Compte-rendu de projet

SCHOLASTICA

Emanuelle Laurent

Nabile Ouchane

Romain-Pascal Saillot

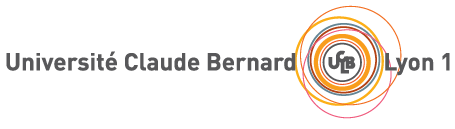


Table des matières

[1. Introduction 3](#_Toc517289119)

[1.1. Contexte 3](#_Toc517289120)

[1.2. Objectif 3](#_Toc517289121)

[1.3. Collecte d’informations 3](#_Toc517289122)

[2. Le cahier des charges 4](#_Toc517289123)

[3. Conception 6](#_Toc517289124)

[3.1. Planification du projet 6](#_Toc517289125)

[3.2. Conception de la base de données 6](#_Toc517289126)

[4. Choix techniques 8](#_Toc517289127)

[4.1. IDE 8](#_Toc517289128)

[4.2. Java 8](#_Toc517289129)

[4.3. Java IHM et la palette de NetBeans 8](#_Toc517289130)

[4.4. Frameworks Java 8](#_Toc517289131)

[4.5. UML 8](#_Toc517289132)

[4.6. Conception Merise 9](#_Toc517289133)

[4.7. MySQL 9](#_Toc517289134)

[4.8. Git et GitHub 9](#_Toc517289135)

# Introduction

## Contexte

Il existe de nombreux logiciels pour la gestion des établissements scolaires mais ceux-ci sont plutôt destinés aux collèges et aux lycées, sont payants ou ne répondent pas aux besoins réels d’une école primaire.

## Objectif

On propose de créer un logiciel gratuit qui permette la gestion d’une école primaire, des élèves, de leurs parents, de l’équipe enseignante ainsi que des classes.

## Collecte d’informations

Les informations sont collectées sous forme de questionnaire papier distribué aux élèves en début d’année.

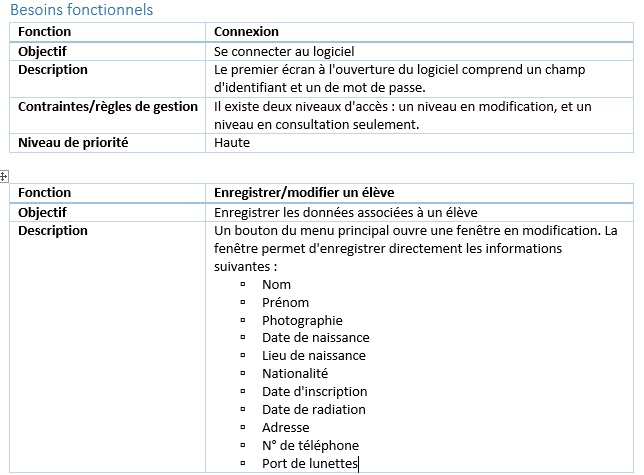
# Planification

## Cahier des charges

Après nous être mis d’accord sur le projet à réaliser, nous avons rédigé le cahier des charges.

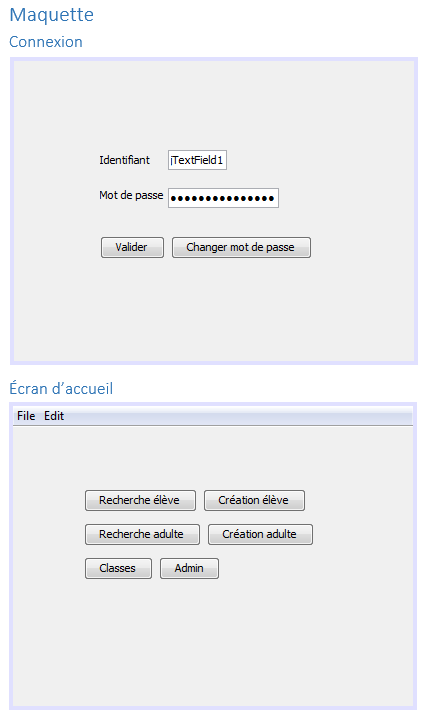
Dans celui-ci, nous avons défini les besoins d’une école primaire et donc les fonctionnalités que devra intégrer notre logiciel.

