# Dédicace

​ Nous dédions ce travail à :  
Nos mères, sources de tendresse et d’amours pour leurs soutiens tout le long de notre vie scolaire.  
Nos pères, qui nous ont toujours soutenus et qui ont fait tout possible pour nous aider.  
Nos frères et nos sœurs, que nous aimons beaucoup.  
Notre grande famille.  
Nos cher ami (e) s, et enseignants.  
Tous qu’ont collaboré de près ou de loin à l’élaboration de ce travail.  
  
Que dieu leur accorde santé et prospérité.

# Remerciement

# Résumé

# Introduction

# Problématique

## Besoins :

En tant qu’étudiants à l’ISCAE durant 3 ans on a vu que le site web de l’institut ne répond pas à certains besoins :

### Interaction :

Le site web doit offre un espace de l’interaction entre ses utilisateurs par exemple entre les professeurs et ses étudiants, entre les étudiants eux même …etc.

### Sécurité :

D’un point de vue technique, la sécurité de l’information est quasi-indispensable pour le bon fonctionnement d'un système d’information, le site web de l’institut doit effectivement être en mesure de prévenir les vulnérabilités communes des applications web.

### Etre responsive :

Le Responsive Web design est une approche de conception web qui vise à l'élaboration de sites offrant une expérience de lecture et de navigation optimales pour l'utilisateur quelle que soit sa gamme d'appareil (téléphones mobiles, tablettes, liseuses, moniteurs d'ordinateur de bureau).

## Solution :

La solution qu’on a proposée basée sur le platform .NET, répond aux besoins cités précédemment et offre des avantages supplémentaires :

### Interaction :

La solution offre un environnement d’échange d’information et de collaboration entre les utilisateurs, les étudiants peuvent ajouter des documents, discuter entre eux. Les professeurs peuvent ajouter des documents, envoyer des messages aux étudiants. L’administration capable d’ajouter des avis, des résultats, la liste des étudiants, activer et désactiver des comptes.

### Sécurité :

Tout au long du développement de la solution on a considéré les recommandations de la communauté de sécurité des applications web. Le site OWASP offre une liste des vulnérabilités des applications web chaque année. Cette considération va être le sujet de la partie sécurité (page xxx).

### Etre responsive :

L’utilisation du Framework Bootstrap, de la bibliothèque JQuery et d’autres nous permet de réaliser une interface d’utilisation responsive.

### Optimisation des ressources :

La programmation générique, et l’utilisation de quelque patron de conception (Design Pattern) nous servi à optimiser les ressources par la diminution de nombre des codes pour réaliser des taches similaires en plusieurs classes, et le nombre des instances dans la RAM.

### Maintenabilité :

La maintenance des logiciels est divisée en trois types :

#### Maintenance corrective :

Elle consiste à corriger les défauts de fonctionnement ou les non-conformités d'un logiciel.

#### Maintenance adaptive :

Sans changer la fonctionnalité du logiciel, elle consiste à adapter l'application afin que celle-ci continue de fonctionner sur des versions plus récentes des logiciels de base, voire à faire migrer l'application sur de nouveaux logiciels de base (un logiciel de base étant un logiciel requis pour l'exécution d'une application ; exemples : système d'exploitation, système de gestion de base de données).

#### Maintenance évolutive :

Cela consiste à faire évoluer l'application en l'enrichissant de fonctions ou de modules supplémentaires, ou en remplaçant une fonction existante par une autre, voire en proposant une approche différente.

Notre solution suivre les principes S.O.L.I.D qui vont être discuter plus tard. Elle est divisée en plusieurs couches chaque couche est dédié pour des taches spécifiques ce qui facilite l’accès en cas d’erreurs aux codes responsables et les corriger. Elle permet aussi l’adaptation et l’évolution des fonctionnalités.

# Développement

## Les principes S.O.L.I.D

## Couche D.A.L

### Entity Framework

### Log4Net

### Repository Pattern

## Couche B.L.L

### Contraintes

### Complexité

## Couche Web

### Web APIs

### NInject

### Design

### MVC

# Sécurité

# Conclusion

# Annexe

# Biographie