

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

Les différentes maquettes du site permettent d'améliorer la vision des besoins de l'application. Ces visuels ne sont pas définitifs et sont simplement là pour nous aider à mieux structurer l'application et organiser les objectifs nécessaires à la réalisation de celle-ci.

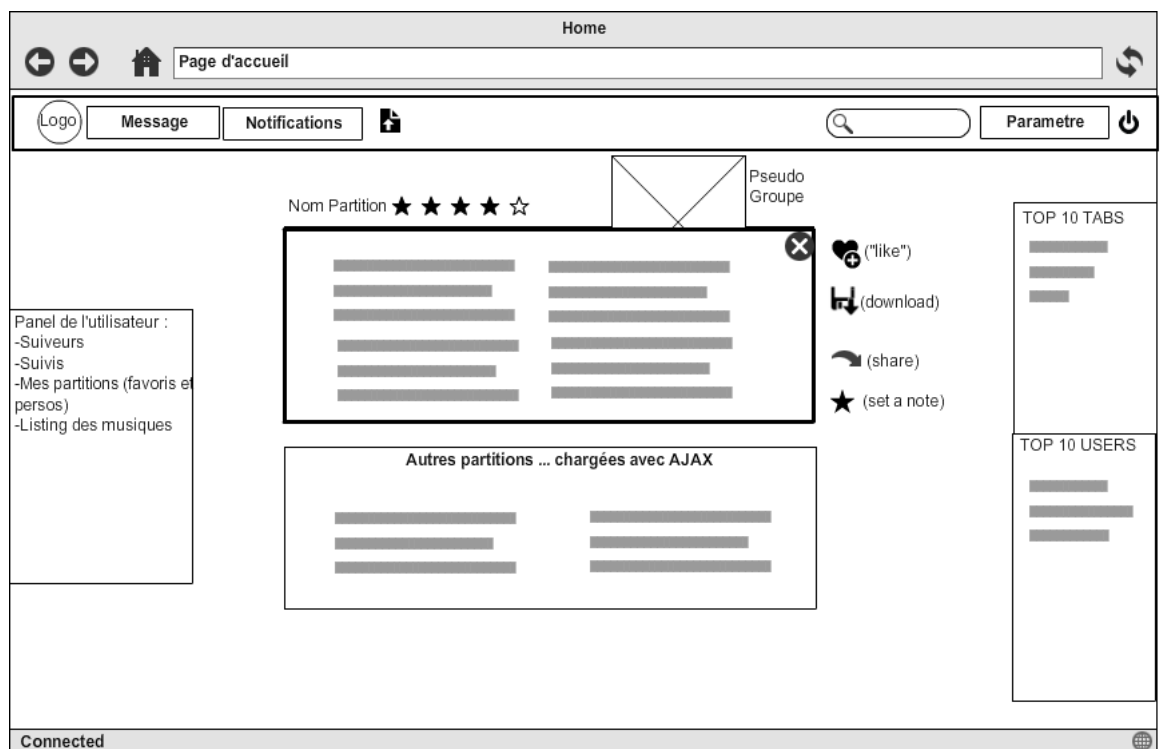


FIGURE 1 – Maquette de la page principale du site

La barre du haut est une barre de menu qui sera disponible sur toute les pages du site, elle n'est pas représentée sur toutes les maquettes par soucis de clarté. En effet, le principal sujet des maquettes suivantes n'est pas la présence de la barre de menu mais le contenu de celles-ci