Exemple de fonctionnalités :



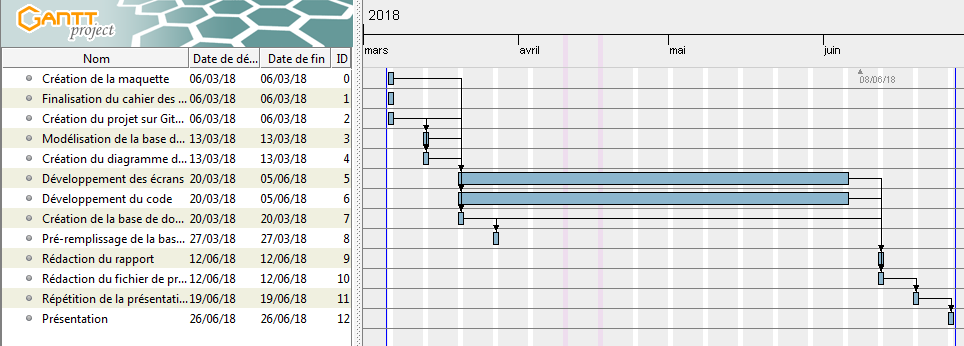
Nous avons aussi commencé à réfléchir sur les différentes fenêtres de l’application, leurs rôles et leurs liens. Nous avons donc établi la maquette pour avoir un aperçu simplifié de la vue.

Exemple d’écrans tirés de la maquette :



## Diagramme de GANTT

Nous avons établi un diagramme de GANTT afin de pouvoir organiser les différentes tâches à effectuer pour mener le projet à bien en respectant le temps qui nous a été imparti.



# Choix techniques

## IDE

Nous avions précédemment utilisé NetBeans en cours, donc nous avons naturellement décidé de continuer à l’utiliser pour le projet, d’autant qu’il est disponible gratuitement.

## Java

Le choix du langage de développement nous a été imposé par la consigne.

## Java IHM et la palette de NetBeans

La consigne nous a aussi imposé d’utiliser les IHM natives Java pour l’interface utilisateur. Pour gagner du temps et faciliter la mise en page, et parce que nous utilisions tous le même IDE, nous avons décidé d’utiliser la palette de NetBeans.

Ce choix de la palette a parfois posé problème, parce que nous l’avions très peu utilisée précédemment et nous n’avions pas encore acquis d’automatismes. De plus, les morceaux de code trouvés sur internet ont généralement dû être adaptés pour pouvoir être utilisés avec du code généré automatiquement par la palette.

