

# Projet Strawpoll

Rapport de conception

Université Évry-Val-d'Essonne

MIAGE en alternance

LA MISE EN PLACE D'UNE APPLICATION WEB STRAWPOLL  
POUR LA CREATION DU SONDEMENT

Réalisé par :  
MESLAHA Sara  
MILI Raef  
FORTIN Kevin

Encadré par :  
M<sup>me</sup>. BENZAKI  
M<sup>r</sup>. JULIEN



# Information sur le document

---

## Information pour identification :

Références du document :

Version du document : 2.0

Date du document : 23/11/2017

Auteurs : MESLAHA Sara  
MILI Raef  
FORTIN Kevin

## Éléments de vérification :

Validé par

: Mme. BENZAKI

Mr. JULIEN

Validé le

: 23/11/2017

Soumis le

: 23/11/2017

Type de diffusion

: Document électronique (.DOCX)

Confidentialité

: Standard / Étudiants

UFR Paris Saclay de l'université Paris Descartes

# Tables des matières

---

Information sur le document	3
Introduction générale	6
Un sondage : qu'est-ce que c'est ?	6
Concept de base	6
Contexte	6
Problématique ?	7
Existant (strawpoll) ?	7
Objectif	7
Fonctionnalités	7
Chapitre I :	8
Cahier des charges	9
Structure des pages	10
Constitution des pages	10
Header	10
Body	10
Footer	10
Utilisateurs : des besoins différent	10
Menus et vues de l'application	10
Page d'inscription	10
Page d'authentification	10
Page de profil	11
Page d'accueil / page de création du sondage	11
Page de sondage	12
Page des résultats	12
Page de gestion des utilisateurs	13
Page de gestion des sondages	13
Contraintes	14
Présentation des différentes contraintes	14
Référentiel des exigences	14
Plan de mise en œuvre	15
Démarche projet	16
Cycle du projet	17
Planning du projet	18

<b>Chapitre II :</b>	<b>19</b>
Conception UML	20
Analyse et conception	21
Architecture fonctionnelle	22
Diagramme de cas d'utilisation	25
Diagramme de classe	30
Diagramme de séquence	31
Diagramme d'état transition	36
Diagramme de package	37
Architecture applicative	33
Version WEB OU Client/SERVEUR :	38
 <b>Chapitre III :</b>	 40
Implémentation	41
Base de données	42
Serveur	43
Développement	43
Nos maquettes	44
Problèmes rencontrés	46
Nos figures	47

# Introduction générale

## Un sondage : qu'est-ce que c'est ?

Un sondage est une interrogation exprimée à laquelle on demande à un panel de personnes de répondre. On peut ensuite déterminer, à partir des résultats du sondage, les courants de pensée majoritaires au sein de la population donnée (par exemple, on peut qualifier une enquête d'opinions ou un référendum de sondage).

Sur internet, le principe reste inchangé et il existe de nombreux sites permettant à quiconque le souhaite de créer une enquête, un questionnaire ou bien un sondage et ce, toujours dans le même but : recueillir suffisamment d'informations pour en tirer des conclusions sur ce que les gens pensent.

## Cahier des charges – expression des besoins

Notre cahier des charges a pour but de transcrire les besoins recueillis auprès de la MOA pour les fournir à la MOE, ceci dans le but de mener à bien ce projet. Il se caractérise donc par une présentation et une définition précise de ce qui est attendu.

Ainsi, il fixera les différentes fonctions de base, les contraintes, les impératifs ainsi que son organisation et tout ce que cela implique. Dans ce document, nous vous présenterons toutes les étapes nécessaires à la réalisation de ce projet.

## Contexte

La réalisation d'un projet est une phase importante de la première année de Master MIAGE. Cette année, plusieurs projets nous ont été proposés, mais c'est le sujet dit « Strawpoll » (en référence au célèbre site internet) proposé par M. Hairapian qui a retenu notre attention.

Nous présenterons d'abord notre projet plus en détail : contexte, objectifs et exigences de l'application (23 novembre 2017).

## Problématique ?

Actuellement, il est difficile pour les élèves d'une classe de recueillir l'avis de tous leurs camarades sans que cela ne soit trop fastidieux, un site dédié à cela permettrait de gagner du temps en facilitant la démarche.

## Existant (Strawpoll) ?

L'application web « strawpoll » offre la possibilité de créer des sondages en ligne rapidement et facilement, il donne la possibilité de les partager et de représenter les résultats sous forme graphique.

Il permet aussi de commenter les résultats et empêche les participants de voter plusieurs fois si tel est le souhait du créateur du sondage.

## Objectif

L'application PollHub permettra à quiconque le souhaite de créer un sondage en ligne facilement, sans obligation d'inscription. Elle permettra aussi aux personnes le souhaitant de s'exprimer par l'intermédiaire de commentaires et fournira un lien de façon à partager le sondage sur les principaux réseaux sociaux.

Nous insisterons sur l'intuitivité de l'application, nous aimerais faire en sorte qu'elle soit accessible au plus grand nombre sans que quiconque n'ait besoin d'aide pour cela.

## Fonctionnalités

Après avoir recueilli l'avis de diverses personnes dans notre entourage, nous avons pu cibler les différentes fonctionnalités essentielles à notre application.

Ainsi, notre application proposera les fonctionnalités suivantes :

- Création de sondage
- Gestion administrative
- Inscription
- Partage de sondage
- Consultation des résultats et représentation graphique

# Chapitre I : Cahier des charges



Ce chapitre énonce comment les différentes fonctionnalités choisies seront intégrées au sein de l'application. Il détaille ainsi le contenu des pages, les contraintes amenées à se poser ainsi que les exigences ayant fixer un cadre à notre projet.

''

# Structure des pages

---

## Constitution des pages :

Chacune des pages possède la même structure : header, body et footer. On retrouve notamment des éléments communs à chacune des pages dans le header et le footer.

### Header :

Situé en haut de chacune des pages du site, il est composé du logo du site redirigeant vers l'index, d'une barre de recherche, d'un bouton d'authentification, d'un raccourci vers le profil ainsi que vers les différents réseaux sociaux sur lesquels l'application est présente.

### Body :

Les éléments qui sont propres à la page sur laquelle nous sommes.

### Footer :

Le footer comporte des informations nous concernant nous ainsi que notre projet.

## Utilisateurs : des besoins différent :

Nous avons choisi de distinguer trois « types » d'utilisateurs au sein de notre application :

- Utilisateur non-enregistré (anonyme)
- Utilisateur enregistré
- Administrateur

Ces utilisateurs se distinguent principalement par les accès qu'ils ont au sein de l'application. Un utilisateur anonyme n'aura pas accès à autant de pages qu'un utilisateur enregistré, il aura donc des possibilités restreintes par rapport à ce dernier.

## Menus et vues de l'application

Lorsqu'un sondage en ligne est créé, il est rendu accessible à tout le monde (utilisateurs inscrits comme non-inscrits). Chacun peut donc décider de participer à un sondage et voir les résultats de ces derniers.

Tout ce processus se fait au travers de diverses pages que nous listerons ci-dessous :

### Page d'inscription :

Cette page permet à un utilisateur non-enregistré de s'inscrire, pour cela, il sera invité à remplir un formulaire (pseudo, adresse e-mail, mot de passe, confirmation du mot de passe ainsi que captcha à valider). Il devra ensuite valider son inscription à partir d'un mail reçu à l'adresse qu'il a renseigné.

## Page de connexion :

Une fois inscrit, un utilisateur a la possibilité de se connecter sur la page d'authentification grâce à ses identifiants préalablement définis (pseudo ou e-mail et mot de passe). Il aura aussi la possibilité de redéfinir son mot de passe si jamais il a oublié ce dernier.

## Page de profil :

Chacun des utilisateurs enregistrés aura accès à sa page de profil, cette page lui permettra de faire plusieurs choses :

- Modifier son mot de passe
- Visualiser les sondages qu'il a lui-même créé
- Supprimer ses sondages
- Modifier son sondage (ajout de temps)

## Page d'accueil / Page de création de sondage

Sur la page d'accueil, un formulaire à remplir permettra directement de créer un sondage, ce formulaire se compose d'une question ainsi que de 3 cases destinées aux réponses (par défaut, extensible).

Il dispose aussi de la possibilité d'ajouter une option de vérification par IP, d'autoriser ou non les choix multiples.

### Créer un sondage :

---

Question :	<input type="text" value="Saisissez votre question"/>
Réponse 1 :	<input type="text"/>
Réponse 2 :	<input type="text"/>
Réponse 3 :	<input type="text"/>
Réponse 4 :	<input type="text"/>

Autoriser plusieurs réponses au sondage  
 Vérification par adresse IP

Figure 1 : maquette du formulaire de création d'un sondage

## Page du sondage :

Ici l'utilisateur peut sélectionner la ou les réponses pour lesquelles il souhaite voter, s'il a déjà voté auparavant et que l'option « IP duplication » est activée alors un message d'erreur apparaîtra. Il peut aussi accéder aux résultats ou encore récupérer le lien du sondage afin de le partager.

### Comment trouvez-vous notre rapport ?

Sélectionnez une réponse:

- Bof
- Génial
- Bien
- Nul

**Vote**

[Voir les résultats](#)

Figure 2 : maquette du vote pour un sondage

## Page des résultats :

La page de résultats énumèrera les différentes réponses possibles pour le sondage avec un pourcentage représentant la part des sondés ayant voté pour ces dernières, ces mêmes réponses seront aussi représentées sous la forme d'un graphique qui s'actualise en temps réel. La possibilité de récupérer un lien afin de partager le sondage se réitère sur cette page. L'utilisateur peut aussi, s'il le souhaite, retourner sur la page de vote depuis cette page s'il n'a pas déjà voté.

### Comment trouvez-vous notre rapport ?

- |         |        |
|---------|--------|
| 37.04 % | Génial |
| 32.41 % | Bien   |
| 21.30 % | Bof    |
| 9.26 %  | Nul    |

**108 votes total**



[Retour](#)

Figure 3 : maquette des résultats d'un sondage

## Page de gestion des utilisateurs :

Seuls les administrateurs peuvent accéder à cette page : elle permet de consulter et de gérer l'ensemble des utilisateurs enregistrés (modification d'informations, suppression ...).

Elle peut servir dans le cas où les administrateurs seraient amenés à faire le « support » pour un utilisateur (par exemple pour un changement de pseudonyme).

## Page de gestion des sondages :

Cette page n'est accessible qu'aux administrateurs, tout comme pour la page de gestion des utilisateurs, elle permet de consulter et de gérer l'ensemble des sondages enregistrés

Elle peut servir dans le cas où un sondage serait jugé inapproprié, un administrateur supprimerait donc le dit sondage).

# Contraintes

---

## Présentation des différentes contraintes

### Contraintes matérielles :

Utilisation du langage C#

Hébergement du projet sur GitHub

### Contraintes du délai :

La durée de notre projet dépend de plusieurs dates préalablement définies, ainsi, on distingue plusieurs « étapes » au sein de ce dernier :

Ces différentes étapes font office de « dates buttoirs » et nous avons appris à nous organiser autour de ces dernières pour mener à bien notre projet.

## Référentiel des exigences

### Qualité:

Nous nous sommes fixé certaines exigences en terme de qualité :

1 – Maintenance (web) : l'application est réalisée à partir du framework .NET afin de faciliter sa maintenance. Nous avons choisi de la coupler à l'architecture MVC toujours dans cette même dynamique – nous reviendrons sur ce choix dans le volet architecture applicative.

2 – Portabilité (web) : toutes les bibliothèques nécessaires au démarrage de l'application sont directement disponibles dans le path relatif à l'application (pas de dépendance externe).

3 – Extensibilité (web) : L'ajout de modules à l'application est facilité.

4 – Facilité d'emploi : Cette application est intuitive et même une personne non-introducte est capable de l'utiliser sans assistance – cette affirmation résulte d'une enquête réalisée auprès de notre entourage.