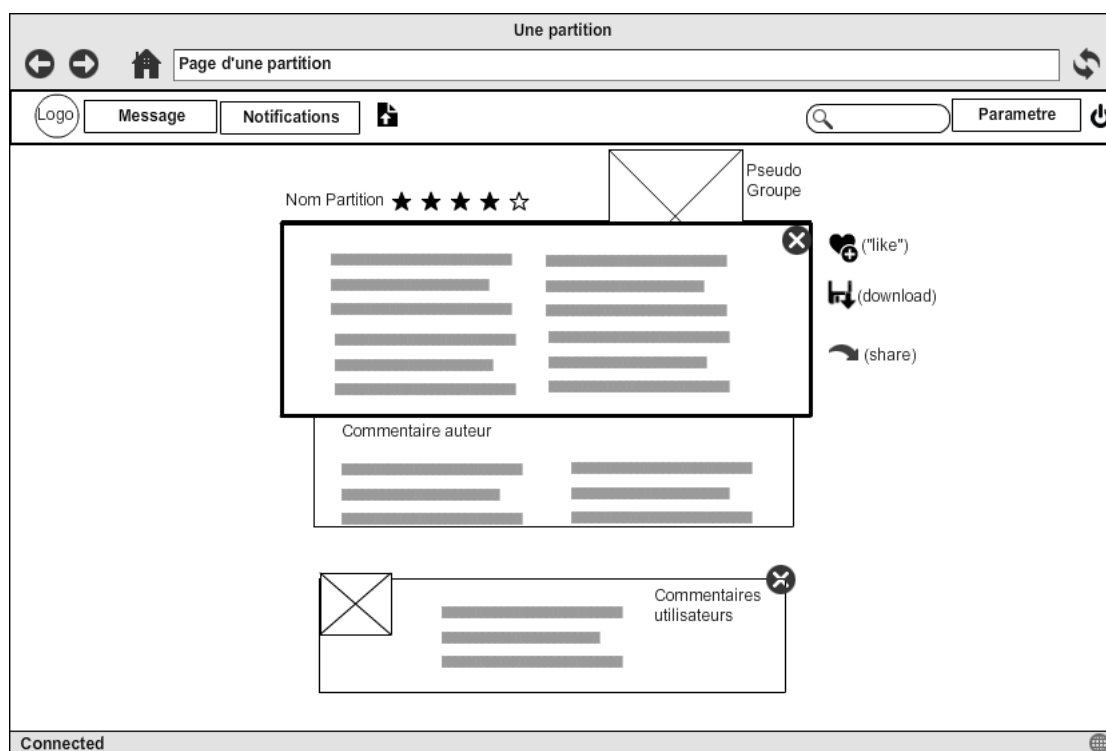


FIGURE 2 – Page principale d’une partition de musique

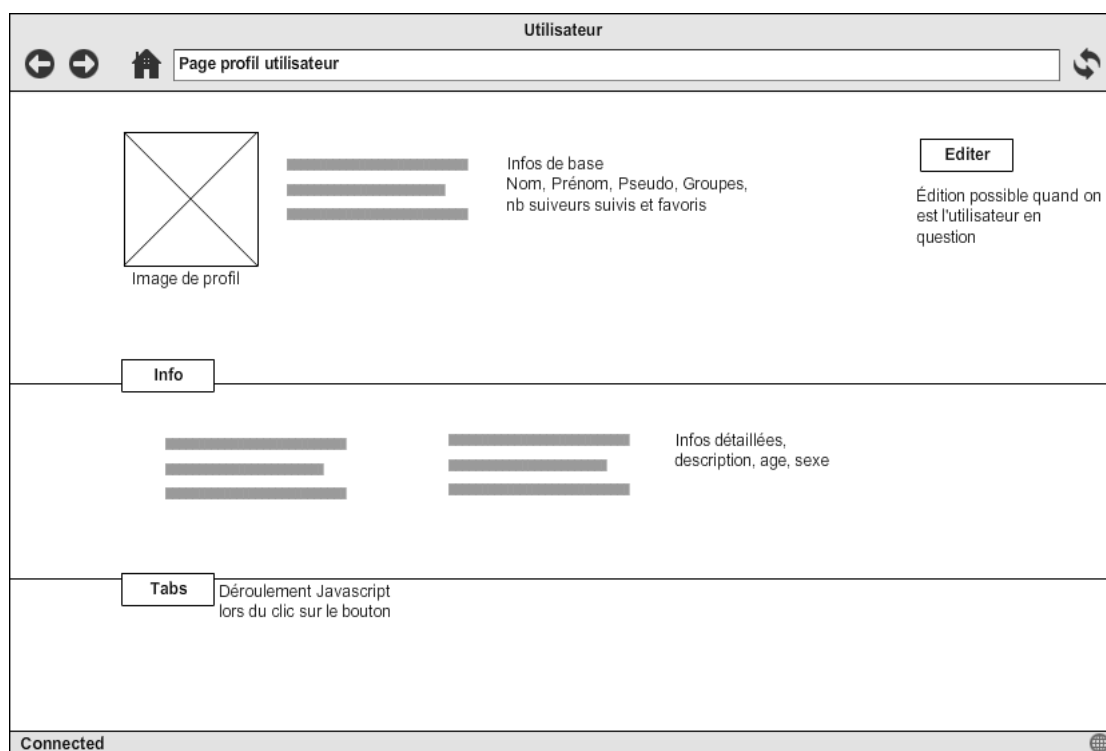


FIGURE 3 – Page de profil utilisateur

Cette page permettra à l'utilisateur concerné de modifier ses informations s'il le souhaite.

Les autres utilisateurs verront sur sa page de profil les informations rendues visibles par l'utilisateur en question.

