SPECIFICATIONS TECHNIQUES

Module JAVA

Projet Carnet d’adresses

|  |  |  |  |
| --- | --- | --- | --- |
| Kévin DIEZ | Vincent TESCARI | Olivier WELTER | Vincent BRITELLES |
| **Observateur** | **Observateur** | **Observateur** | **Formateur** |

PAGE DE REVISION

|  |  |  |  |
| --- | --- | --- | --- |
| **Rév.** | **Date** | **Paragraphe** | **Modification** |
| 00 | 27/03/2017 | Tous | Création du document |
| 01 | 07/02/2017 | Tous | Relecture et validation du document |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |



SOMMAIRE

1 introduction 4

2 Specifications logicielles 4

2.1 Considérations générales 4

2.2 Structure des répertoires 4

2.3 Convention de nommage 4

3 specifications donnees et interface/flux 5

3.1 Ecole 5

3.2 Classe 6

3.3 Personnel 6

3.4 Elèves 7

3.5 Parent (coordonnées précisées sur chaque parent si différentes.) 7

3.6 Bulletin scolaire : 8

4 Compôsants systemes 8

4.1 Poste Client 8

4.2 Serveurs 8

5 ARCHITECTURE MATERIELLE 8

5.1 Serveurs 8

5.2 Sauvegardes 8

# introduction

Il s’agit d’un document purement technique à destination des développeurs. Il leur permet de se lancer dans le code sans laisser d'ambiguïté sur le contenu à coder et la structure du programme. On ne détaillera ici que le logiciel client, le logiciel sur le serveur étant en PhP nous ne donnerons pas de détail sur celui-ci.

* Les spécifications logicielles
* Les spécifications données et interface/flux
* Les composants systèmes
* L’architecture Matérielle

# Specifications logicielles

## Considérations générales

Les pages principales proposant une IHM seront développées avec HTML 5 puis mises en forme avec du CSS 3. Les avertissements issus de la validation W3C seront tolérés, cependant, en aucun cas des erreurs devront subsister. La gestion d’évènements sera gérée en JavaScript, l’utilisation de Framework ou plugin dépendra du besoin des développeurs. Le back-end, développé en PHP permettra de traiter l’ensemble des requêtes reçues et émises par le serveur ainsi que les interactions avec la base de données qui sera de type mySQL.

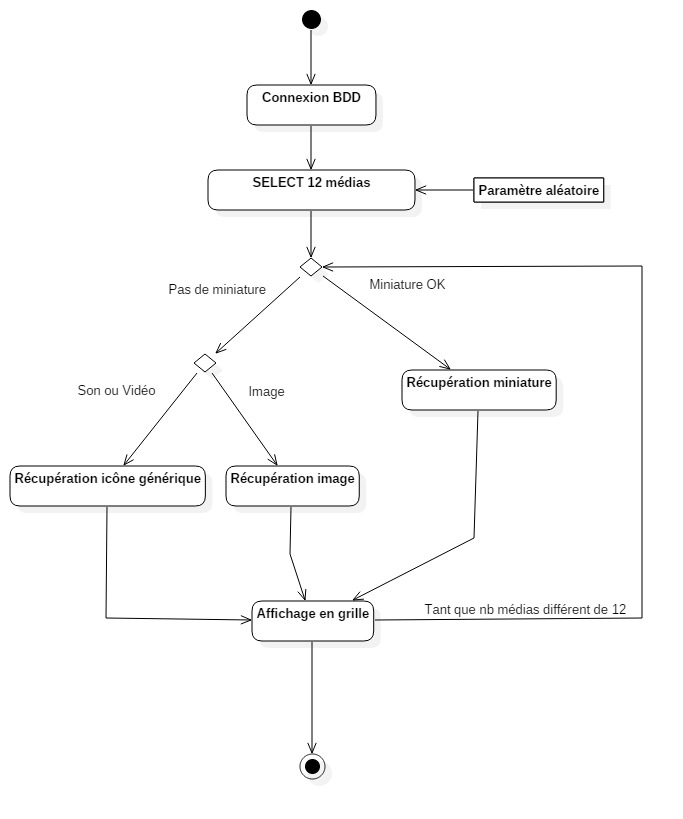
## Convention de nommage

Afin de faciliter la compréhension et l’utilité d’un fichier, d’une fonction, méthode, classe ou variable, ceux-ci devront être nommé de la façon la plus explicite possible. Les noms composés seront donc gérés en utilisant la syntaxe de « Camel Case », aussi, ils pourront contenir des traits d’union ou « Under scores » afin de rentre leur lecture et leur compréhension plus aisée.

