

Dossier de conception technique

CHALLENGES PLANET



Développeurs
Bastien LEDON & Saskia GIRAUD

16/01/2015
Version 4

Table des matières

Les diagrammes de cas d'utilisation	4
Etudiant.....	5
Public.....	5
Modélisation des données	7
Tables:	7
STRUCTURE GENERALE	9
CONFIGURATION générale.	9
Structure GitHub - Git	9
FRONT – (web site).....	9
Home	9
Page 404	10
Création events	11
Création projet/équipe/groupe	12
Inscription.....	13
Connexion	14
DASHBOARD - (admin).....	15
Connexion	15
Reconnexion.....	16
Notifications	16
Projets en cours.....	16
Messages.....	16
Compte	16
Ajouter un utilisateur.....	16
Ajouter un projet.....	16
Ajouter un évènement	16
Voir un/des utilisateur(s)	16
Voir un/des projet(s)	16
Voir un/des évènement(s).....	16
Equipe.....	17
Cookie	17
Mesure d'audience	17
SENARIOS TESTS.....	18
DASHBOARD – (admin).....	18
Connexion	18
Home	18
Page 404	18
Ajout utilisateurs	18
Afficher projets	18
Afficher évènements.....	18
Afficher utilisateurs	18
Afficher administrateurs	19
Déconnexion	19
Super admin	19

Considérations générales

Objectif du document

Ce document a pour objectif de donner les fonctionnements et réflexions élaborées en amont de la réalisation du projet.

Les technologies utilisées

- HTML 5
- CSS / CSS 3
- JS / jQuery / AJAX
- PHP / PDO / POO
- MySQL

Les outils utilisés

- Sublime Text 3 / Coda
- FileZilla
- Github
- Git
- MAMP

Le vocabulaire utilisé

- Base de donnée (BDD)
- Mot de passe (mdp)
- Back office (BO)
- Model View Controller (MVC)
- Dans les schéma: C = Controller & M = Model

