

# Cahier des charges

Mini projet BDR

## Système de gestion des écoles

Groupe 6

Professeur : René Rentsch

Assistant : Sébastien Rosat

## Table des matières

1. Introduction .....	3
2. Objectif .....	3
3. Description des fonctionnalités et données nécessaires .....	4
3.1 Horaire.....	4
3.2 Étudiants.....	4
3.3 Professeurs.....	4
3.4 Bâtiments et salles de cours .....	5
3.5 Leçons.....	5
3.6 Cours.....	5
3.7 Notes .....	6
3.8 Historique des données .....	6
3.9 Statistiques .....	6
3.9.1. Statistiques sur les horaires .....	6
3.9.2. Statistiques sur les notes .....	7
3.9.3. Statistiques sur les étudiants .....	7
3.10 Parking et macaron.....	7
4. Interface utilisateur.....	8
4.1 Gestion des personnes .....	8
4.2 Gestion des personnes .....	8
4.3 Gestion des leçons .....	9
5. Logiciels utilisés .....	9
6. Si le temps le permet .....	10

## 1. Introduction

Le but de ce mini-projet du cours BDR est de mettre en pratique la théorie vue durant le cours afin de créer une application complète utilisant une base données relationnelle. Ce projet se déroule en plusieurs phases. La première étant le choix du sujet ainsi que l'analyse des besoins. Par la suite aura lieu la modélisation conceptuelle et relationnelle puis le projet se terminera par une implémentation du tout avec la création d'une application permettant de gérer les données via une interface utilisateur conviviale. Ce présent document a pour but de définir les besoins en données ainsi que les besoins fonctionnels tout en décrivant de manière générale et détaillée le choix du projet.

## 2. Objectif

Dans le cadre de ce mini-projet du cours BDR, nous avons décidé de créer une application pour la gestion d'une école. Le but de cette dernière est de permettre aux écoles une gestion simplifiée de tout le secteur "Formation et éducation".

La structure d'une école étant relativement complexe, nous l'avons séparée en deux axes principaux. D'un côté la partie liée à la formation et l'éducation des étudiants et de l'autre la partie liée à la gestion de l'école (personnel administratif et technique). Pour des raisons de simplification, ce mini-projet s'adresse donc uniquement à la partie formation et éducation.

Ce programme permet donc de gérer toutes les entités liées à la formation dans une école à savoir :

- Les étudiants
- Les professeurs
- Les salles de cours
- Les horaires
- Les cours
- Les notes

La gestion de cette application se fait via une page web simple et moderne. La description détaillée de toutes les entités et les possibilités se trouvent ci-dessous.

### 3. Description des fonctionnalités et données nécessaires

#### 3.1 Horaire

L'horaire d'une entité (étudiant, professeur ou salle de classe) correspond aux différentes leçons qui la concernent. Pour des raisons de simplification, les horaires ne sont pas générés automatiquement. La liste des leçons suivies par un étudiant doit être manuellement définie. Lors de l'attribution d'une leçon à un étudiant, des vérifications sont faites pour éviter que des leçons se chevauchent. Les horaires des professeurs sont quant à eux automatiquement définis en fonction des leçons qu'ils donnent. De manière similaire, l'horaire d'une salle dépend des leçons qui y ont lieu. Un horaire est valable que pour 1 seul semestre, lors du semestre suivant, les leçons peuvent changer et de se fait l'horaire avec.

#### 3.2 Étudiants

Les étudiants sont caractérisés par les attributs suivants :

- Nom
- Prénom
- Date Naissance
- Raison du départ
- Souhaite avoir un macaron pour le parking
- Distance en km du domicile (Distance domicile-école)

Les étudiants peuvent participer à des leçons. Il est possible d'afficher l'horaire ainsi que les notes et la moyenne d'un cours spécifique pour un étudiant. Un étudiant ne peut pas participer à 2 leçons différentes qui ont lieu en même temps. Pour un cours donné, un étudiant ne peut que participer à des leçons données par le même professeur. Il ne peut pas y avoir d'étudiant qui suit 2 cours de semestre différents mais d'une même année. La raison de départ d'un étudiant peut être soit : réussite, échec ou arrêt. La raison du départ peut être vide si l'étudiant est toujours en cours d'étude.