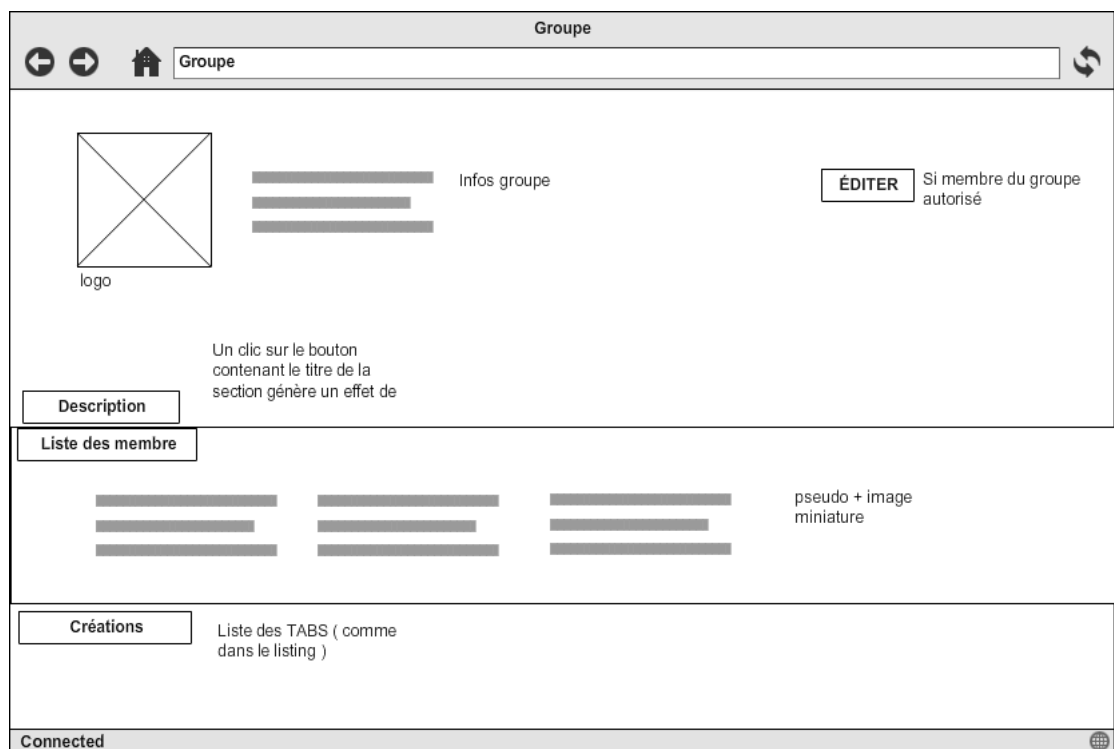
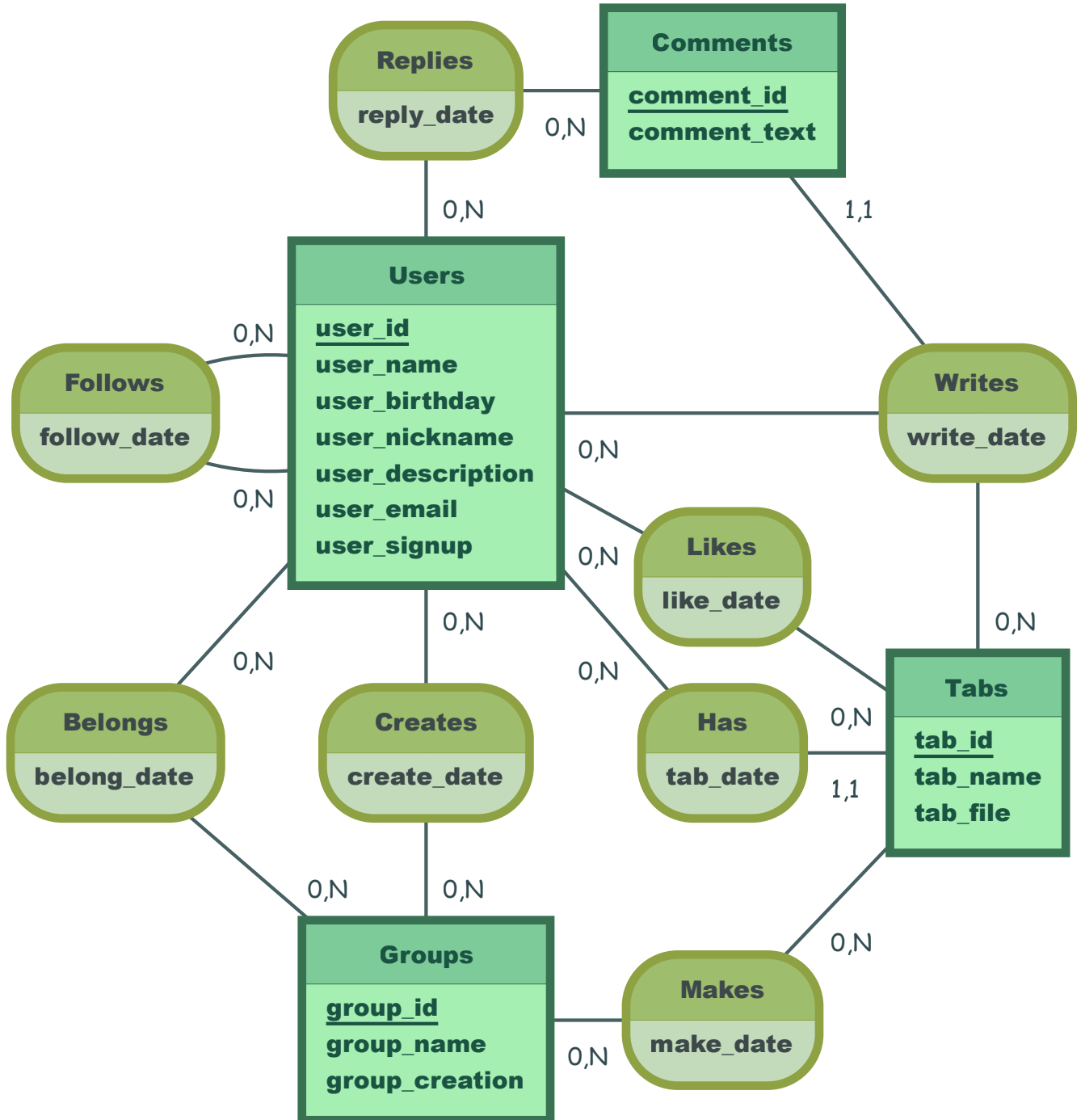


FIGURE 4 – Page d'un groupe de musique
 Cette page permettra de modifier le groupe si l'utilisateur en possède les droits.
 Cette page contiendra la liste des membres du groupe ainsi que les différentes
 partitions liées au groupe concerné.



FIGURE 5 – Liste de toutes les partitions
Le listing des partitions permettra une recherche plus approfondie qu’avec la recherche disponible dans la barre de menu (la barre du haut).

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

controllers (gèrent les animations et DATA ?) ANGULAR : home.js || controleur général compte.js || controleur du compte => gère les effets groupe.js || controleur 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