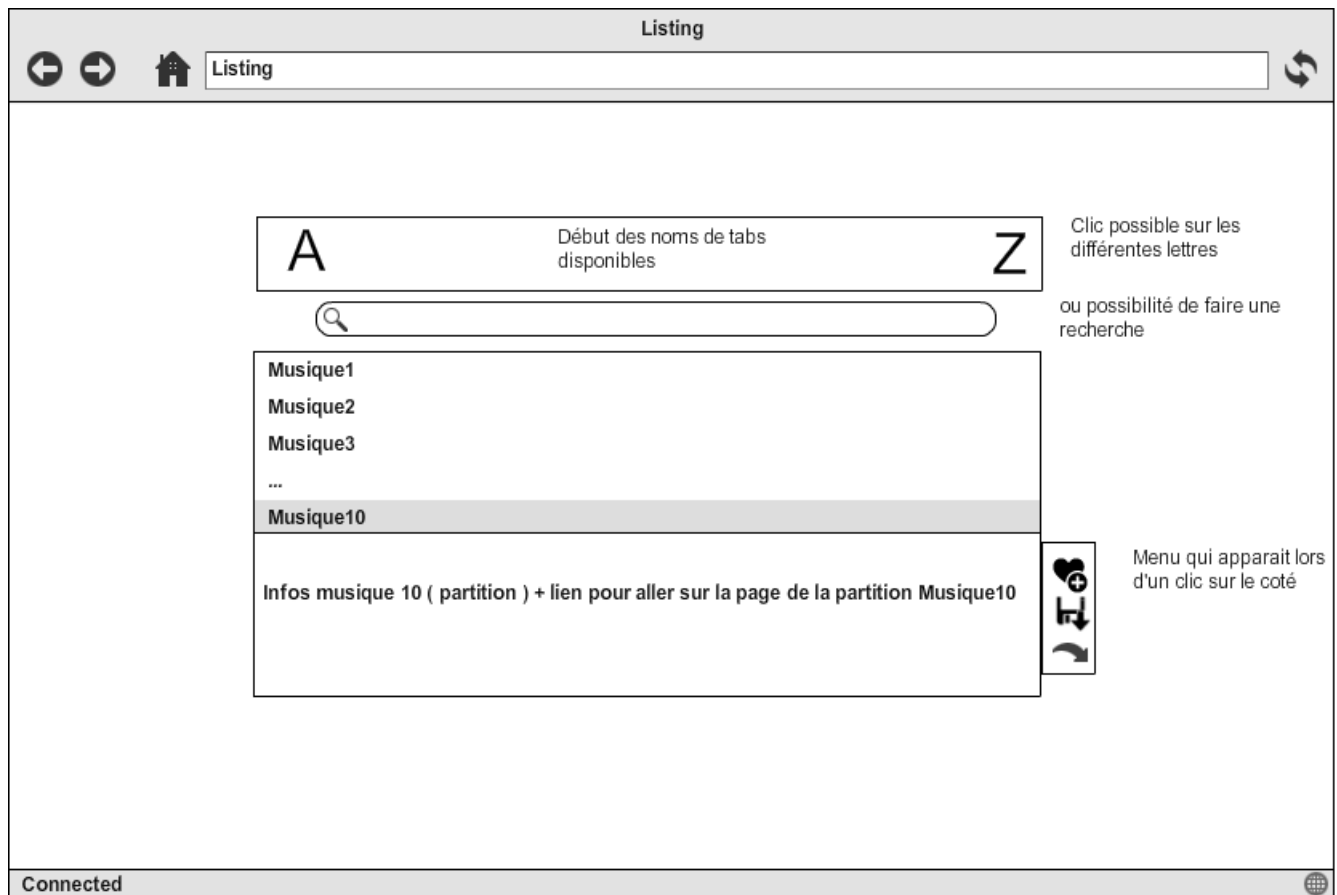
Liste des membre

pseudo + image miniature

Créations

Liste des TABS ( comme dans le listing )

Connected



### 3.2 Modèle Merise MCD

### 3.3 Modèle Merise MLD

Replies (#user\_id, #comment\_id, reply\_date)

Comments (comment\_id, comment\_text, #tab\_id, #user\_id, write\_date)

Follows (#user\_id, #user\_id, follow\_date)

Users (user\_id, user\_name, user\_birthday, user\_nickname, user\_description, user\_email, user\_signup)

Likes (#tab\_id, #user\_id, like\_date)

Belongs (#user\_id, #group\_id, belong\_date)

Creates (#group\_id, #user\_id, create\_date)

Tabs (tab\_id, tab\_name, tab\_file, #user\_id, tab\_date)

Groups (group\_id, group\_name, group\_creation)

Makes (#group\_id, #tab\_id, make\_date)

### 3.4 Structure de l'application

—Structures— views : -home // invité ou non -compte // config ou non - groupe // config ou non -listing -barre en haut? (gérer upload) -page partition -partition elle même

controllants (gèrent les animations et DATA ? ) ANGULAR : home.js || control-  
leur général compte.js || contrôleur du compte => gère les effets groupe.js ||  
contrôleur gère effet listing.js || gèrent effet bare.js partition.js partition-page.js  
home.js => gestion des animations + gestion des datas à afficher  
DATA => voir base de données Active record => une table = un module  
-methode de base (findAll,byId,insert,update,delete)+ sorted by ...  
les routes express gèrent les chemins de navigations  
Routes : chaque lien est une route qui peut contenir une action et qui actionne  
une méthode du controller lié à la route