#### 3.3 Professeurs

Les professeurs sont caractérisés par les attributs suivants :

- Nom
- Prénom
- Date Naissance
- Trigramme
- Souhaite avoir un macaron pour le parking

Les professeurs peuvent donner des cours. Il est possible d'afficher l'horaire d'un professeur (les leçons qu'il donne). Un professeur ne peut pas donner 2 leçons différentes qui ont lieu en même temps.

### 3.4 Bâtiments et salles de cours

Une école peut avoir différents bâtiments qui possèdent les différentes salles de cours. Les numéros de salles de cours sont relatifs aux numéros de bâtiments. Les données nécessaires sont :

- No du bâtiment
- S'il possède un parking ou non. Si oui, son nombre de place max
- Liste des salles de cours d'un bâtiment

Il est possible d'afficher les horaires d'une salle (leçons qui l'utilisent). Les salles ne sont pas forcément toujours occupées.

### 3.5 Leçons

Les leçons sont le point central de l'application. Elles permettent de définir les périodes de cours dans une certaine salle d'un bâtiment. Une journée est séparée en X périodes de Y minutes. Une leçon est donnée par un seul professeur. Chaque cours possède plusieurs leçons qui correspondent aux différentes périodes durant la semaine. Les leçons ne sont valables que durant 1 semestre. Les données nécessaires sont :

- Période de début
- Nombre de périodes
- Jour de la semaine
- Semestre
- Année d'étude

### 3.6 Cours

Les cours font références aux différentes matières enseignées (par ex: Maths, Français, Histoire). Un cours donné possède une ou plusieurs leçons d'enseignement. Il peut y avoir plusieurs professeurs qui donnent le même cours mais pour des leçons différentes. Pour chaque cours, des évaluations écrites peuvent être effectuées.

Les cours sont caractérisés par :

- Nom
- Nombre d'évaluations écrites
- Nombre d'évaluations pratiques

### 3.7 Notes

Il est possible d'enregistrer les notes de chaque élève pour chaque cours suivi. Il existe deux types de notes :

- Travaux écrits
- Travaux pratiques

La moyenne d'un élève pour un cours donné est calculée en faisant la somme des notes et en la divisant par le nombre d'évaluations du cours. Il faut également prendre en compte leur pondération (0.8 pour les travaux écrits, 0.2 pour les travaux pratiques). La note de 1.0 est attribuée pour chaque évaluation non effectuée.

Une note est caractérisée par :

- Note
- Type (TE, TP)

### 3.8 Historique des données

Comme mentionné plus haut, les leçons ne sont valables que durant un seul semestre. A la fin de ce dernier, les données restent enregistrées à des fins statistiques. Le nouveau semestre, l'attribution de leçons aux étudiants ainsi qu'aux professeurs doit se faire manuellement. Ce programme ne permet pas de générer cela automatiquement. Un étudiant pourra alors suivre des nouvelles leçons et aura de ce fait un horaire différent du semestre précédent.

### 3.9 Statistiques

Grâce aux données précédentes, une interface permet d'obtenir des statistiques pour chaque semestre d'étude. Il est possible de sélectionner individuellement chaque semestre pour obtenir ses statistiques ou de sélectionner plusieurs semestres afin de les comparer.

Plusieurs types de statistiques existent :

- Statistiques sur les horaires
- Statistiques sur les notes
- Statistiques sur les étudiants

#### 3.9.1. Statistiques sur les horaires

- Nombre d'enseignants impliqués dans des cours
- Nombre de salles utilisées (taux d'occupation)
- Nombre moyen de cours suivis par étudiant
- Le nombre moyen de leçon par élève ou professeur
- Le taux d'élèves asynchrones (élève qui suit des leçons ayant des années d'études différentes)

### 3.9.2. Statistiques sur les notes

- Taux d'échec ou de réussite d'un cours
- Moyenne générale d'un étudiant pour tous les cours qu'il suit
- Comparer le taux de réussite d'un même cours mais donné par des enseignants différents

### 3.9.3. Statistiques sur les étudiants

- Taux d'étudiants ayant terminé leurs études avec diplôme
- Taux d'étudiants ayant arrêté leurs études
- Taux d'étudiants ayant échoué leurs études

## 3.10 Parking et macaron

Comme vu précédemment, une école possède plusieurs bâtiments qui peuvent ou non avoir un parking. Cette fonctionnalité permet de générer la liste des personnes (élèves ou professeurs) éligibles à un macaron pour le parking.