# Plan de mise en œuvre

---

## Démarche projet :

Le processus de développement de notre application comporte plusieurs étapes :

### Ces étapes sont :

C'est le résultat de l'enquête auprès de nos amis : nous recueillons leur avis et spécifions ce qui nous convient. Une fois le besoin exprimé, nous déterminons les ressources matérielles et logicielles nécessaires à l'établissement de ce dernier pour réaliser notre application.  
De cette étape résulte les schémas fonctionnels présents dans le chapitre N°1, ils nous servent à nous guider lors de la phase de conception.

### L'étude préalable (réalisation du cahier des charges) :

Cette étape sert à identifier le problème, et à élaborer une version de base du cahier des charges, qui doit inclure :

- la décision de faisabilité : mon application est-elle faisable ?
- le plan général du projet : comment vais-je structurer mon projet ?
- une estimation approchée du coût et des délais de réalisation : comment vais-je planifier mon projet ? Quels sont les coûts de ce projet ?

### La spécification

C'est le résultat de l'enquête auprès de nos amis : nous recueillons leur avis et spécifions ce qui nous convient. Une fois le besoin exprimé, nous déterminons les ressources matérielles et logicielles nécessaires à l'établissement de ce dernier pour réaliser notre application.

De cette étape résulte les schémas fonctionnels, ils nous servent à nous guider lors de la phase de conception.

## La conception

La conception de notre application se fera via sa modélisation avec le langage UML :

- Diagramme de séquence : pour les communications entre les différents acteurs (utilisateur, système)
- Diagramme de classes : pour exprimer les relations entre les différents objets persistants (notamment les tables de données qui sont représentées sous forme d'objet pour le traitement des données)
- Diagramme de cas d'utilisation : pour rassembler les actions réalisables par les différents utilisateurs
- (Diagramme de packages à rajouter)

## La codification

C'est dans cette étape que nous nous penchons sur le code : chaque algorithme ou requête est traduit en un ou plusieurs langages de programmation, l'IHM (le site web) quant à elle s'appuiera sur ce même code (notamment C#) pour répondre aux besoins des utilisateurs.

## Les tests unitaires

Les tests unitaires nous permettent de tester chacun des modules (dans notre cas des parties de code nouvellement écrites) indépendamment de façon à vérifier leur bon fonctionnement.

## Les tests systèmes (intégration de l'application au sein du système existant)

Cette étape consiste à intégrer tous les modules créés et testés précédemment et de les tester ensemble.

Lorsque nous serons assurés du fonctionnement complet du système, nous rédigerons une documentation de notre application, récapitulant les différentes étapes nécessaires pour chacune des actions possibles en fonction du type d'utilisateur.

## L'exploitation

Une fois la documentation rédigée, nous mettrons à disposition de tous notre application, de façon à avoir une situation réelle de production.

## Cycle du projet :

Pour le développement de notre application, nous avons décidé d'utiliser le modèle en cascade de façon à valider chacune de nos étapes avant de passer à la suivante.

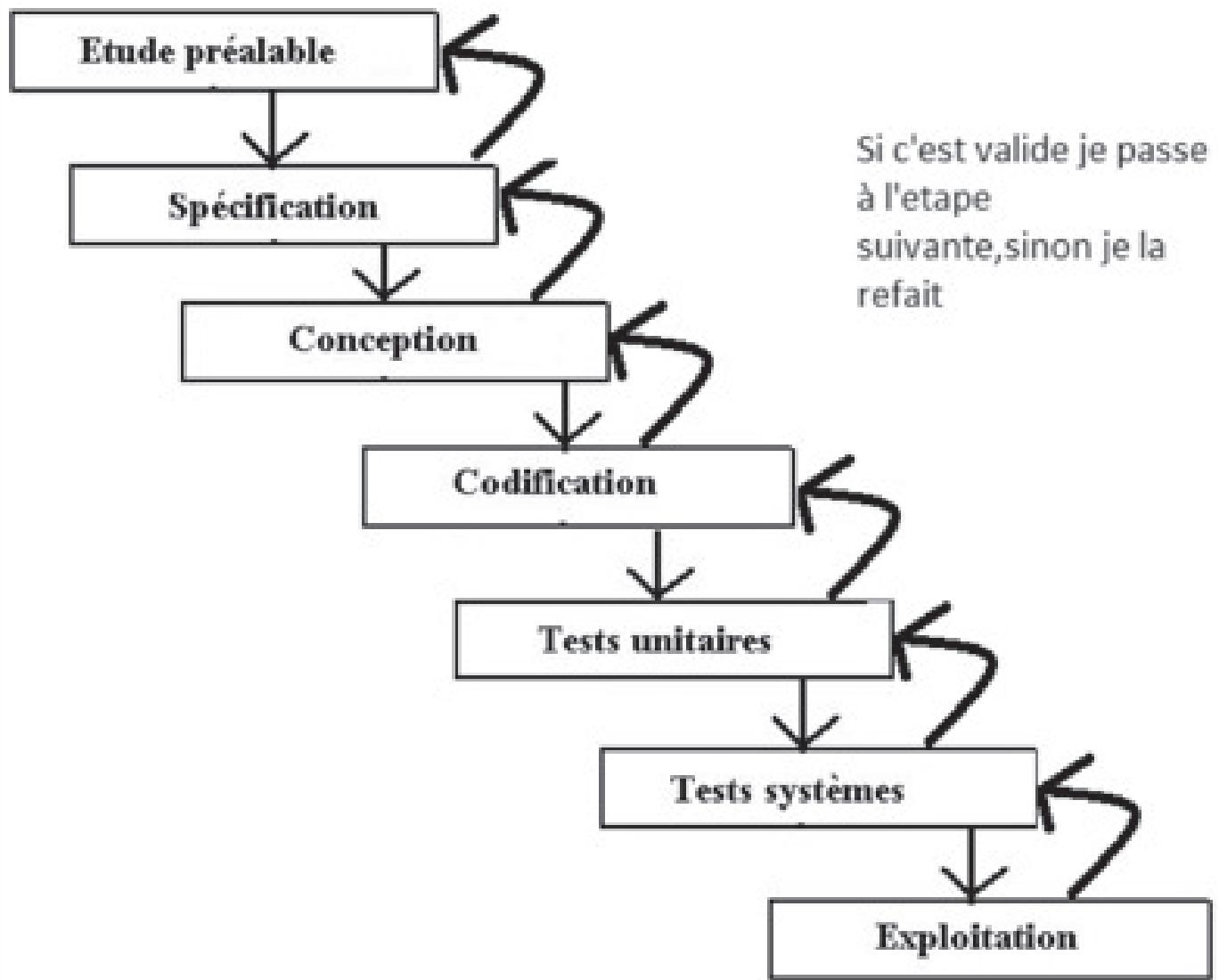


Figure 4 : modèle de conception de notre projet, notre choix s'est porté sur un modèle en cascade

L'avantage de ce modèle est de proposer une démarche de réduction des risques, en minimisant au fur et à mesure l'impact des incertitudes. L'impact d'une incertitude dans la phase de développement étant plus faible que l'impact d'une incertitude dans les phases de Conception ou de Spécifications, plus le projet avance, plus les risques diminuent.

Le modèle en cascade est adapté aux projets de durée inférieure à un an à forte composante réglementaire.

## Planning du projet :

Nous avons décidé, en amont du projet, d'estimer en temps chacune des tâches nécessaires à l'accomplissement de notre projet. Ainsi nous pouvons avoir une idée globale du temps que peut potentiellement nous prendre le projet ainsi que son déroulement.

#	①	Name	Duration	Start	Finish	Predecessors
1	豇豆	Etude préalable	3days?	11/23/2017	11/27/2017	
2		Spécification	2days?	11/28/2017	11/29/2017	1
3		Conception	90days?	11/30/2017	04/04/2018	2
4		Codification & test unitaire	60days?	04/05/2018	06/27/2018	3
5		Documentation	15days?	06/28/2018	07/18/2018	4
6		Exploitation	2days?	07/19/2018	07/20/2018	5

Figure 5 : Tâches à accomplir au sein de notre projet et estimation en temps de ces dernières

Cela nous laisse donc un certain temps avant de choisir les technologies que nous utiliserons.

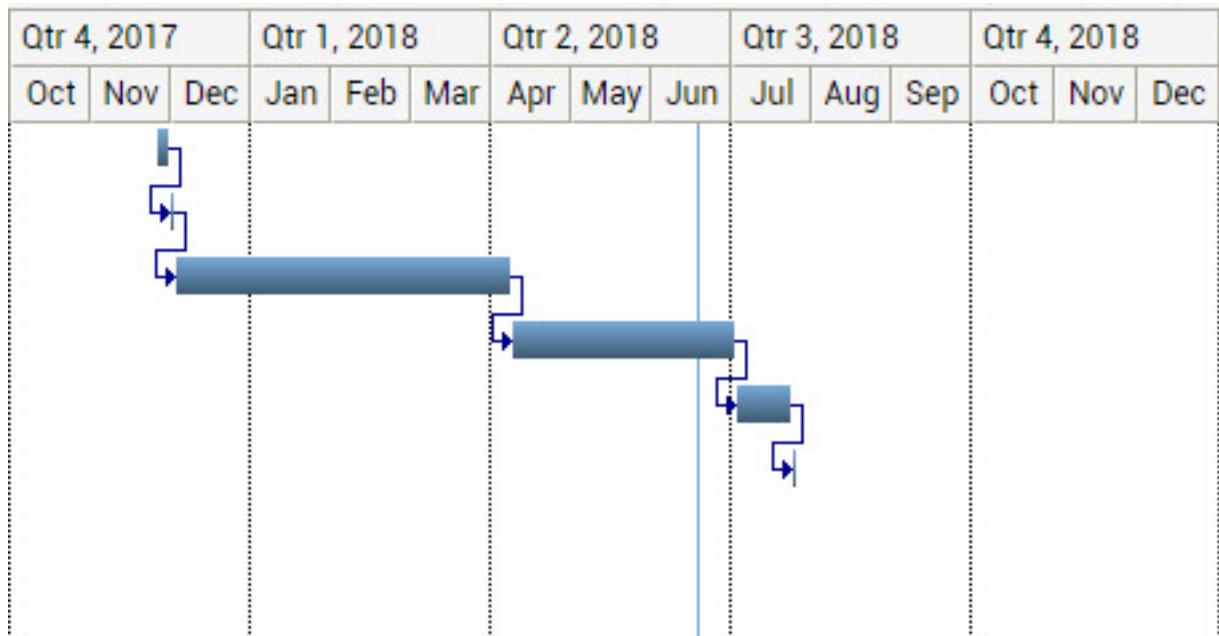


Figure 6 : Diagramme de Gantt représentant graphiquement le déroulement de notre projet en fonction du temps

# Chapitre II : Conception UML



|- - - - -  
| Ce chapitre présente les différents  
| diagrammes UML du projet. Quels  
| sont les différents use cases ?  
| Que peuvent faire les différents  
| utilisateurs au sein de l'application  
| ?  
|- - - - -

“

# Analyse et conception

---

Afin d'être sûr des fonctionnalités dont dispose notre application, nous avons choisi de la modéliser avec le langage UML.

Nous avons trois types d'utilisateur qui peuvent accéder à notre application :

## Administrateur :

Un administrateur (après son authentification) peut :

- créer, supprimer, modifier (CRUD : Create, Read, Update, Delete) les différents profils existant et leur donner des droits d'accès.
- visualiser le reporting liés aux résultats des sondages.
- créer, supprimer, modifier (CRUD : Create, Read, Update, Delete) les différents sondages existant

## Utilisateur connecté :

Un utilisateur (après son authentification) peut :

- visualiser le reporting liés au progrès des sondages.
- consulter les sondages.

## Anonyme :

Un utilisateur (non authentifié) peut :

- s'inscrire.
- créer un sondage .
- voter.

# Architecture fonctionnelle

L'architecture fonctionnelle (qui résulte notamment de l'enquête faîte auprès de nos proches) a été réalisée grâce au logiciel « Enterprise Architect » - logiciel de modélisation et de conception UML.

Comme dit précédemment, on distingue plusieurs utilisateurs au sein de notre application avec chacun des accès plus ou moins restreints.

