

Spécifications

School’ In

Logo

Version

31/03/2015

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 21/10/2015 | 1 | 1er regroupement de doc | Victor |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sommaire

1 Introduction 4

1.1 Mission 4

1.2 Objectifs 4

1.3 Glossaire 4

2 Description générale 5

2.1 Acteurs 5

2.2 Cas d’utilisations 5

3 Spécifications fonctionnelles 6

3.1 Carte de navigation 6

3.2 Détails des cas d’utilisations 6

4 Spécifications non fonctionnelles 9

4.1 Contraintes de conception et d’implémentation 9

4.2 Documentation utilisateur 9

# Introduction

## Mission

Nous allons concevoir et réaliser une application client de gestion d’une école afin de faciliter le travail du service pédagogique. Cette application permettra à un établissement de réaliser des taches tel que l’assignation des professeurs à différentes classes, de définir si un élève passe de classe ou redouble.

## Objectifs

* Faciliter la gestion administrative d’une école.
* Gestion du budget.
* Gestion du planning et des salles.
* Une interface d’utilisation simple.
* Optimiser le suivi pédagogique des élèves.
* Gestion des absences.

## Glossaire

### Termes du domaine

School'In: Nom du projet.

Trello : Trello est un outil de gestion de projet en ligne.

Intech’ Info : L'école à laquelle appartient les membres de l'équipe School'IN.

### Termes techniques

Pascal case : C'est une convention qui spécifie que la première lettre de chaque mot doit être en majuscule.

Nunit : Nunit est un cadre de travail open source de tests unitaires, pour Microsoft.net

Visual Studio : C'est un ensemble complet d'outils de développement permettant de générer des applications.

# Description générale

## Acteurs

|  |  |
| --- | --- |
| **Nom** | **Description** |
| Administrateur | L’administrateur est celui qui fait la gestion de l’Api. C’est lui l’acteur principal, il interagit le plus avec le système, il a pour mission de :   * Créer l’école. * Créer un étudiant. * Gérer une école. * Créer des classes. * Créer planning. * Modifier planning. * Ajouter des professeurs. * Assigne un élève à une classe. * Assigne un professeur à une classe. * Exclure un élève. * Renvoyer un professeur. |
| Elève | L’élève est un utilisateur de l’outil, il fait partie des acteurs principal qui interagit assez régulièrement avec le système, il a comme activité :   * Consulter l’emploi du temps. * Consulter ses notes. |
| Professeur | Le professeur est un acteur comme l’élève du système, lui aussi interagit assez régulièrement avec le système en ;   * Consultant l’emploi du temps. * Mettant la note d’un élève. |

## Cas d’utilisations

1. Consulter l’emploi du temps.
2. Consulter notes.
3. Créer une école.
4. Gérer une école.
5. Ajouter un professeur.
6. Créer classe.
7. Créer planning
8. Modifier planning
9. Assigner un élève.
10. Assigner un professeur.
11. Exclure un élève.
12. Renvoyer un professeur.
13. Mettre la note d’un élève.
14. Supprimer une école.

### Diagramme de cas d’utilisation

****

### Listes des cas d’utilisations

|  |  |  |  |
| --- | --- | --- | --- |
| **Cas d’utilisations** | **Acteurs** | **Priorité** | **Risques** |
| Consulter l’emploi du temps | Elève, Professeur | Avant d’accéder à une salle, le professeur ou l’étudiant devra consulter son emploi du temps | Voir un emploi du temps obsolète si le contenu de l’API n’a pas été mis à jour. |
| Mettre notes | Professeur |  | Risque que l’élève ne reçoive pas sa note ou ne la retrouve pas dans l’outil. |
| Consulter Notes | Elève |  | Ne pas pouvoir accéder à sa moyenne |
| Créer école | Administrateur |  | Retrouver des données anciennes dans une école nouvellement créé. |
| Gérer école | Administrateur | Pour pouvoir permettre à l’école de fonctionner | Mauvaise manipulation de l’outil |
| Créer classes | Administrateur | Rajouter des classes pour des nouvelles promotions | Mauvaise manipulation |
| Créer planning | Administrateur | Bien s’assurer que les classes sont disponibles | Mauvaise manipulation de l’outil |
| Modifier planning | Administrateur | Corriger erreur s’il ya conflit des salles par exemple | Mauvaise manipulation de l’outil |
| Ajouter un professeur | Administrateur | Rajouter un professeur s’il Ya manque d’effectif | Risque d’ajouter un professeur déjà existant. |
| Assigner un élève à une classe | Administrateur | L’élève assiste aux cours de son niveau en allant dans la bonne salle | Mettre un élève dans une mauvaise classe |
| Assigner un professeur à une classe | Administrateur | Permet à un prof de ne pas se tromper de filière | Mettre un professeur dans une mauvaise classe |
| Exclure un élève | Administrateur |  | Renvoyer un élève à la place de l’autre |
| Renvoyer un professeur | Administrateur |  |  |
| Supprimer une école | Administrateur | Supprimer une ancienne école lors de la rentrée afin d’avoir de l’espace | La perte de certaines données importantes |