Les règles pour l'attribution d'un macaron à une personne sont les suivantes :

- Le taux maximum de places associées à un macaron est de 50% (calculé sur la somme totale des places des différents parking). Les 50% restant sont libres.
- Les élèves peuvent occuper max 60% des macarons à disposition, les 40% restant pour les professeurs.
- Les professeurs sont ensuite choisis aléatoirement parmi tous ceux qui souhaitent obtenir un macaron.
- Les étudiants sont choisis en fonction de la distance de leur lieu de domicile par rapport à l'école. Ils sont éligibles seulement s'ils habitent à plus de 20km de l'école. S'il y a plus d'élèves éligibles que de places disponibles, ils sont choisis aléatoirement.
- S'il reste des macarons disponibles, ils sont à disposition au secrétariat.

## 4. Interface utilisateur

Les mockups ci-dessous présentent certaines des interfaces qui seront implémentées. Elles sont donc susceptibles de changer par la suite.

Chaque entité principale aura une page individuelle avec toutes ses informations de base affichées et plus si nécessaire.

### 4.1 Gestion des personnes

La création et la modification des élèves se feront de manière similaire pour les professeurs. Les interfaces seront donc les mêmes. Les élèves et les professeurs ont leurs horaires respectifs affichés sur leur page individuelle. Sur la page d'un élève il sera aussi possible de lui ajouter des leçons.

### 4.2 Gestion des personnes

La création et la modification des élèves se feront de manière similaire pour les professeurs. Les interfaces seront donc les mêmes. Les élèves et les professeurs ont leurs horaires respectifs affichés sur leur page individuelle. Sur la page d'un élève il sera aussi possible de lui ajouter des leçons.

Nom	Prénom	Date de Naissance	
Giacomo	Guizzoni	18.12.1996	<input type="checkbox"/>
Marco	Botton	8.12.1998	<input checked="" type="checkbox"/>
Mariah	MacLachlan	10.11.2001	<input type="checkbox"/>
Valerie	Liberty	23.11.2000	<input checked="" type="checkbox"/>

Supprimer

Nom:  Prénom:  Date de Naissance:

Ajouter



Tables

- Elèves
- Professeurs
- Leçons
- Cours
- Semestres

### Leçons de Giacomo Guilizzoni

Période de début	Nombre de périodes	Jour de se	Semestre	Cours	Professeur	
8h30	2	Lundi	20-21 1	PRG1	RRH	<input type="checkbox"/>
10h25	1	Mercredi	21-22 2	PRG2	RRH	<input checked="" type="checkbox"/>
13h15	3	Mardi	19-20 1	BDR	RRH	<input type="checkbox"/>
14h50	1	Vendredi	20-21 2	MATH1	RRH	<input checked="" type="checkbox"/>

Ajouter une leçon

ComboBox

## 4.3 Gestion des leçons

Tables

- Elèves
- Professeurs
- Leçons
- Cours
- Semestres

### Leçons

Période de début	Nombre de périodes	Jour de se	Semestre	Cours	Professeur	
8h30	2	Lundi	20-21 1	PRG1	RRH	<input type="checkbox"/>
10h25	1	Mercredi	21-22 2	PRG2	RRH	<input checked="" type="checkbox"/>
13h15	3	Mardi	19-20 1	BDR	RRH	<input type="checkbox"/>
14h50	1	Vendredi	20-21 2	MATH1	RRH	<input checked="" type="checkbox"/>

periode de début Nb de périodes Jour Semaine Semestre Cours Professeur

ComboBox 2 ComboBox ComboBox ComboBox ComboBox

## 5. Logiciels utilisés

Pour la création de cette application, nous avons décidé d'utiliser les outils suivants. La base de données sera créée avec PostgreSQL. Ce choix est imposé pour ce projet. L'interface de l'application sera un site web, développé avec le langage PHP. Nous utiliserons la librairie Bootstrap pour la mise en forme graphique ainsi que le langage CSS afin de créer des interfaces simples et modernes.

## 6. Si le temps le permet

Si le temps le permet les fonctionnalités suivantes pourront être intégrées dans l'application :

- Gestion des examens
- Gestion des orientations
- Gestion des assistants de cours