Norme de nommage

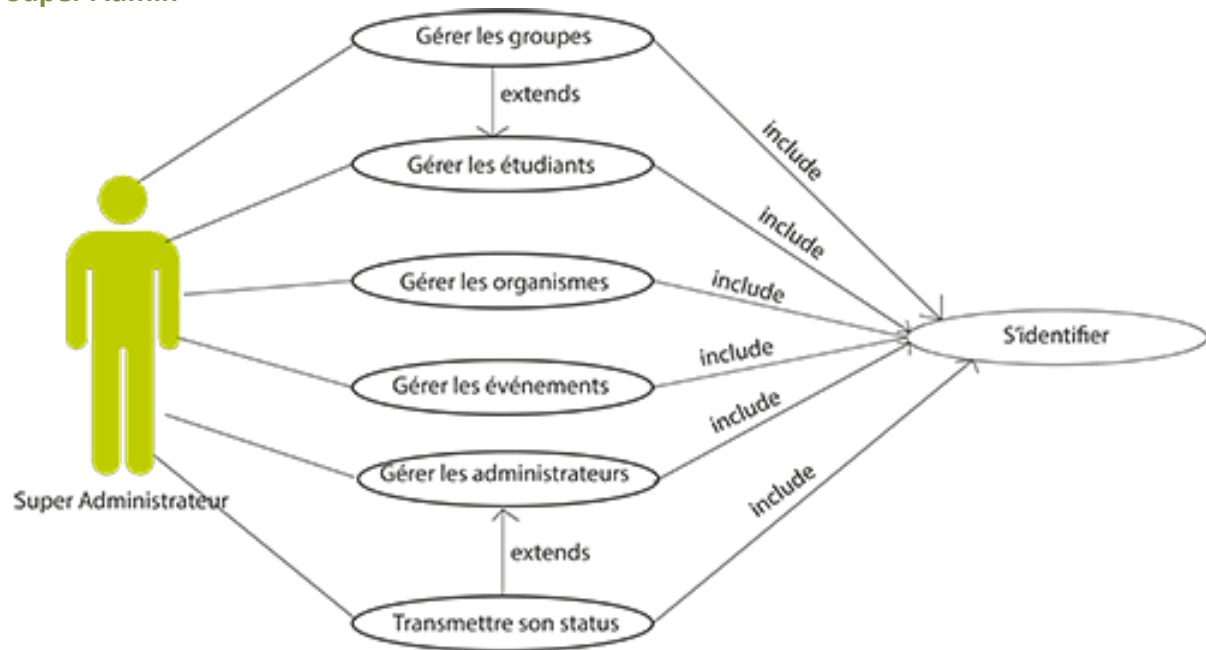
- camelCase

Les acteurs du site

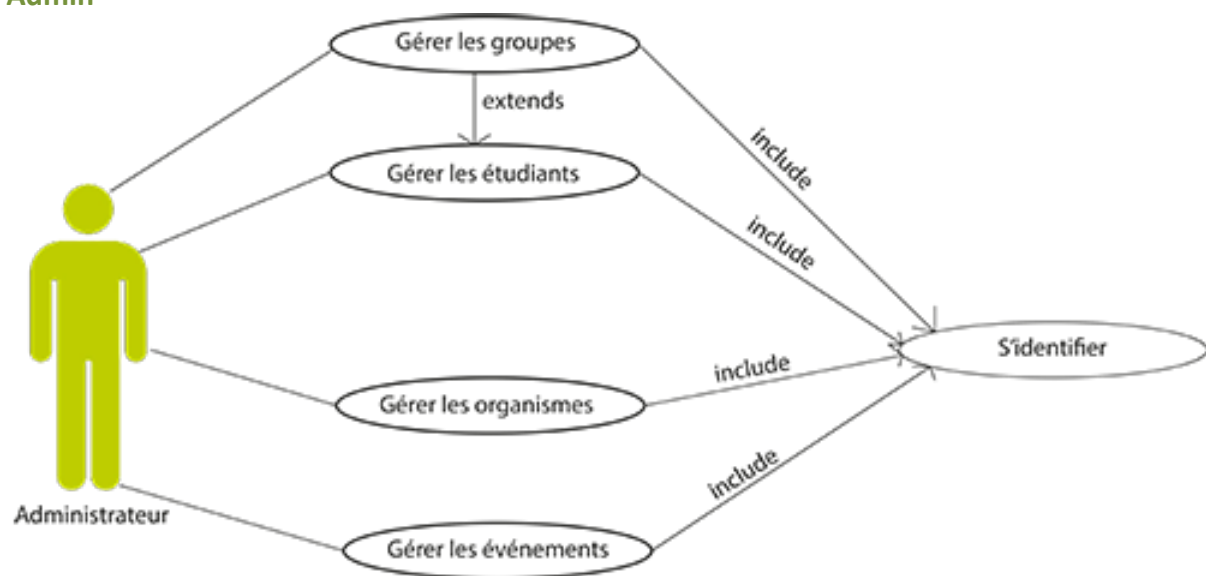
- Le super admin, assure les fonctions de gestion des utilisateurs et des événements ainsi que les administrateurs et membres.
- L'administrateur, assure les fonctions de gestion des utilisateurs et des événements.
- L'étudiant identifié, crée et rejoint des équipes liés à des événements.
- L'organisme identifié, sponsorise les équipes.
- L'utilisateur public, crée des événements.