# Spécifications fonctionnelles

## Carte de navigation

## Détails des cas d’utilisations

L'acteur principal actionne un fichier .exe en cliquant dessus.

A tout moment l'application peut être arrêtée par le système grâce à un bouton "arrêt" en haut à droite de l'application.

A chaque action terminé (ex: création d'une école), le système rouvre le menu proposer au lancement de l'application.

L'application s'ouvre avec un menu qui propose à l'acteur principal plusieurs options :

- créer une école

- Créer une classe dans une école existante.

- Ajouter un professeur à une classe existante.

- Ajouter un élève à une classe.

- Supprimer une école, un professeur, un élève ou une classe.

- Modifier les informations d’une école, un professeur, un élève ou une classe.

L'acteur principal choisi une option présente dans le menu et alors le système exécute l'option choisi.

Dans le cas de l'option créer une école, le système affiche une fenêtre qui permet d'inscrire le nom/budget/emplacement d'une école.

Dans le cas de la création d'une classe, le système affiche une fenêtre qui permet d'inscrire le nom/taille maximale/emplacement d'une classe ainsi que l'école à laquelle appartient la classe.

Dans le cas de l'ajout d'un professeur à une classe, le système affiche une fenêtre qui permet d'inscrire le nom/matières d'un professeur ainsi que la classe à laquelle il appartient, Le système ajoute automatiquement le professeur à une école en fonction de la classe à laquelle il est assigné.

Dans le cas de l'ajout d'un élève à une classe, le système affiche une fenêtre qui permet d'inscrire le nom/note d'un élève ainsi que la classe à laquelle il appartient, Le système ajoute automatiquement l'élève à une école en fonction de la classe à laquelle il est assigné.

Dans le cas d'une suppression d'une école, le système ouvre une fenêtre qui permet de choisir l'école à supprimer. En cas de suppression d'une école, le système supprime toutes les classes professeurs et élèves appartenant à l'école.

# Spécifications non fonctionnelles

## Contraintes de conception et d’implémentation

**Convention**

Les propriétés commenceront par un \_.

L'application sera écrite en Pascal Case (Chaque lettre de chaque mot commence par une majuscule).

Les tests unitaires doivent finir par le mot "Test".

Un problème grave lancera une exception.

Les conditions lançant une exception n'auront pas de accolade {}.

Le/Les auteur(s) d'un code doivent préciser en commentaire qu'ils en sont les créateur.

Une class par fichier.

Les propriétés auront par défaut une visibilité "private".

Ne pas utiliser le mot clé "this" pour faire référence à une propriété.

L'indentation sera effectuée avec le style Allman.

Un élément passé en argument à une fonction faisant référence à une propriété doit avoir le même nom que la propriété et être en minuscule.

**Contraintes**

- Le langage de programmation utilisé est le c#.

- Les outils utilisés sont Visual studio avec Nunit pour les tests unitaire.

- Un fichier sera utilisé pour stocker les informations relatives à l'application.

- Une documentation sera fournie afin d'expliquer le fonctionnement du programme.

- Les conventions sont indiquées dans le document Conventions.txt

## Documentation utilisateur

Une documentation sera fournie aux utilisateurs afin d'expliquer le fonctionnement de l'application.