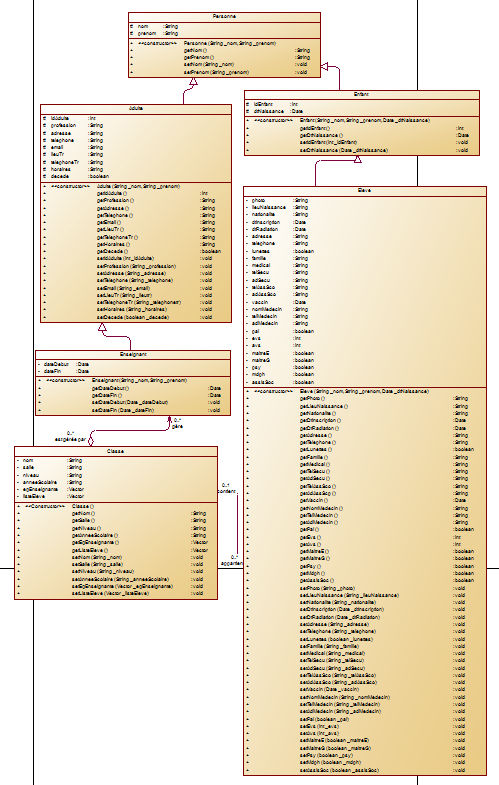
## Frameworks Java

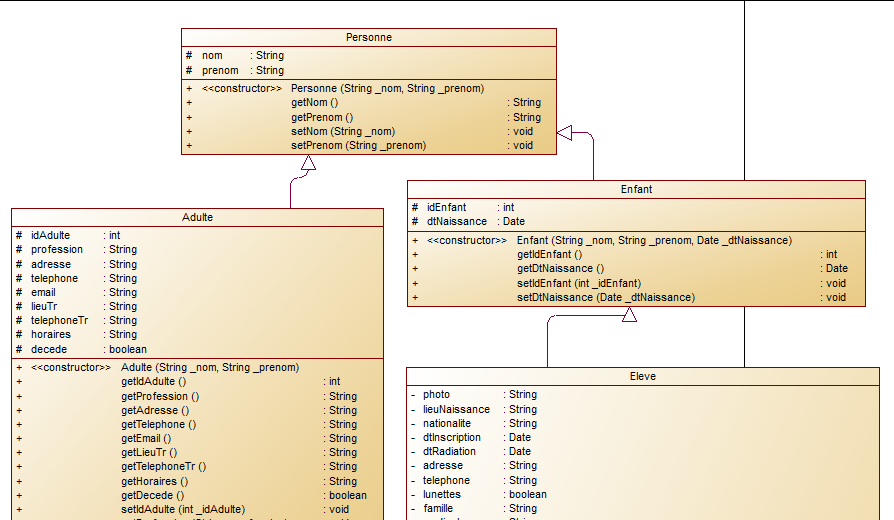
Nous avons réfléchi à la possibilité d’utiliser le framework Hibernate, et nous avons finalement décidé de ne pas le faire. Le temps que nous aurions pu gagner en utilisant un framework aurait été perdu en cherchant à apprendre comment utiliser ce framework.

Nos connaissances actuelles nous ont suffi pour écrire le projet.

## UML

Nous avons utilisé un diagramme de classe en UML sur PowerAMC pour générer le squelette du code Java. Les limitations de PowerAMC nous ont obligé à modifier manuellement les noms de méthodes dans NetBeans.