Les diagrammes de cas d'utilisation

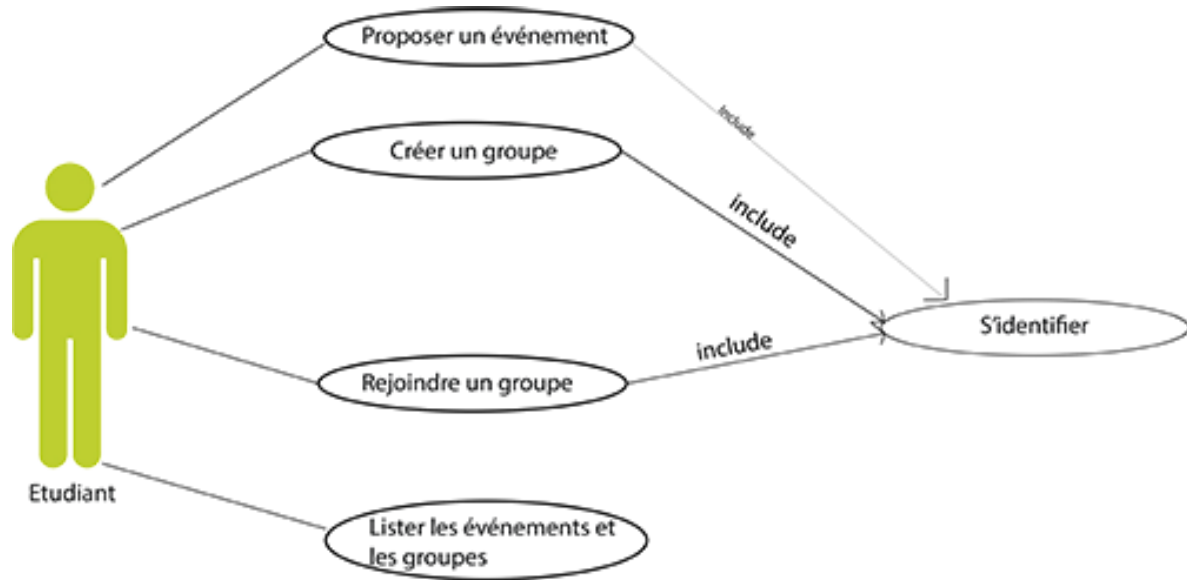
Super Admin



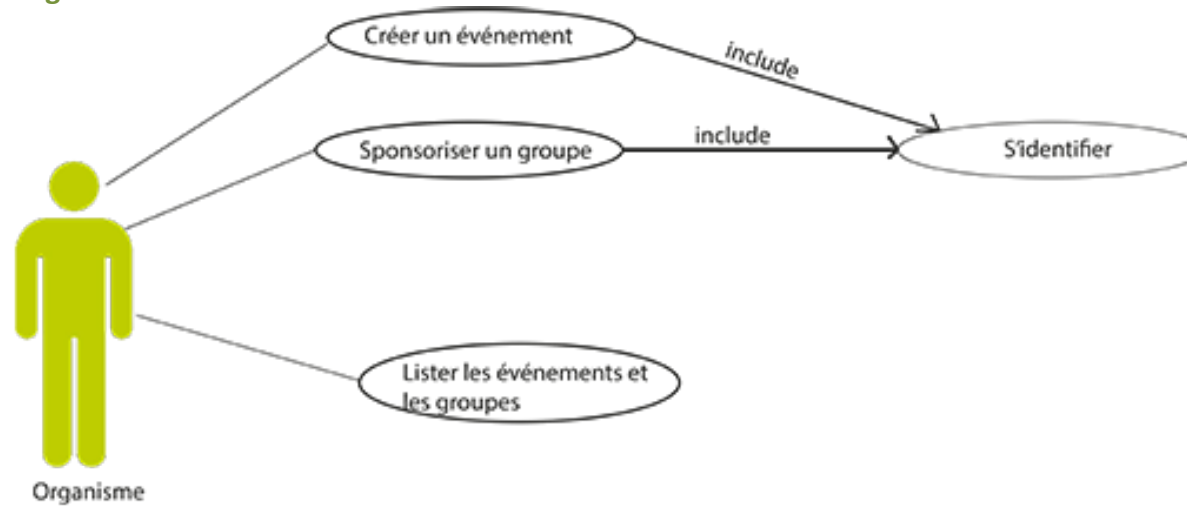
Admin



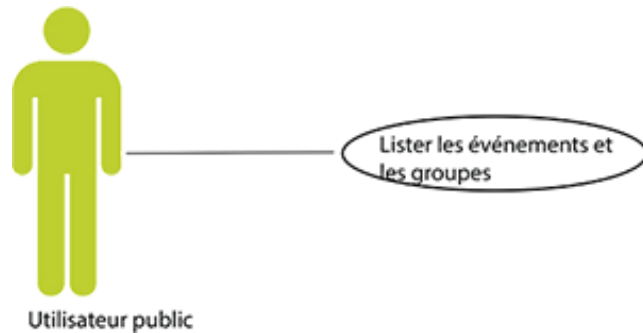
Etudiant



Organisme



Public



Environnement de développement

- Serveur de dev: Local & [giraudsa](#)
- Serveur de test: [cambour](#)
- Serveur de prod: [daguanno](#)
-

Architecture technique

App

- conf
 - **Configurations**
 - *conf_define*
 - *conf_mysql*
 - *conf_url*
 - *message*
 - *mysql*
 - ...
- controller
 - **Controller spécifiques**
 - *404Controller*
 - *PageController*
 - *UserController*
 - ...
- model
 - **Model spécifiques**
 - *UserModel*
 - ...
- view
 - **Vues**
 - *dossiers de fichiers*
 - *fichiers html en .php*
 - ...
 - *page*
 - *fichiers html en .php*
 - ...
 - *layout*
 - *fichier de header*
 - *fichier de footer*
 - ...
- app.php