## Utilisateur non-enregistré / Anonyme :

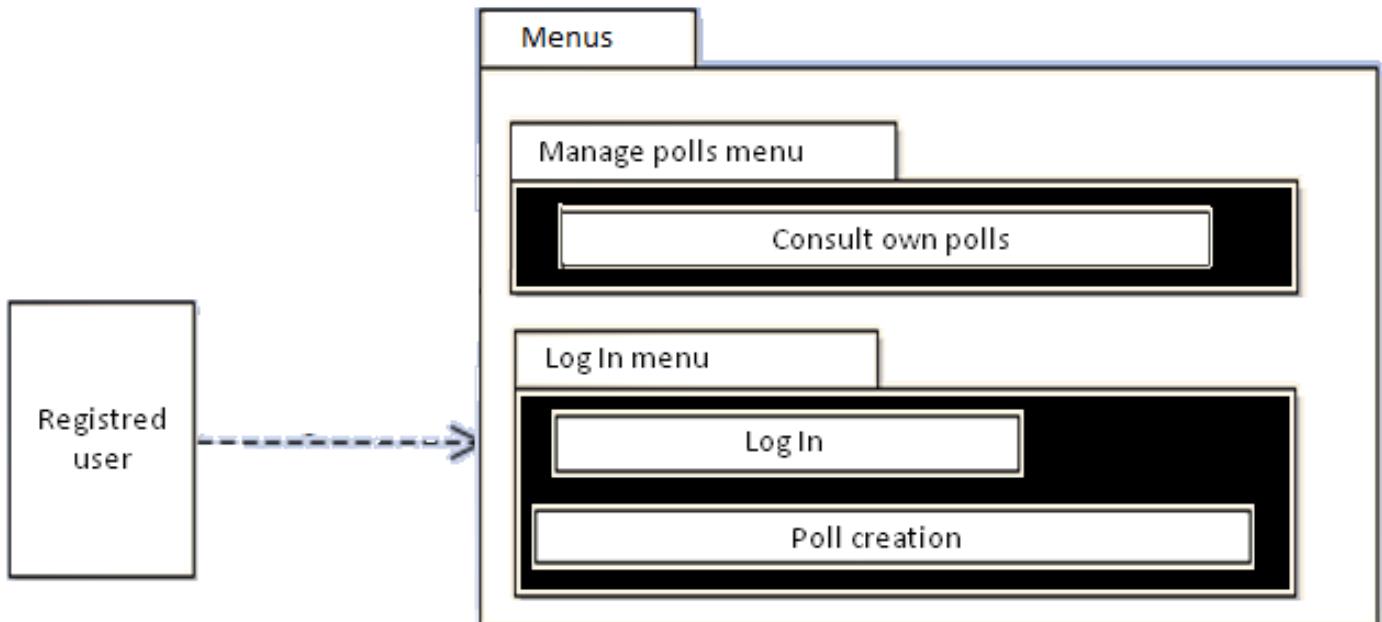


Figure 7 : architecture fonctionnelle des possibilités et raccourcis d'un utilisateur enregistré sur les différents menus auxquels il a accès

## Utilisateur enregistré :

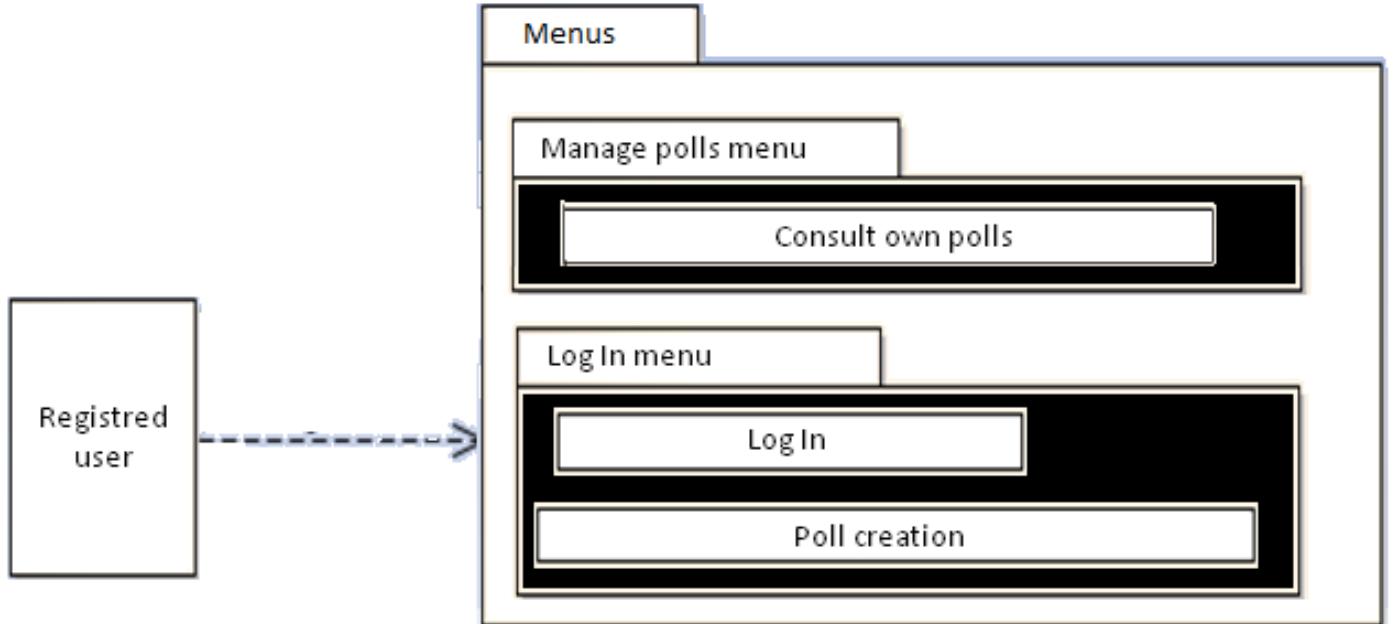


Figure 8 : architecture fonctionnelle des possibilités et raccourcis d'un utilisateur enregistré sur les différents menus auxquels il a accès

## Administrateur :

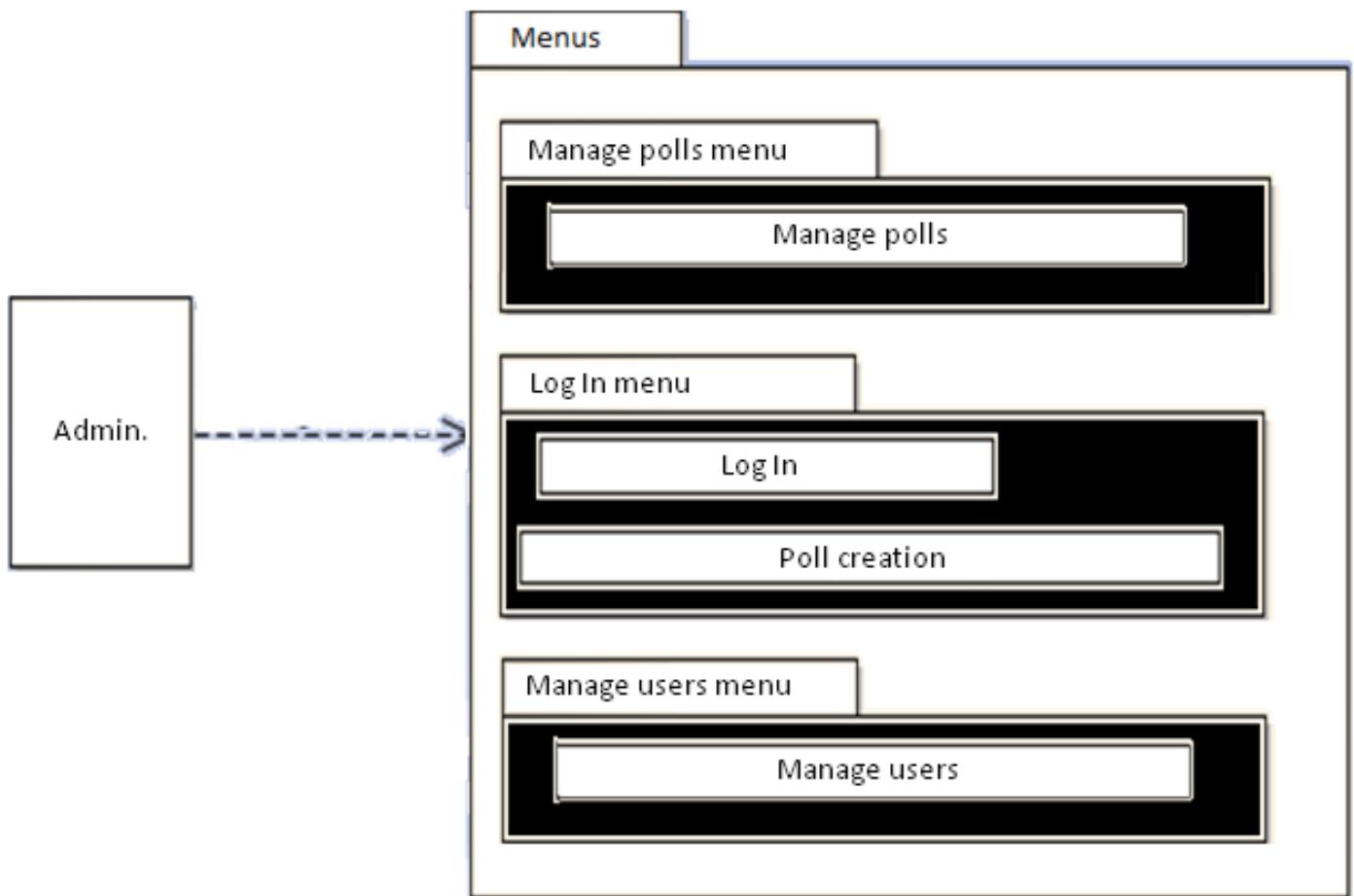


Figure 9 : architecture fonctionnelle des possibilités et raccourcis d'un administrateur sur les différents menus auxquels il a accès

## Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation nous permettent d'identifier les différents acteurs et les fonctionnalités qui leur sont attribuées, ils permettent d'avoir une meilleure vision des attentes du client. L'objectif étant de définir plus en détails les besoins de chacun des types d'utilisateurs en répondant à la question : QUI devra pouvoir faire QUOI dans l'application ? Dans notre cas, les diagrammes définiront les fonctionnalités des trois acteurs : administrateur, utilisateur connecté et utilisateur non connecté.