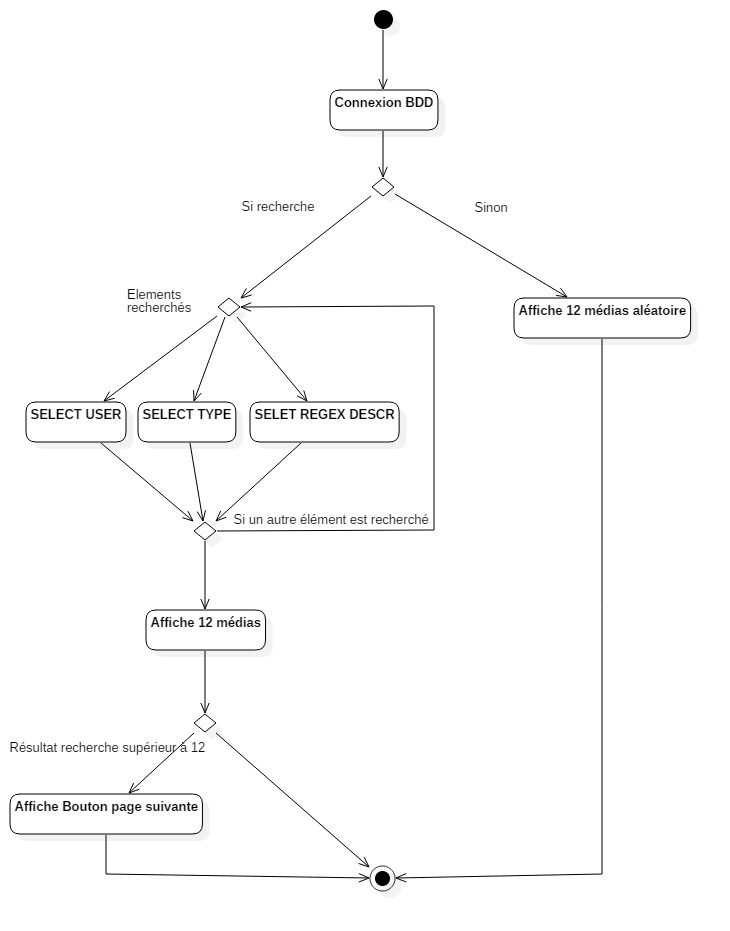
## Algorithmique

Cette partie permet de schématiser l’algorithme simplifié suivi par chacune des principales fonctionnalités de l’application. Pour les représenter, un diagramme UML d’activités est utilisé.