Core

- **Controller globaux**
- *CoreController*
- *CoreModel*

- Load

Lib

- fichiers de librairies de fonctions

Public

- css
- js
- images
- fonts
- index.php

Modélisation des données

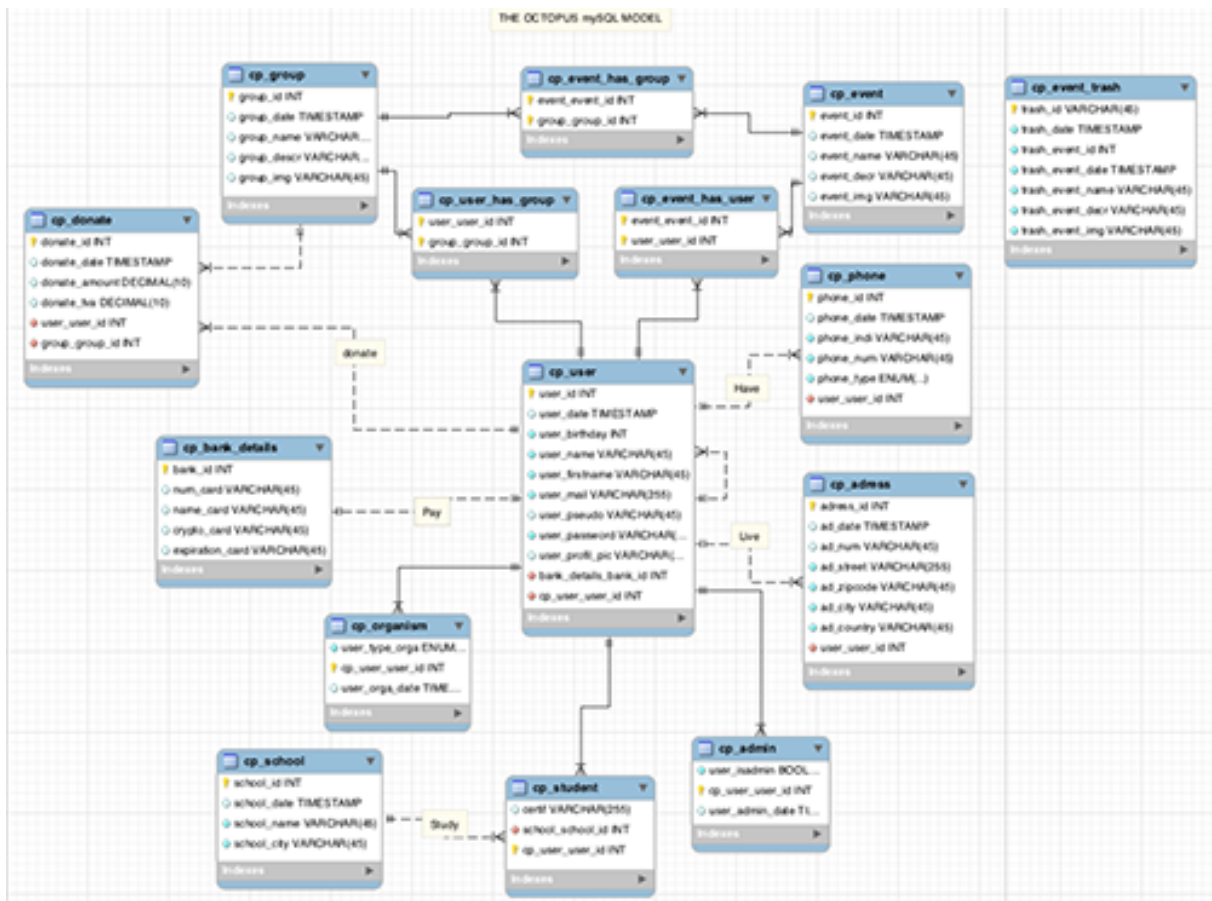
Tables:

- cp_user
 - o cp_admin
 - o cp_student
 - cp_school
 - o cp_organism
- cp_adress
- cp_phone
- cp_bank_details
- cp_donate
- cp_group
- cp_event

Explications:

- Un **USER** contient des informations globales. Il peut être un admin, un étudiant ou un organisme (entreprise ou association).
Chaque user a une table associée qui fournit les détails de son adresse postale, son téléphone, et ses informations bancaires.
Un étudiant doit fournir des informations sur son école.
Il peut effectuer différentes actions.
- Une **DONATION** peut être effectuée par un user qui est de type organisme. Les informations bancaires seront récoltées et enregistrées dans la table **cp_bank_details**.
La donation est faite sur un groupe.
- Un **GROUPE** est un ensemble de users avec une description et une image.
Il reçoit des dons et participe à des event.
- Un **EVENT** est l'évènement auquel participent les groupes.
ex: 4L Trofy
Un event peut avoir plusieurs groupes, et un groupe peut avoir plusieurs event.

- Les **TABLES INTERMEDIAIRES** permettent la relation **n:n** entre **cp_group** et **cp_event**



STRUCTURE GENERALE

CONFIGURATION générale.

Structure GitHub - Git

- **.gitignore**
Il regroupe l'ensemble des adresses de fichiers indésirables, tels que les fichiers de mots de passes, de logs, de cache etc...
- **back-cp**
Framework L&G pour le dashboard
- **coming-soon**
Ensemble des fichiers pour la page d'attente
- **front-cp**
Framework L&G pour le front
- **m.challengesplanet**
Ensemble des fichiers pour l'application Android

FRONT – (web site)

Home

Il existe 4 types d'utilisateurs.

- L'étudiant identifié, crée et rejoint des équipes liés à des événements.
- L'organisme identifié, sponsorise les équipes.
- L'utilisateur public, crée des événements.
- L'administrateur, assure les fonctions de gestion des utilisateurs et des événements.