### Cas de création d'un compte

Action: création d'un compte

Acteur: utilisateur non inscrit

Données d'entrées: l'utilisateur clique sur « s'inscrire »

Scenario:

- la page d'inscription est ouverte
- l'utilisateur saisit son pseudo, son adresse mail, son mot de passe (2x)
- l'utilisateur valide le formulaire en cliquant sur le bouton « je m'inscris »
- l'administrateur valide l'inscription de l'utilisateur

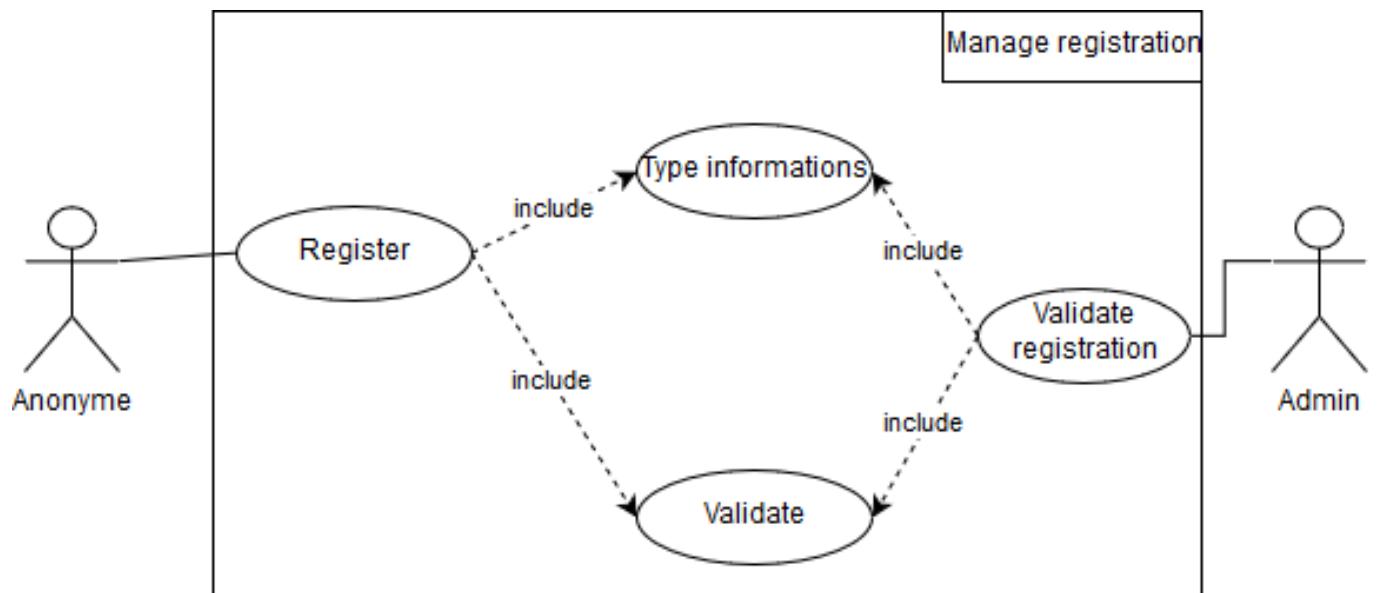


Figure 10 : Schéma représentant le scenario de gestion d'inscription

## Cas de connexion au site

Action: connexion au site  
Acteur: utilisateur non connecté  
Données d'entrées: l'utilisateur clique sur « se connecter »  
Scenario:  
- la page de connexion est ouverte  
- l'utilisateur saisit son pseudo ou son adresse mail ainsi que son mot de passe  
- l'utilisateur clique sur « connexion » pour se connecter

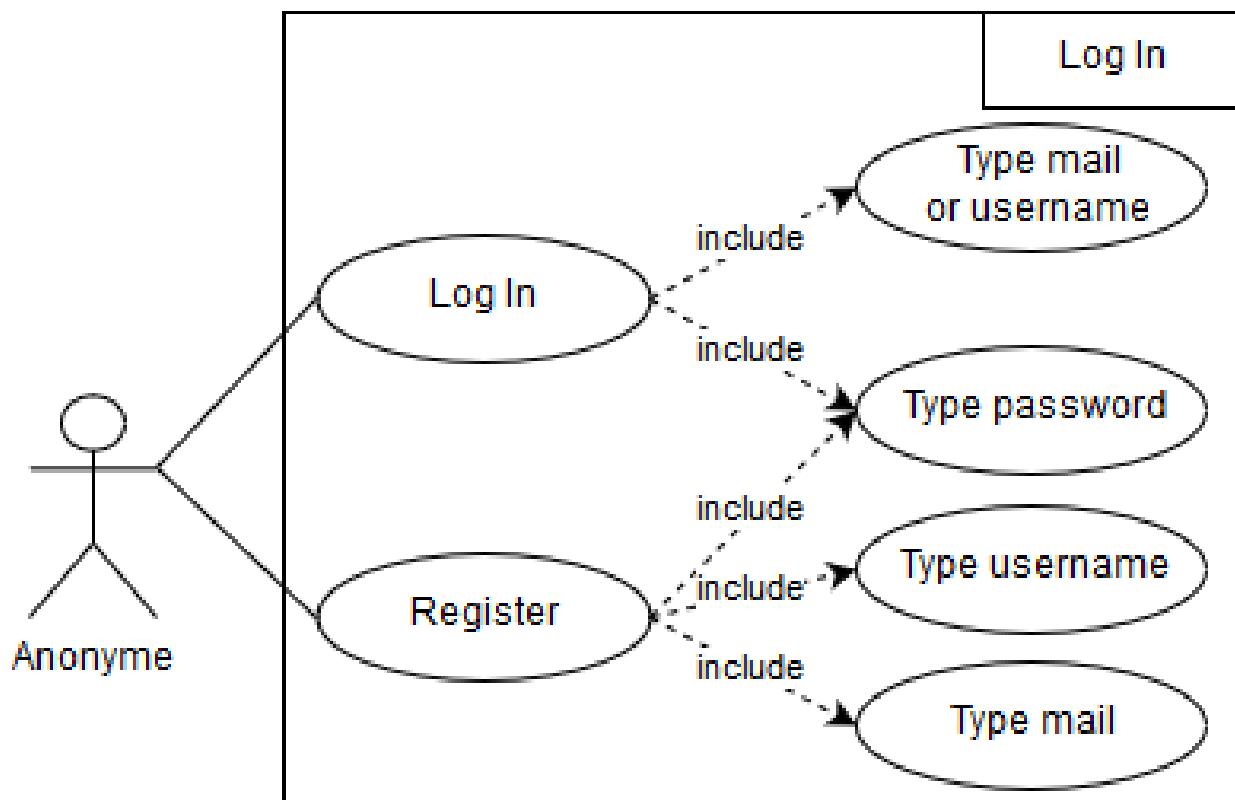


Figure 11 : Schéma représentant le scenario de gestion de connexion

## Cas de création d'un sondage

Action:

création d'un sondage

Acteur:

utilisateur connecté (ou non)

Données d'entrées:

l'utilisateur se rend sur la page d'accueil

Scénario:

- la page d'accueil permet de créer un sondage
- l'utilisateur saisit la question ainsi que les réponses souhaitées
- l'utilisateur choisit s'il souhaite autoriser ou non une même adresse IP à voter deux fois
- l'utilisateur choisit s'il autorise plusieurs réponses
- l'utilisateur peut publier le sondage
- s'il le souhaite, l'utilisateur enregistré peut supprimer son sondage

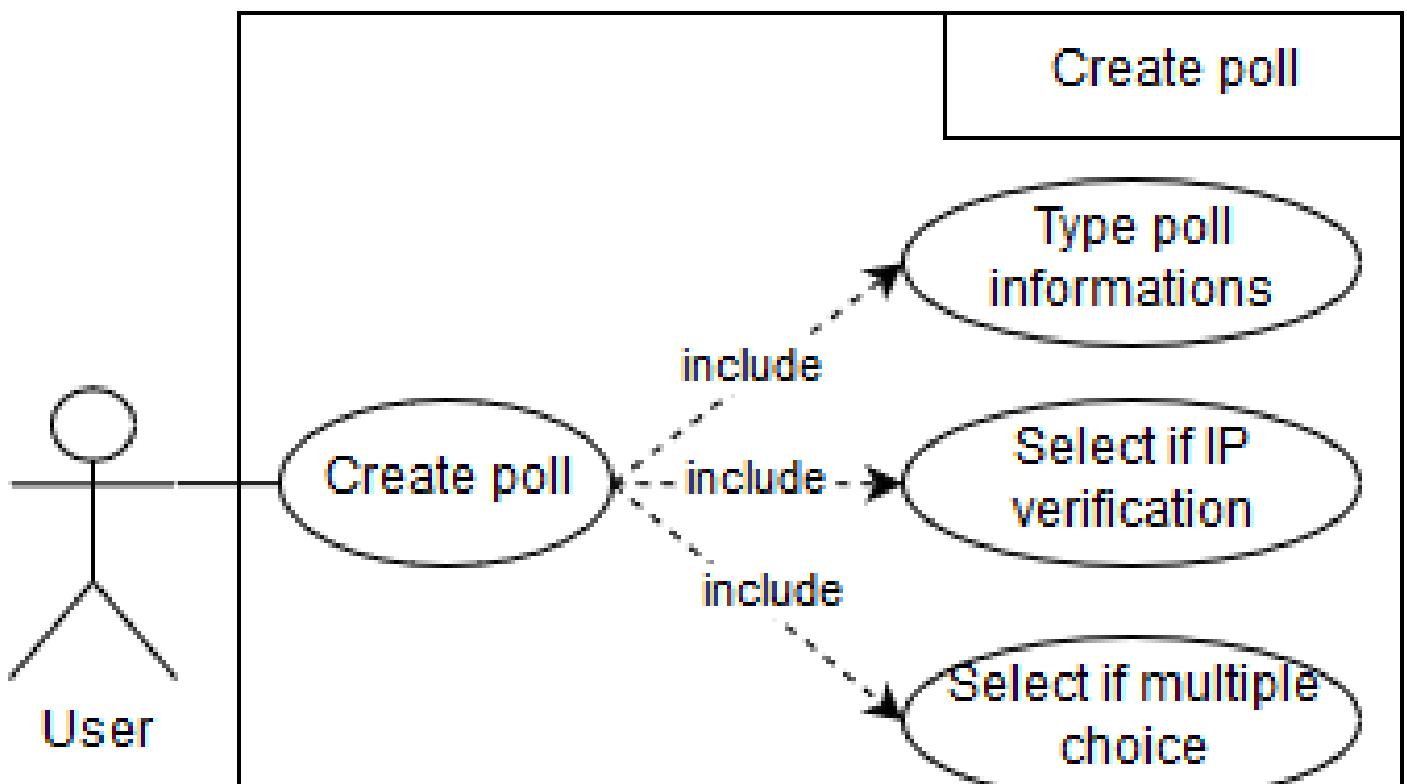


Figure 12 : Schéma représentant le scénario de création de sondage

## Cas d'interaction avec le sondage

Action: interagir avec le sondage  
Acteur: utilisateur connecté (ou non)  
Données d'entrées: l'utilisateur obtient le lien menant vers le sondage  
Scenario:  
- l'utilisateur peut se rendre sur la page de sélection des réponses du sondage  
- l'utilisateur peut voter au sondage  
- l'utilisateur peut consulter les résultats du sondage

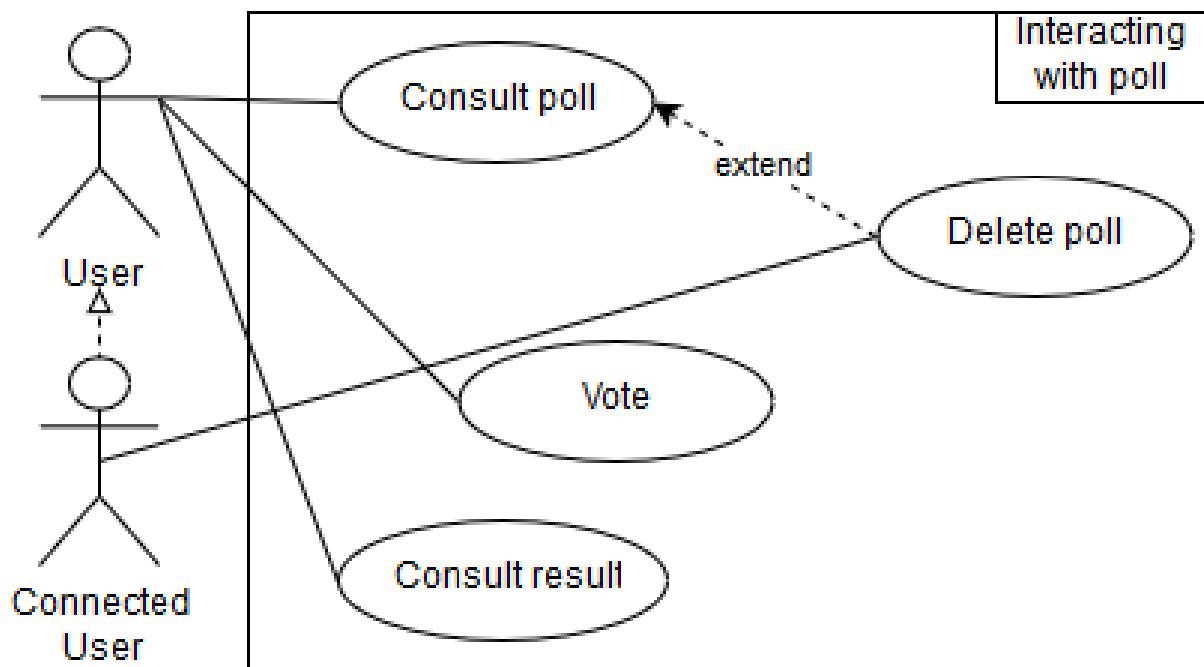


Figure 13 : Schéma représentant le scenario d'interaction sondage

## Cas d'administration

Action:

Administration

Acteur:

personne connectée en tant qu'Administrateur

Données d'entrées:

la personne clique sur « s'authentifier »

Scenario:

- la personne commence par s'authentifier en tant qu'administrateur
- l'administrateur peut valider les inscriptions
- l'administrateur peut supprimer le compte d'un utilisateur
- l'administrateur peut ajouter un sondage
- l'administrateur peut supprimer un sondage
- l'administrateur peut visualiser les résultats d'un sondage

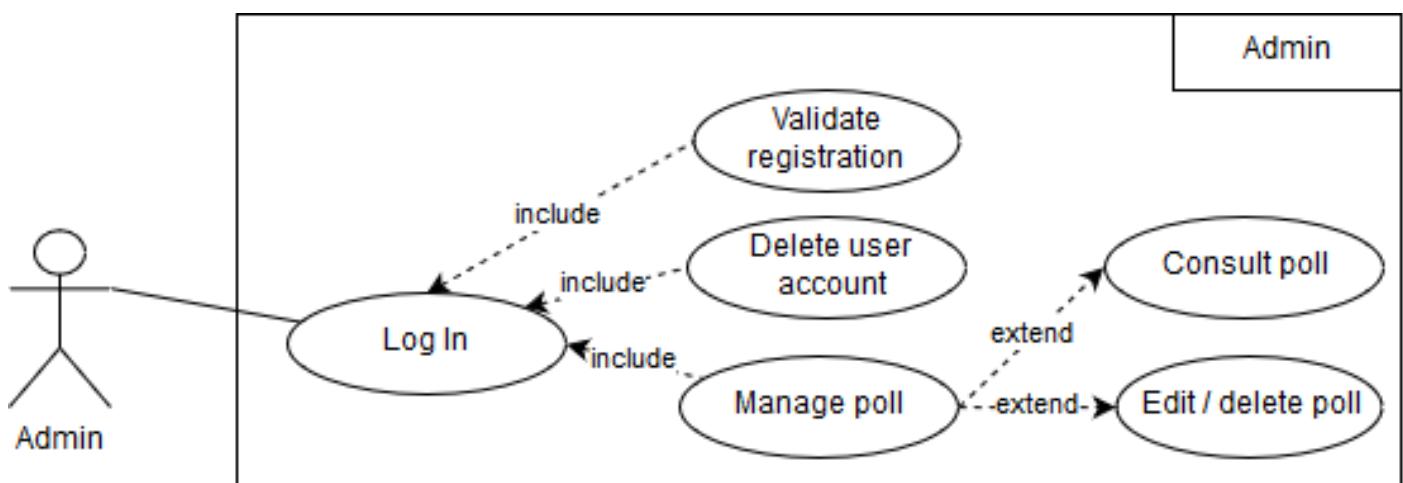


Figure 14 : Schéma représentant le scenario d'administration

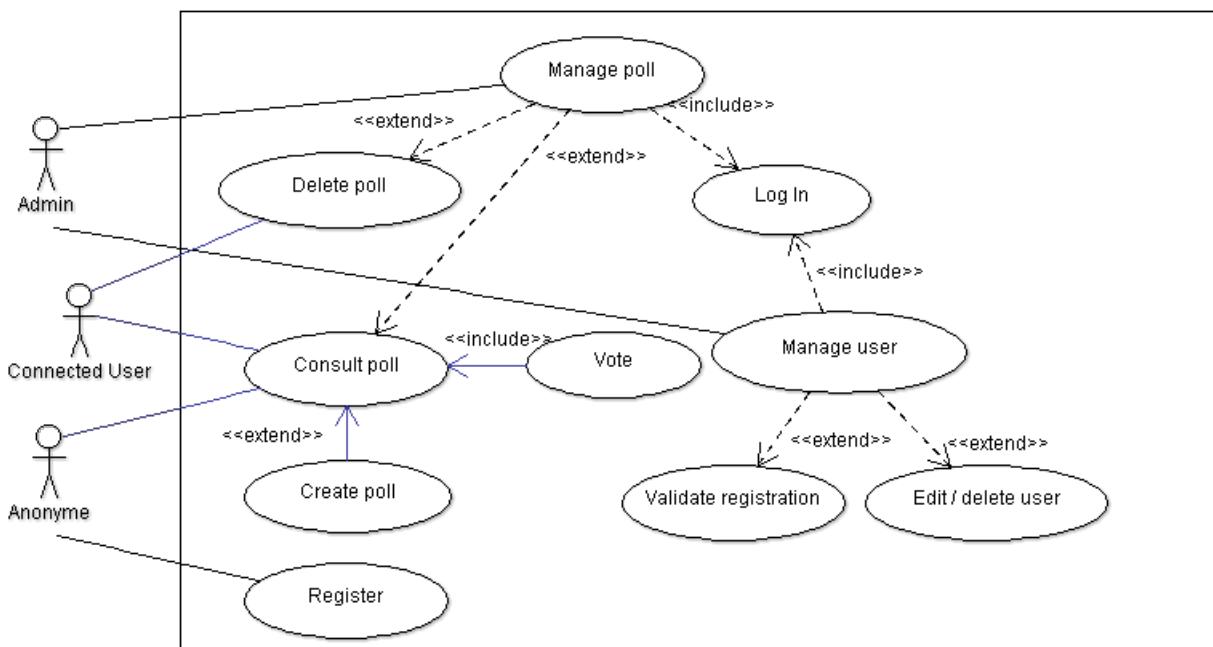


Figure 15 : Diagramme de cas d'utilisation général

## Diagramme de classe

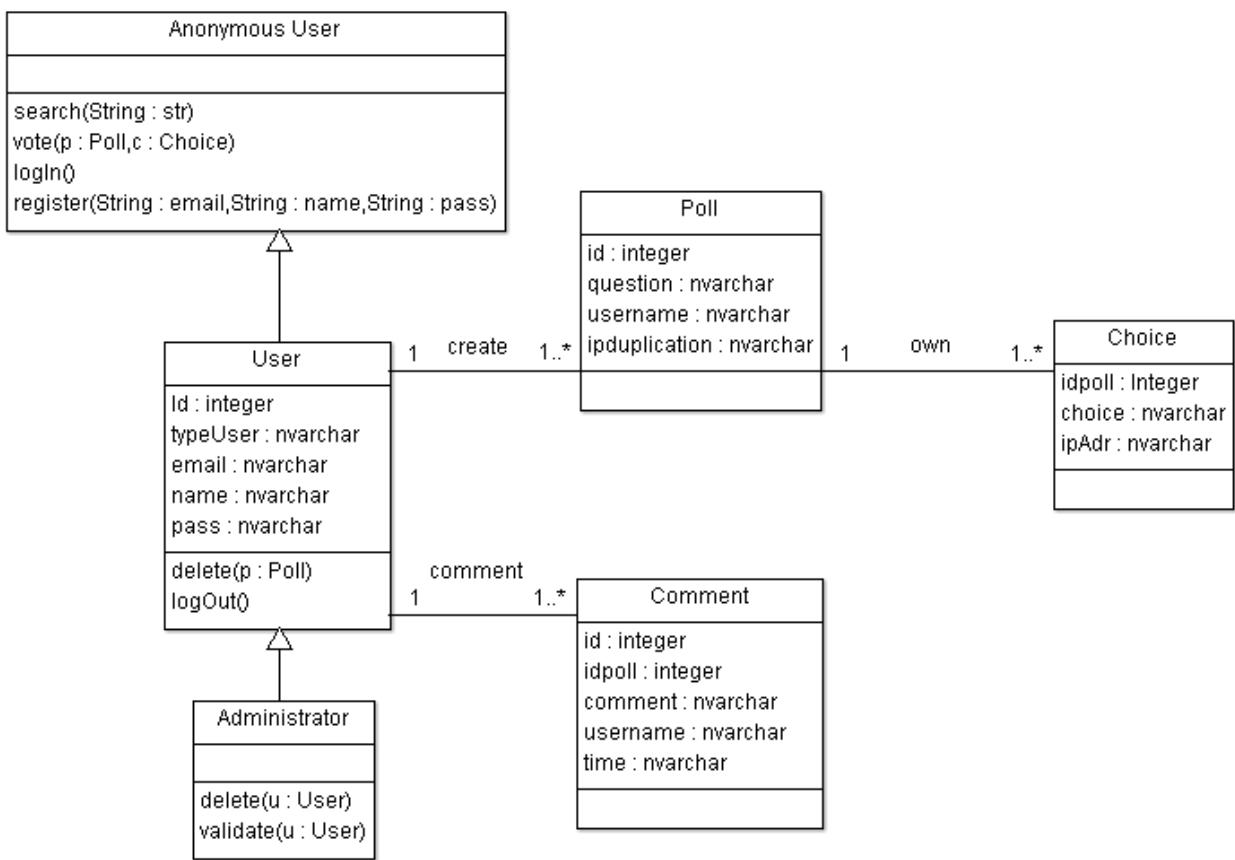


Figure 16 : Schéma définit les différentes classes de l'application « Strawpoll »

# Diagramme de séquence

## « Cycle de vie » d'un compte utilisateur

- l'utilisateur saisit son pseudo, son adresse mail, son mot de passe (2x)
- l'utilisateur valide le formulaire en cliquant sur le bouton « je m'inscris »
- l'administrateur valide l'inscription de l'utilisateur
- l'administrateur se rend sur la page « gestion des utilisateurs »
- l'administrateur supprime l'utilisateur s'il le souhaite