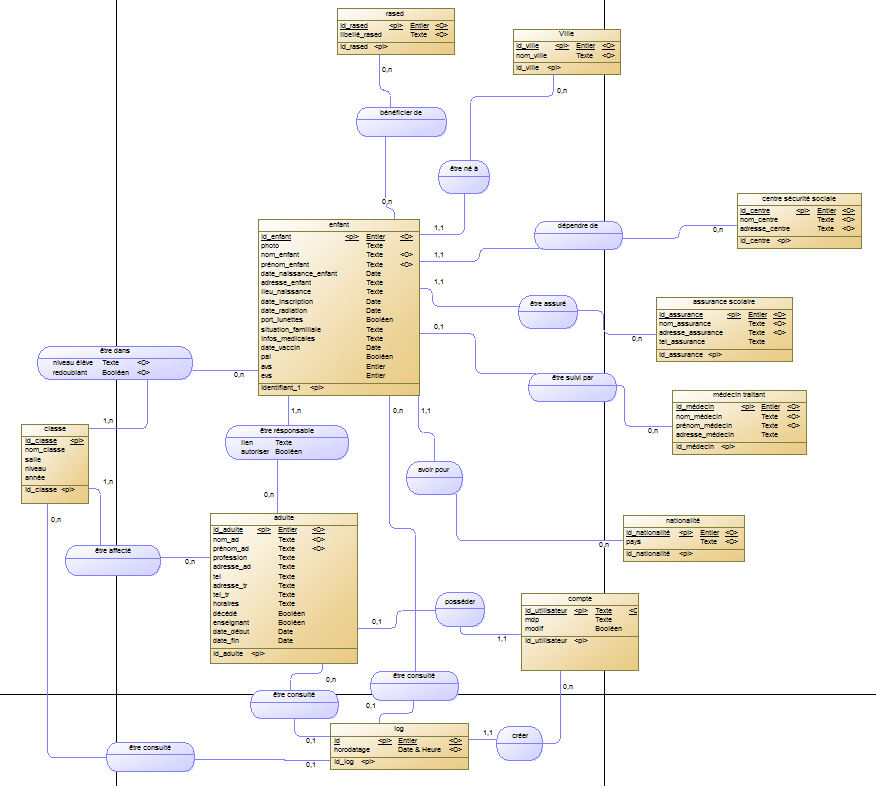


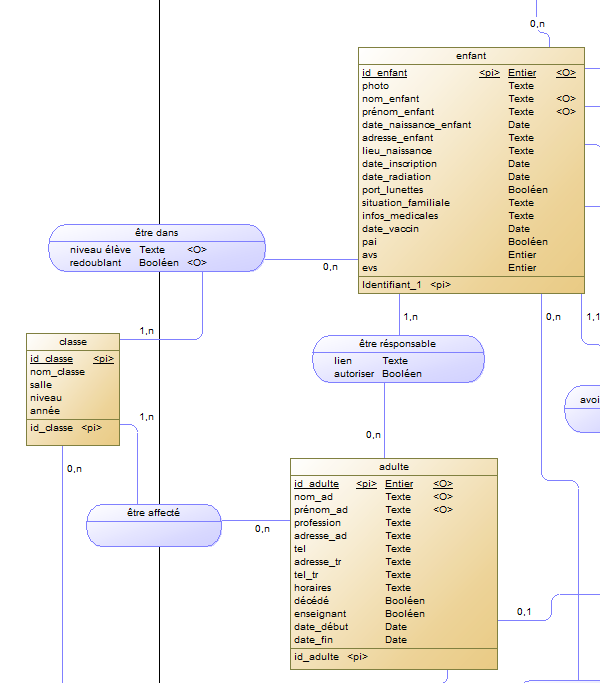


## Conception Merise

Nous avons modélisé la base de données en utilisant Merise sur PowerAMC.

Le code SQL généré par PowerAMC pour créer la base ne fonctionnant pas, nous avons finalement créé la base de données manuellement.





## MySQL

Nous avons hésité entre plusieurs RGBD : Oracle Server ou MySQL que nous avions utilisé en cours ou PostgreSQL que certains d’entre nous utilisent professionnellement.

Notre choix s’est finalement porté sur MySQL parce que, en tant qu’étudiants de Lyon 1, nous avions accès à la base implémentée pour chaque étudiant, ce qui résolvait le problème de l’installation. Le fait que la base soit accessible en ligne permettait aussi de facilement travailler à plusieurs depuis différents endroits.

## Git et GitHub

Nous avons utilisé Git en tant qu’outil de gestion de version. Nous avons créé un dépôt sur GitHub sur lequel nous nous sommes connectés en utilisant Git Bash. Certains d’entre nous utilisent professionnellement ce logiciel et ce site, et cette expérience nous a permis d’apprivoiser plus facilement l’utilisation de Git, et de résoudre les quelques problèmes de merge qui se sont posés au cours du projet.

Nous avons chacun créé des branches sur lesquelles nous avons travaillé indépendamment sur certaines fonctionnalités, et nous avons ensuite mis notre travail en commun sans problèmes majeurs. Git nous a aussi permis de travailler aussi bien de l’IUT que de chez nous.

# Bilan

## Points réussis

Le logiciel est fonctionnel

Résolution des problèmes en équipe

Bonne répartition des tâches

Respect des procédures

Phase de test

Maîtrise de nouveaux outils (palette, MySQL et SQL/PSM, Git)

## Axes d’amélioration

Ce qui était dans le cahier des charges et qui n’a pas été fait

La planification

Les objets Java

Le modèle MVC

# Conclusion