Elle permet d'afficher des évènements, des projets, et des sponsors.

Elle permet de se connecter et de s'enregistrer.

Affichage des équipes de projets

```
/**
 * Page static INDEX
 */
public function Home(){

    // Définition des constantes
    define("PAGE_TITLE", SITE_NAME . " home");
    define("PAGE_DESCR", "Plateforme de mise en relation d'étudiants et entreprises dans le cadre de participation d'évènement sportifs à sponsoriser"); // TODO
    define("PAGE_KW", SITE_NAME); // TODO
    define("PAGE_ID", "home");

    $Group = $this->model = new PageModel();
    $AllfourEvents = $Group->SeeFourEvents();
    $AllLastGroups = $Group->SeeLastGroups();
    $DoneGroup = $Group->SeeDoneGroup();
    $AllLastSponsors = $Group->SeeLastSponsors();

    $array = '';
    $array['slider'] = $AllfourEvents;
    $array['group'] = $AllLastGroups;
    $array['sponsor'] = $AllLastSponsors;
    $array['done'] = $DoneGroup;

    $i = 0;
    foreach($array['group'] as $k => $t){
        $CountLastGroups = $Group->CountLastGroups($t['group_id']);
        $CountDonut = $Group->CountDonut($t['group_id']);
        // var_dump($CountDonut);
        $array['group'][$i] = array_merge($t, $CountLastGroups, $CountDonut);
        $i ++;
    }
    //var_dump($array['group']);

    // Appel de la vue
    $this->load->view('page', 'home', $array); // TODO
}
```

Les `defines` permettent de définir les informations de la page chargée.

\$AllfourEvents = Pour le slider, les 4 derniers évènements ajoutés.
\$AllLastGroups = Pour les teams, les 7 dernières équipes de projet ajoutées.
\$DoneGroup = Pour les teams, le dernier projet dont l'évènement est fini.
\$AllLastSponsors = Pour les sponsors, les 8 derniers sponsors ayant déjà donné.
/ Pour toutes les requêtes, les groupes, et évènements doivent être validés. */*

Construction d'un tableau **\$array**, pour passer les paramètres dans la vue.

Pour compter le nombre de sponsors pour chaque team, nous parcourons le tableau des groupes, récupérons l'`id` du groupe parcouru, exécutons la requête.
De part le return, nous complétons le tableau en ajoutant une deuxième dimension.

Page 404

Il s'agit d'une page simple, contenant un bouton pour revenir à la Home.

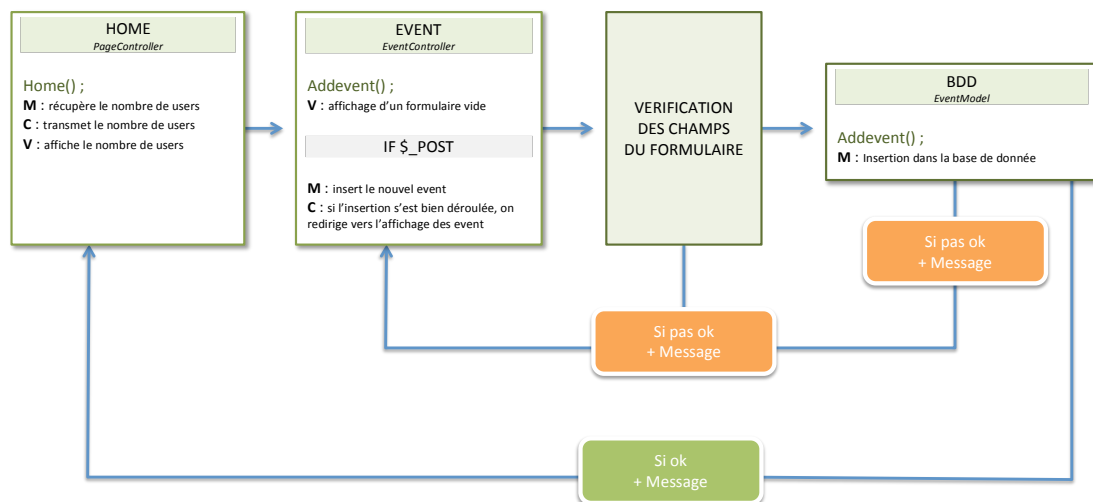
Création events

De la home, on va vers la page de création d'évènement. Sur la création de l'évènement on remplit un formulaire.

On insert dans une ligne dans la BDD avec un booléen à 0. Cela permettra le filtre lors de l'affichage des projets.

- Tester si tous les champs sont remplis
- Si ok => Home avec message. Envoi de mail *"en attente de confirmation"*
- Si nok => Retour à la page création d'évènement avec message.