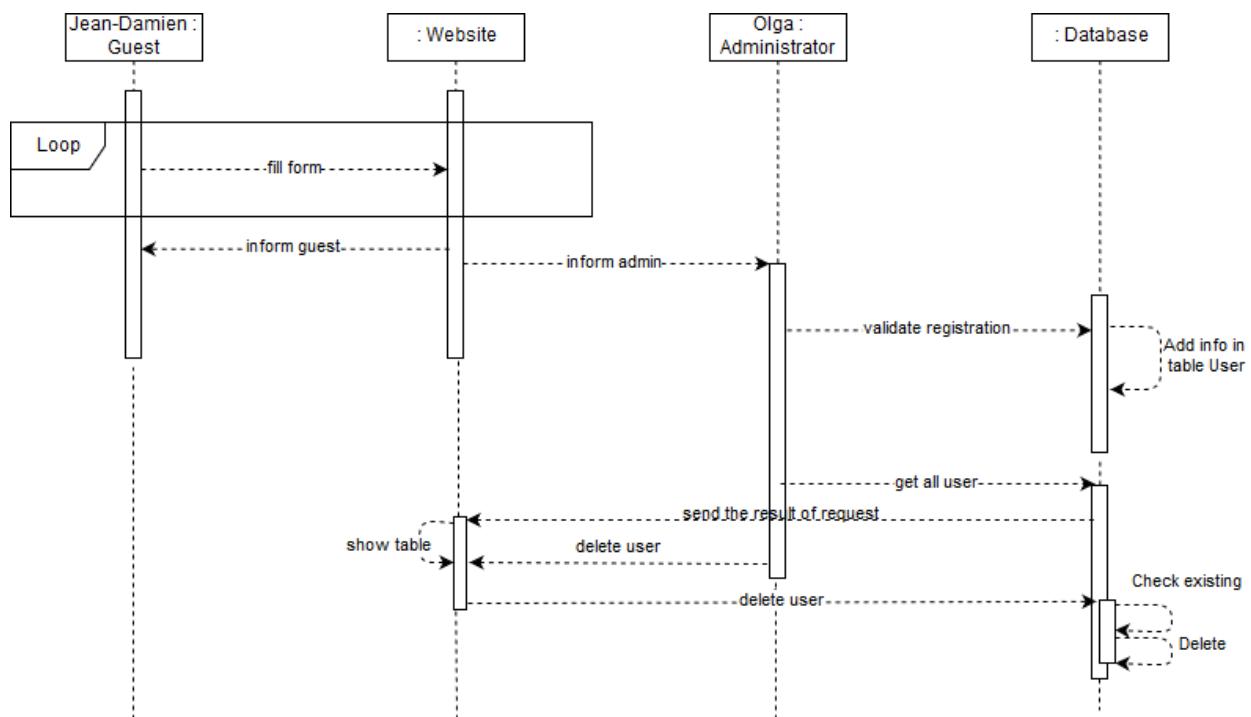


Figure 17 : diagramme de séquence pour l'inscription de l'utilisateur

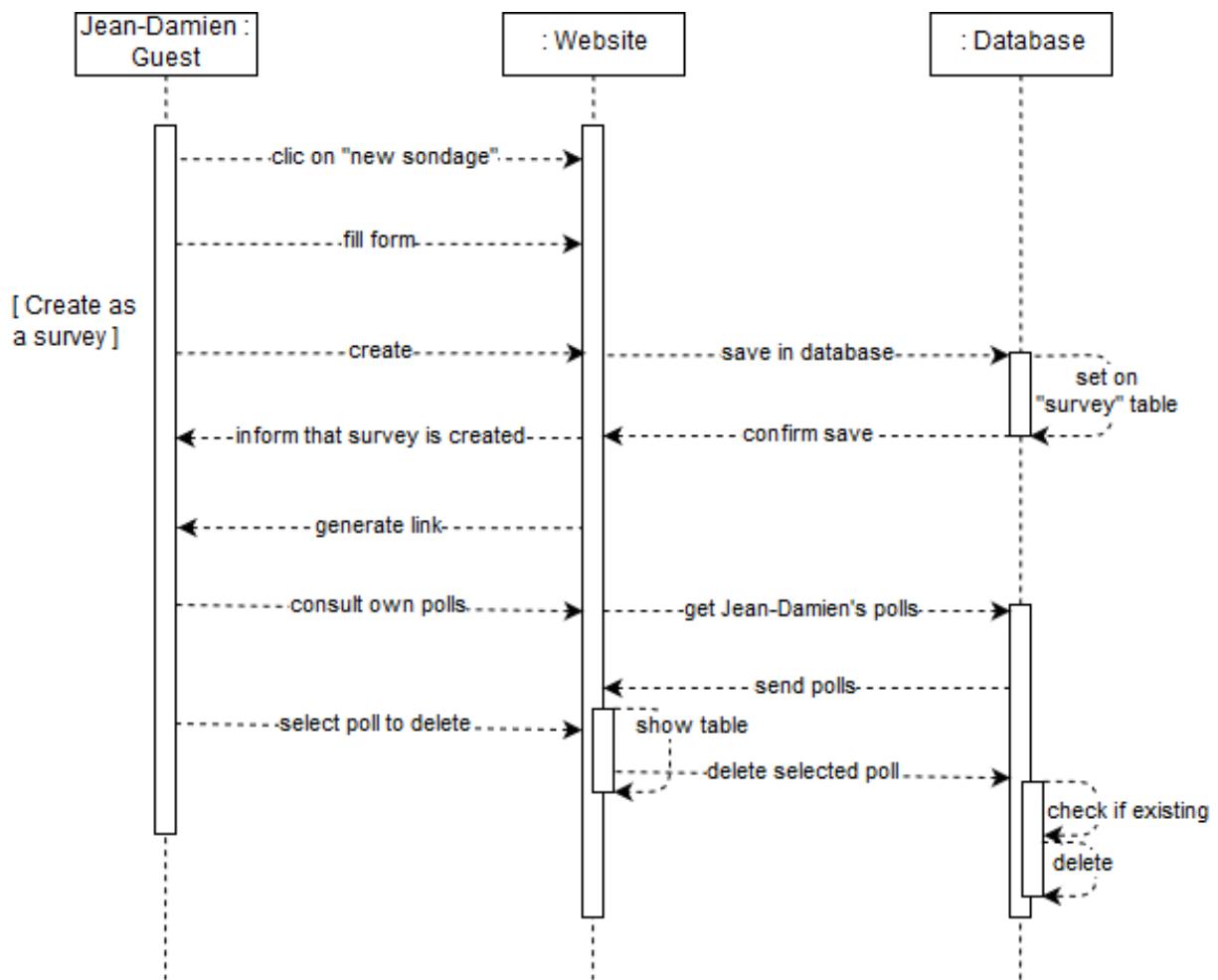


Figure 19 : diagramme de séquence du cycle de vie d'un sondage

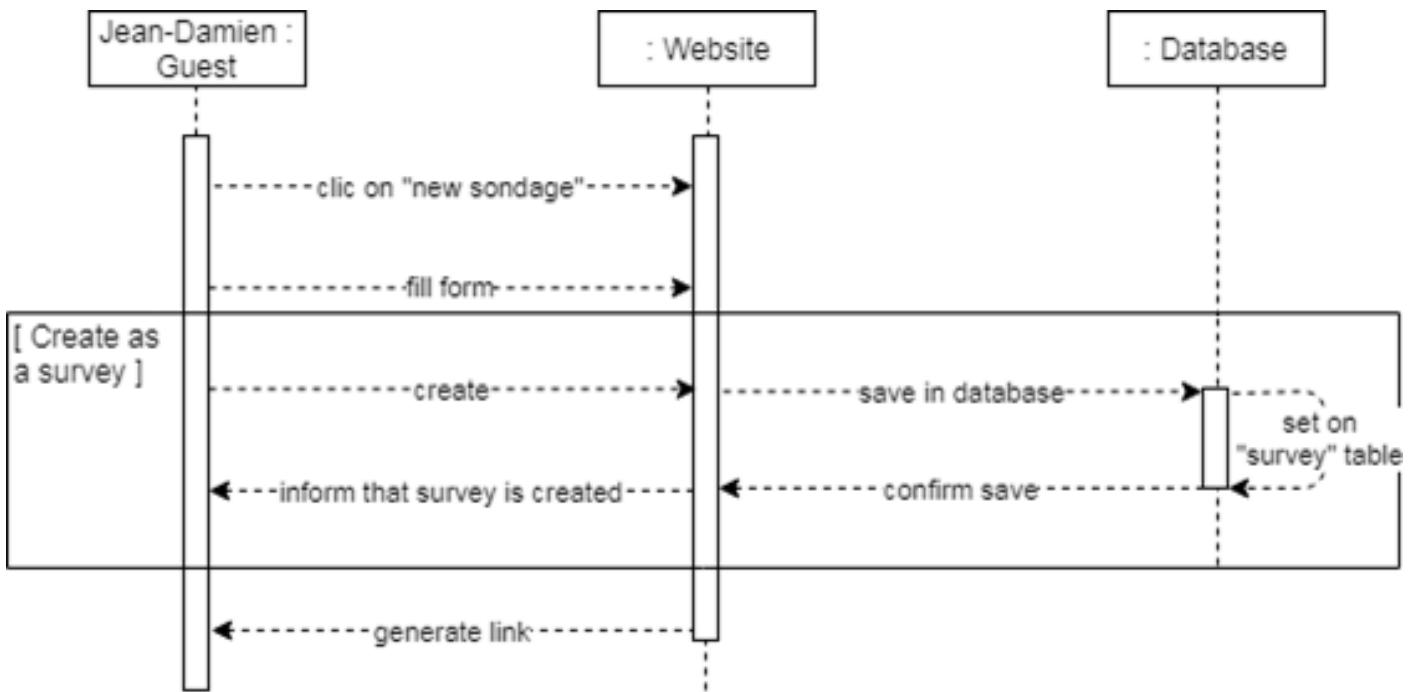


Figure 20 : diagramme de séquence de la création d'un sondage

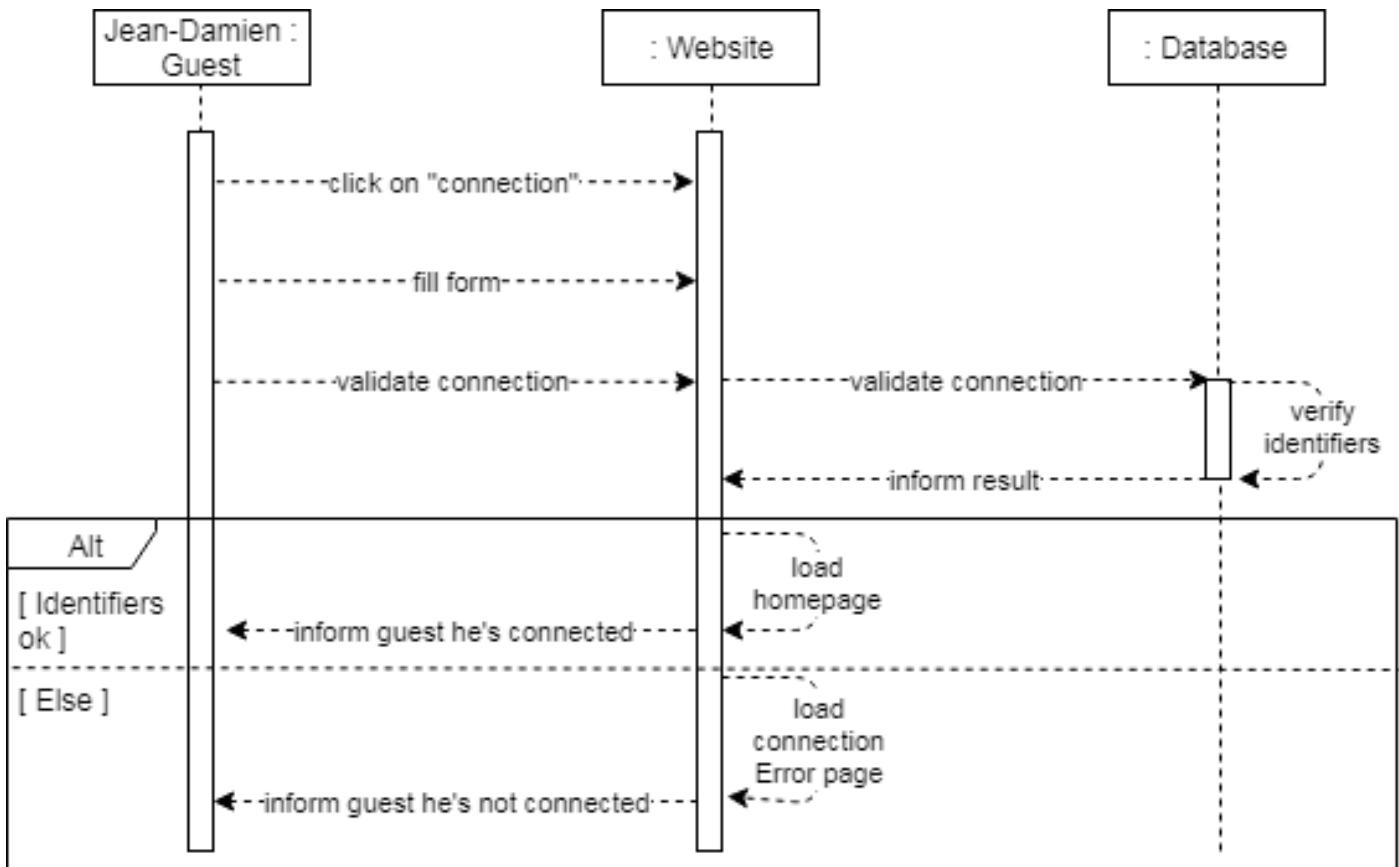


Figure 21 : diagramme de séquence de connexion d'un utilisateur

## Cas de la connexion d'un utilisateur

- Un utilisateur doit avoir un compte enregistré dans la base de données pour pouvoir se connecter
- L'utilisateur est redirigé vers la page d'accueil après s'être connecté