### Afficher 12 médias



### Rechercher des médias

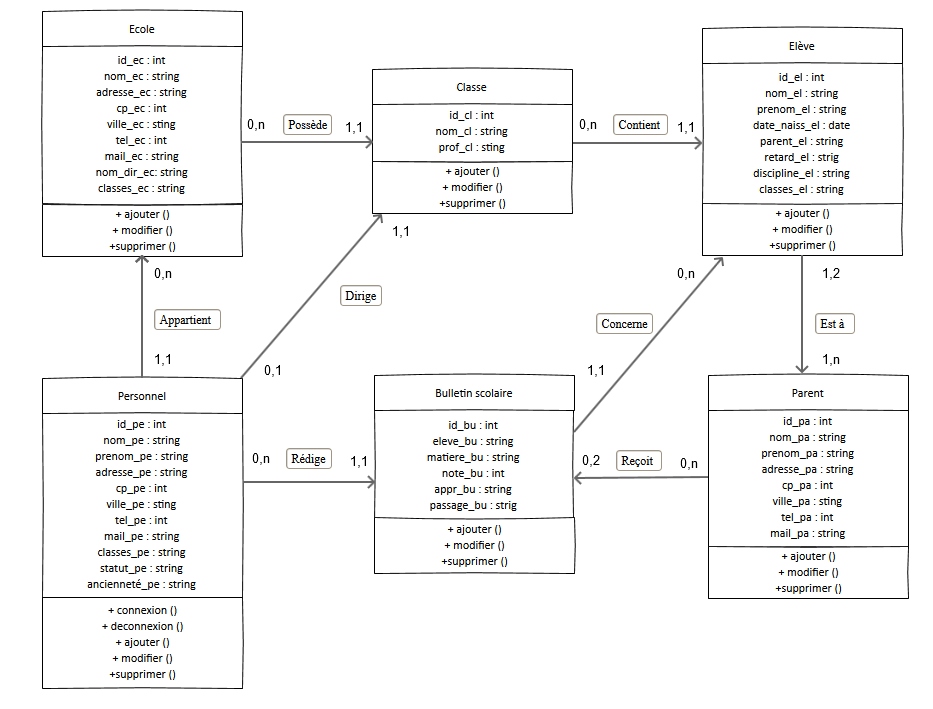


## Structure des répertoires

Tous les fichiers seront rangés dans un unique dossier qui contiendra tout les fichiers HTML de toute les pages de l'application. Le dossier contiendra les dossier 'js', 'css', 'img', 'font', qui contiendront, respectivement, tous les fichiers JavaScripts, les fichiers de styles, les images et les polices.

# specifications donnees et interface/flux

Les objets seront définis comme sur le schéma suivant:



## Ecole

**Nom** : maximum 50 caractères.

**Coordonnées** :

Adresse : maximum 100 caractères.

Code postal : 5 caractères.

Ville: maximum 50 caractères.

Numéro de téléphone : 10 caractères.

Mail : maximum 100 caractères.

**Directeur** :

Nom : maximum 50 caractères.

Prénom : maximum 50 caractères.

**Classes** :

Niveau (CP, CE1, CE2, CM1, CM2) : maximum 15 caractères, une classe peu être mixte.

Numéro : maximum 2 caractères.

## Classe

**Nom** : Niveau + Numéro (exemple CP 3)

**Professeur** : Nom et prénom

**Élèves** : Nom et prénom

## Personnel

**Nom** : maximum 50 caractères.

**Prénom** : maximum 50 caractères.

**Adresse** : maximum 100 caractères.

**Code** **postal** : 5 caractères.

**Ville**: maximum 50 caractères.

**Numéro** de téléphone : 10 caractères.

**Mail** : maximum 100 caractères.

**Classe** : Nom de la classe

**Statut** : admin ou professeur.

**Ancienneté** : maximum 10 caractères.

## Elèves

**Nom** : maximum 50 caractères.

**Prénom** : maximum 50 caractères.

**Date** **de** **naissance** : 8 caractères.

**Parent** : nom + prénom.

**Retards** : maximum 500 caractères.

**Discipline** : maximum 200 caractères.

**Classe** : Nom de la classe.

## Parent (coordonnées précisées sur chaque parent si différentes.)

**Parent** **1** :

**Nom** : maximum 50 caractères.

**Prénom** : maximum 50 caractères.

**Numéro** **de** **téléphone** : 10 caractères.

**Mail** : maximum 100 caractères.

* Adresse : maximum 100 caractères.
* Code postal : 5 caractères.
* Ville: maximum 50 caractères.

**Parent** **2** :

**Nom** : maximum 50 caractères.

**Prénom** : maximum 50 caractères.

**Numéro** **de** **téléphone** : 10 caractères.

**Mail** : maximum 100 caractères.

* Adresse : maximum 100 caractères.
* Code postal : 5 caractères.
* Ville: maximum 50 caractères.

## Bulletin scolaire :

**Nom** **élève** + prénom.

**Matière** : maximum 20 caractères.

**Note** : maximum 5 caractères.

**Appréciations** **pédagogiques** : maximum 100 caractères.

**Passage** **en** **classe** **supérieure** : maximum 50 caractères.

# Compôsants systemes

## Poste Client

Les clients vont utiliser l'interface web HTML pour interagir avec la base de données, comme on code en HTML5 et qu'on utilise css3 il faudra probablement mettre a jour les navigateurs web des clients.

## Serveurs

Un serveur web et local (Apache) installer sur la machine qui tournera sous une version de linux (Debian). Le serveur devra impérativement pouvoir interpréter le PhP.

# ARCHITECTURE MATERIELLE

## Serveurs

Un Intel i5 avec 8 giga de RAM et un disque dur de 2 terra.

## Sauvegardes

2 Machines identiques au serveur.