CREATION D'UN EVENEMENT



Création projet/équipe/groupe

Si évènement validé + connecté

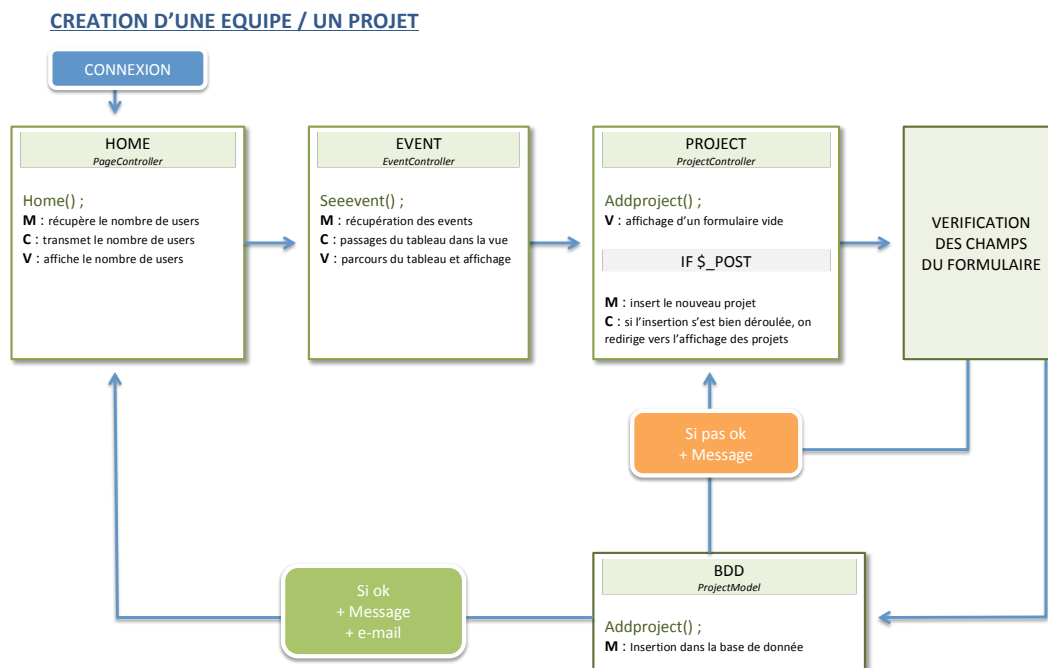
De la home, on va vers la page de création d'une équipe. Sur la création de l'équipe on remplit un formulaire.

Le formulaire envoyé, on insert dans la BDD.

- Tester si tous les champs sont remplis
- Si ok => Home avec message. Envoi de mail "*en attente de confirmation*"
- Si nok => Retour à la page création d'équipe avec message.

Si pas connecté

Message de demande de connexion



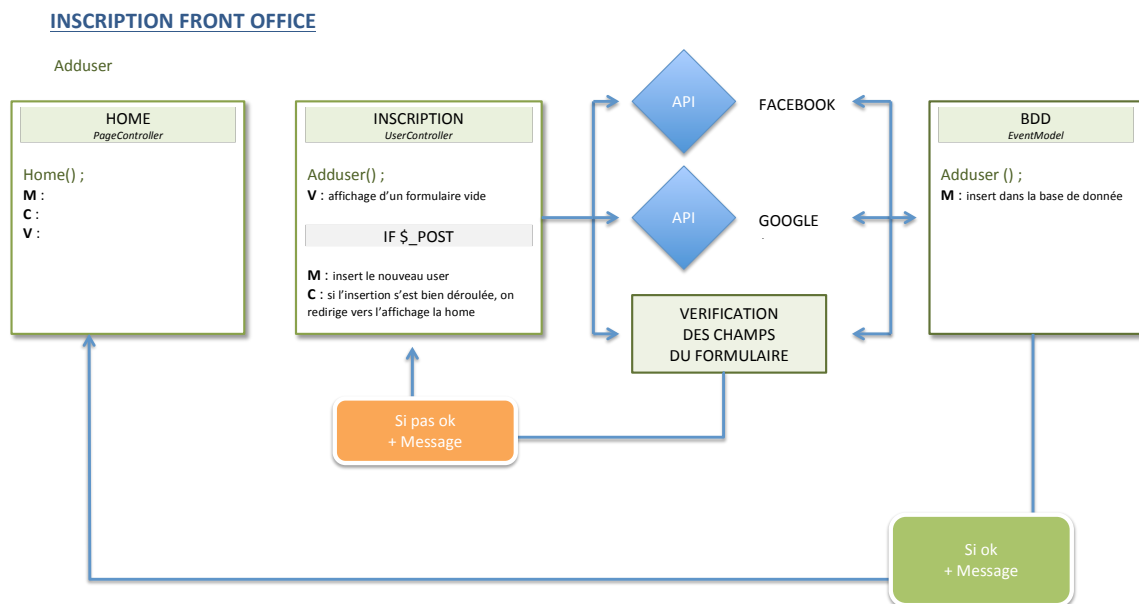
Inscription

De la home, on va sur la page d'inscription.