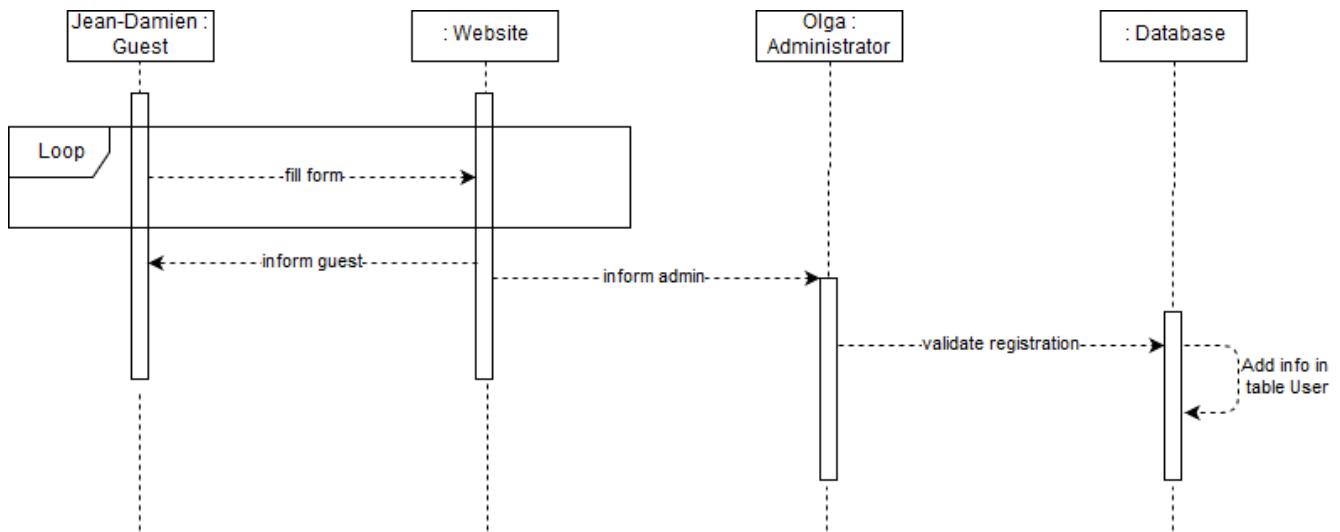


Figure 22 : diagramme de séquence pour l'inscription de l'utilisateur

## Cas de création d'un sondage

- Pour la création d' un sondage il faut remplir les champs du formulaire dédiés au sondage et confirmer pour enregistrement de ce dernier
- Après la confirmation le site génère un lien que l'utilisateur peut partager

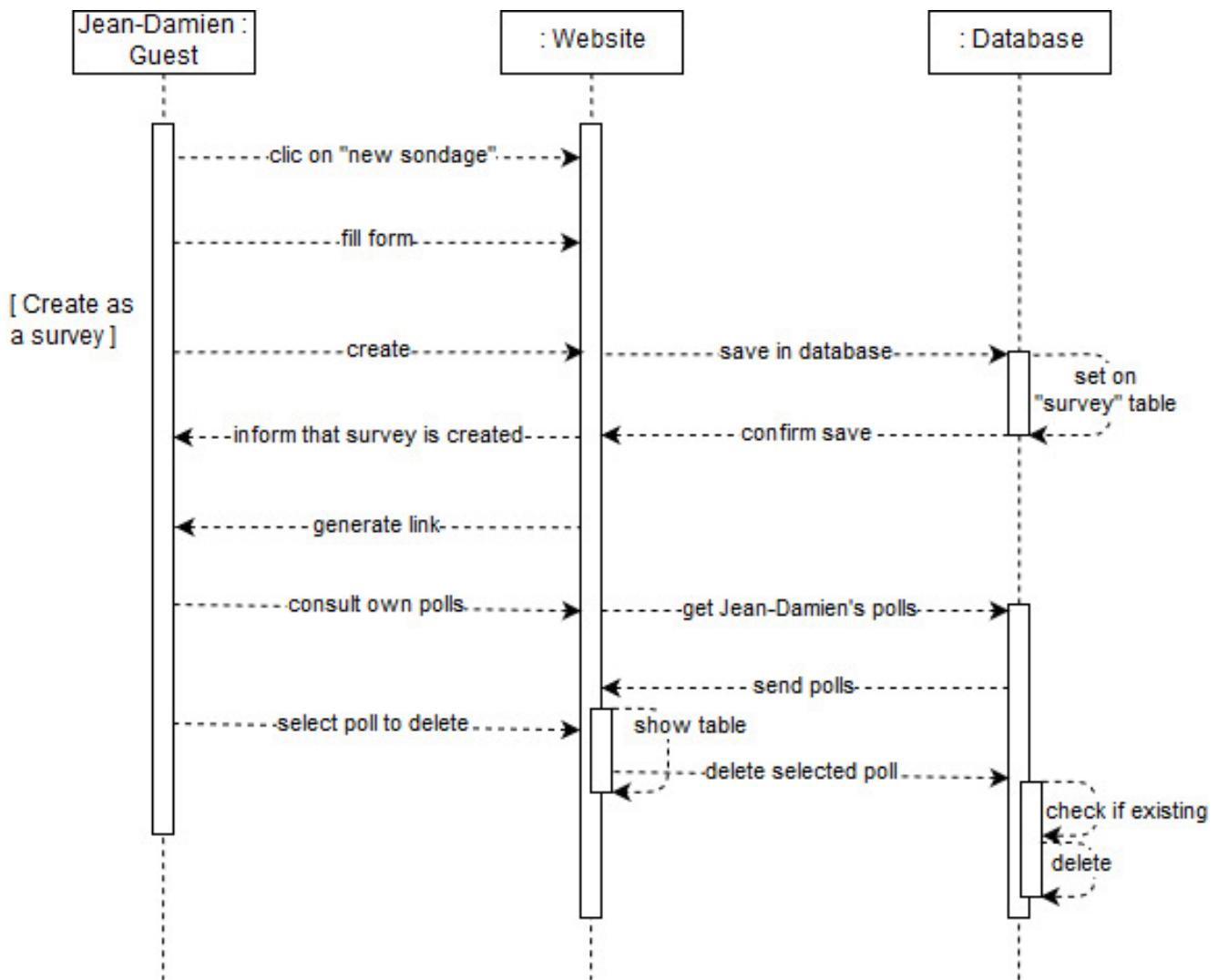


Figure 23 : diagramme de séquence du cycle de vie d'un sondage

## Diagramme d'état transition :

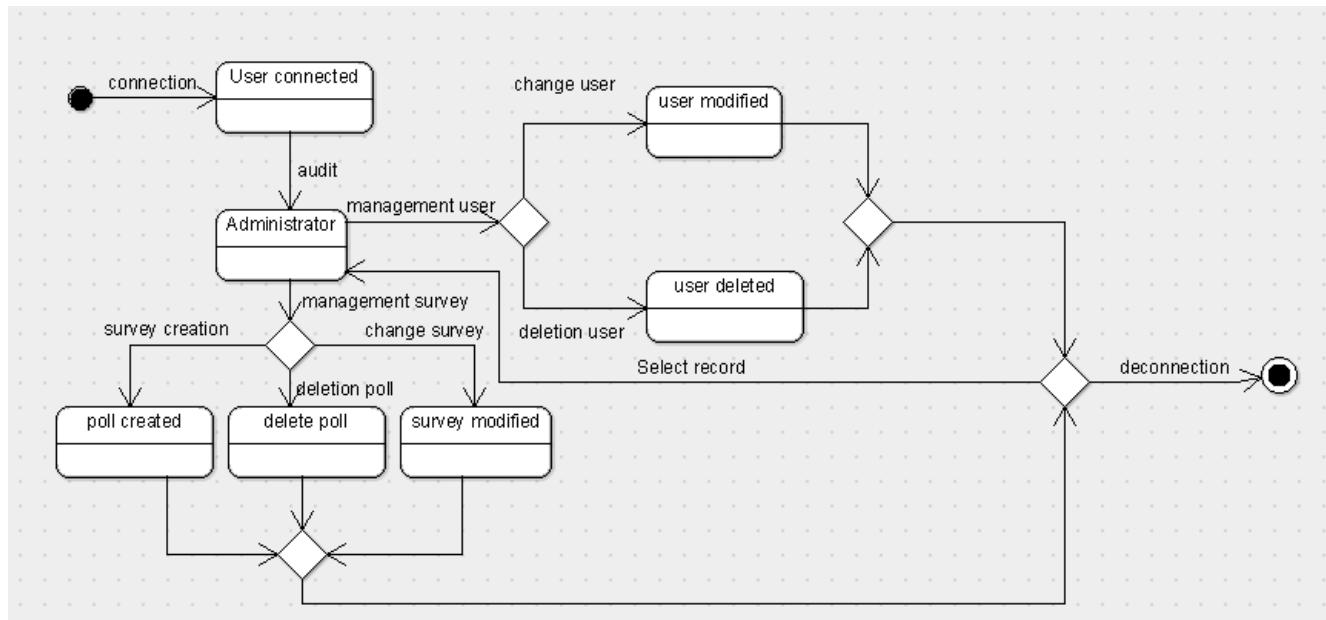


Figure 24 : diagramme d'état-transition pour un utilisateur et pour un sondage

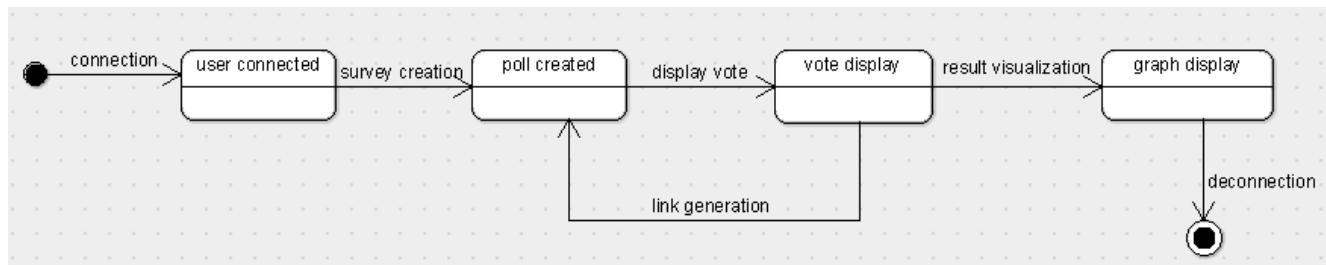


Figure 25 : diagramme d'état-transition

## Diagramme de package :

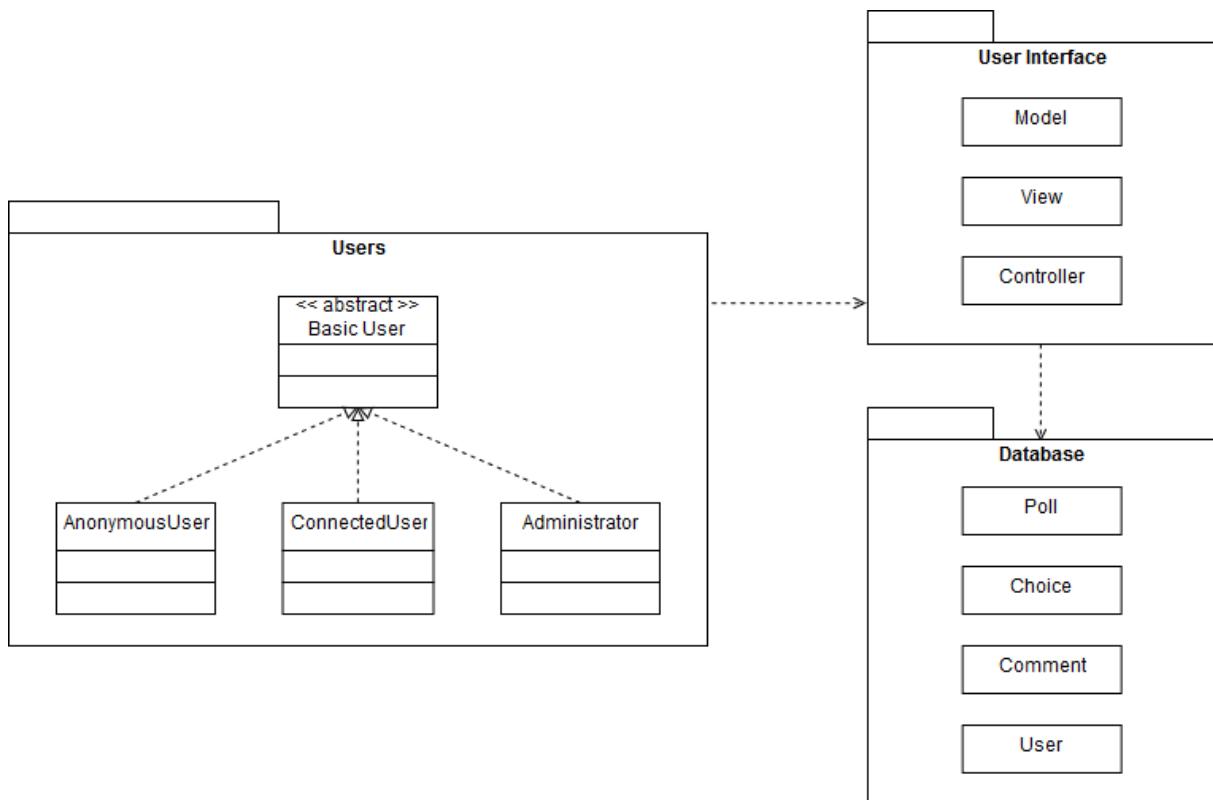


Figure 26 : diagramme de packages

# Architecture applicative

## L'architecture applicative : ce que l'on ne voit pas.

Il faut savoir qu'une application web ne fonctionne pas « toute seule » et que ce qui est affiché à l'écran est le résultat d'une réponse adaptée au « comportement » de l'utilisateur.