- API: Facebook
- API: Google +
- Manuellement

Le formulaire envoyé, on test les champs

- Tester si tous les champs sont remplis
- Si ok => Home avec message.
Envoi de mail "en attente de confirmation d'inscription"
- Si nok => Retour à la page d'inscription avec message.



Connexion

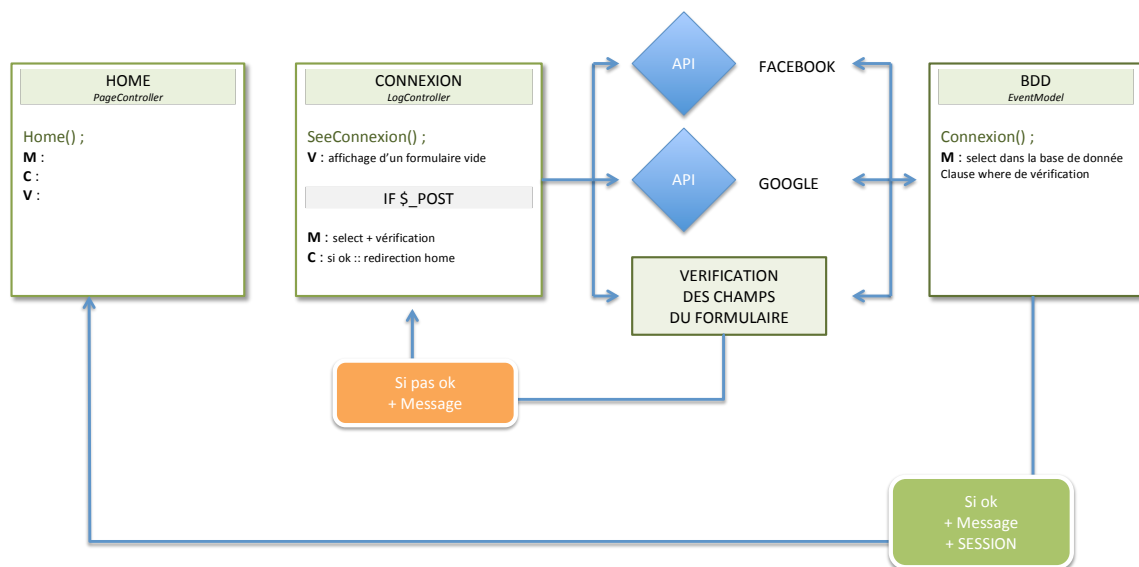
De la home, on va sur la page de connexion.
Trois types de connexion:

- API: Facebook
- API: Google +
- E-mail

Le formulaire envoyé, on test les champs

- Tester si tous les champs sont remplis.
- Si ok => Home avec changement de « connexion » en « mon compte ».
- Si nok => Retour à la page de connexion avec message.

CONNEXION FRONT OFFICE



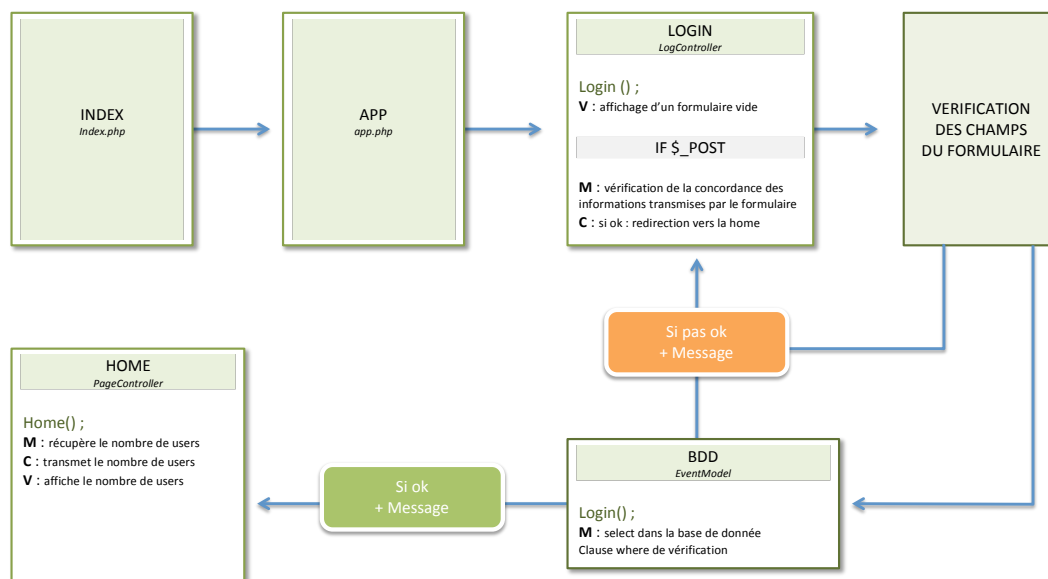
DASHBOARD - (admin)

Connexion

De la home, on va sur la page de connexion
Le formulaire envoyé, on test les champs

- Tester si tous les champs sont remplis
- Si ok => Home avec message.
- Si nok => Retour à la page de connexion avec message.

CONNEXION – DASHBOARD



Reconnexion

...

Notifications

Traitement en AJAX visant à afficher les projets à valider.

[En cours de réflexion]

Projets en cours

Affichage en css de la progression des tâches.

Gestion des projets et de l'avancée par formulaire.

[En cours de réflexion]

Messages

Voir si on peut API Gmail.

Sinon BDD puis affichage avec un booléen lu (1) ou pas lu (0).

[En cours de réflexion]

Compte

- Profile

Lien vers la page d'information du profil

- Déconnexion

Déconnexion: `session_unset` + `session_destroy`

Redirection vers la page de login.

[En cours de réflexion]

Ajouter un utilisateur

Liens vers la page d'ajout d'un utilisateur, un formulaire.

Ajouter un projet

Liens vers la page d'ajout d'un projet, un formulaire avec une liste de participant et une fonction rechercher participant.

[En cours de réflexion]

Ajouter un évènement

Liens vers la page ajout d'évènement, même formulaire que dans le front-end mais validation direct.

[En cours de réflexion]

Voir un/des utilisateur(s)

Liens vers la page qui liste les utilisateurs avec un champ de recherche avancé.

[En cours de réflexion]

Voir un/des projet(s)

Liens vers la page qui liste les projets, chaque projet donnera un aperçu des utilisateurs, avec champ de recherche avancé.

[En cours de réflexion]

Voir un/des évènement(s)

Liens vers la page qui liste les événements, chaque événement donnera un aperçu des projets, avec un champ de recherche avancé.

[En cours de réflexion]

Equipe

[En cours de réflexion]

Cookie

Obligation Cnil

La loi impose désormais aux responsables de sites et aux fournisseurs de solutions d'informer les internautes et de recueillir leur consentement avant l'insertion de cookies ou autres traceurs.

Mesure d'audience

[GAPI - Analytics](#)

SENARIOS TESTS

DASHBOARD – (admin)

Connexion

Se connecter au dashboard:

- Tester si la vérification HTML5 fonctionne (obligation de remplir tous les champs)
Vérifier que le formulaire ne s'envoie pas
- Tester avec un mauvais login, et mauvais mot de passe
Vérifier que l'on revient sur la page du login
- Tester avec un bon login et mauvais mot de passe
Vérifier que l'on revient sur la page du login
- Tester avec un bon login et bon mot de passe
Vérifier que l'on est sur la page de la home
Vérifier que la session est bien remplie

Home

Voir que l'affichage se passe correctement:

- Tester l'affichage du nombre de utilisateurs
Vérifier que le nombre de utilisateur est affiché
Vérifier qu'il affiche bien le nombre de utilisateur dans la base de donnée

Page 404

Voir que l'affichage se passe correctement.

Vérifier le lien "go home".

- Tester le lien
Vérifier que le lien renvoi bien à la home

Ajout utilisateurs

Voir que l'on ne peut pas ajouter d'administrateurs si admin simple.

Voir que l'on peut supprimer.

- Tester avec un pseudo déjà pris
Vérifier que l'on revient au formulaire avec un message d'erreur
- Tester avec un âge inférieur à 13 ans
Vérifier que l'on revient au formulaire avec un message d'erreur

Afficher projets

Voir que l'affichage se passe correctement.

Voir que l'on peut supprimer.

L'ensemble des groupes représentant un projet participant à un évènement

Afficher évènements

Voir que l'affichage se passe correctement.

Voir que l'on peut supprimer.

Afficher utilisateurs

Voir que l'affichage se passe correctement.

Voir que l'on peut supprimer.

L'ensemble des utilisateurs qui ne sont pas administrateurs.

Afficher administrateurs

Voir que l'affichage se passe correctement.

L'ensemble des utilisateurs qui sont administrateurs.

Déconnexion

Se déconnecter.

- Tester la déconnexion

Vérifier que l'on reviens bien sur la page de login / mot de passe

Vérifier que la session est bien vide.

Super admin

Voir que l'on peut supprimer les administrateurs

- Tester la suppression

Vérifier le message de suppression

Voir que l'on peut ajouter des administrateurs

- Tester l'ajout