Les pages auxquelles nous pouvons accéder s'appellent des « vues », elles sont couplées à ce que l'on appelle des contrôleurs qui eux ont pour mission de répondre à la demande de l'utilisateur.

Par exemple, lorsqu'un utilisateur clique sur un bouton pour créer un sondage, le contrôleur lui va récupérer les données nécessaires à sa création, créer un modèle et l'inscrire dans la base de données avant de rediriger l'utilisateur vers la page du sondage qu'il vient de créer.

On distingue plusieurs types de besoins et nous nous basons sur le modèle « CRUD » (Create, Read, Update, Delete) pour pouvoir s'y adapter, c'est en fonction du « type de besoin » que nous réagirons.

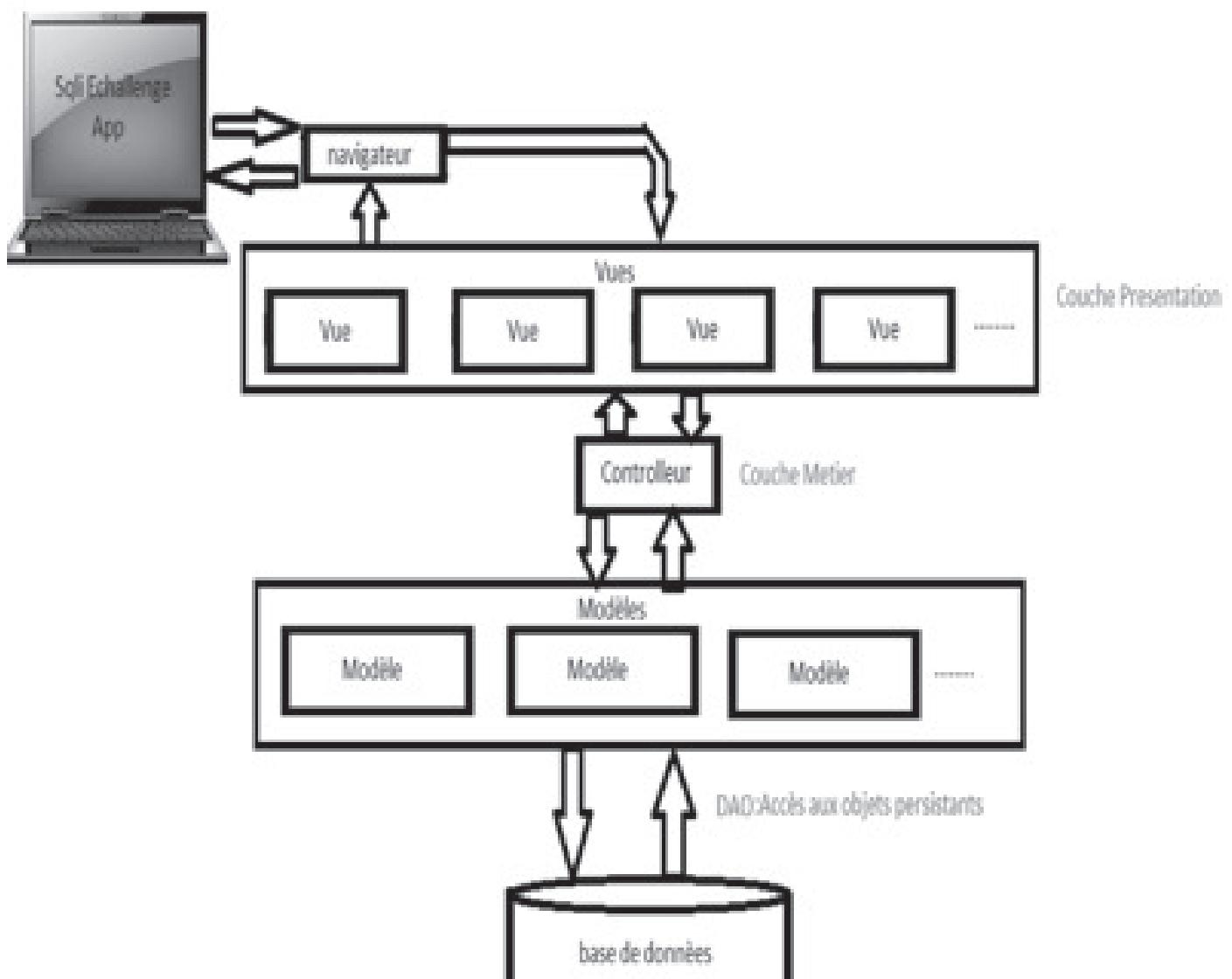


Figure 27 : architecture applicative



# Chapitre III : Implémentation





Ce chapitre présentera les différentes vues de notre application web et reviendra sur les technologies qui nous ont permis de le réaliser

”

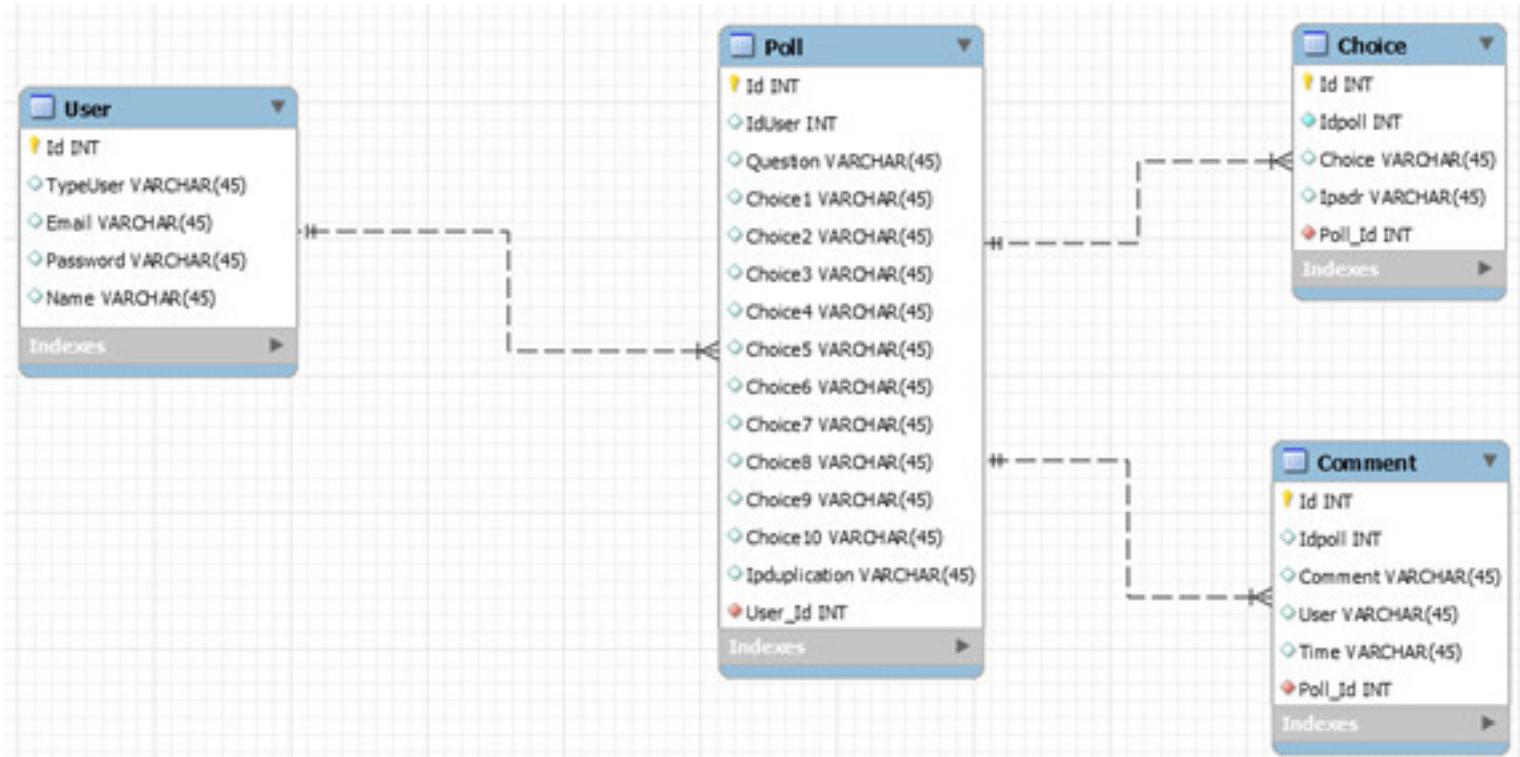


## Technologies utilisées

Tout au long de ce projet, nous avons été amenés à utiliser différentes technologies ou logiciels, certaines fois avec succès, d'autres fois non.

## Base de données

MCD :



Notre MCD représente les données que nous stockons dans notre base de données sous forme de table : chaque table est le résultat d'une réflexion préalable sur comment avoir un MCD simple mais efficace. Nous avons donc décidé de restreindre nos table à des données que l'on jugeait « indissociables » tout en conservant bien sûr des liaisons entre ces dernières via les clefs étrangères lorsque cela est nécessaire.

## Base de données SQL :

Depuis leur création, différents modèles de bases de données ont vu le jour, ces dernières évoluant en fonction des besoins ou des contraintes que leurs prédecesseurs posent.

Notre choix s'est porté sur une base de données relationnelles : un modèle de données que nous connaissons et que nous avons déjà abordé au cours de notre formation MIAGE. Nous aurions très bien pu nous porter sur une base de données NoSQL mais cela n'était pas nécessaire, la quantité de données brassée n'étant pas énorme au point de justifier ce choix.

## Serveur

### ISS :

Notre application nécessite un serveur web : elle doit prendre en charge des interactions entre un ou des clients et un serveur.

IIS est développé par Microsoft : c'est ce qui nous a poussé à le choisir puisqu'il supporte nativement les technologies que nous utiliserons pour notre projet.

## Développement

### ENVIRONNEMENT DE DÉVELOPPEMENT

Visual Studio : Visual Studio est l'environnement de développement proposé par Microsoft : c'est celui vers lequel nous nous sommes tournés puisqu'il est dans la continuité de notre projet : réaliser une application web en utilisant des technologies de l'environnement Windows.

## BACK-END

- C# et ASP.NET : ASP.NET est une référence dans le domaine de la création d'applications web : il simplifie la création de site web toujours en se basant sur les technologies web dites « classiques » (CSS, HTML, JS).

En accord avec IIS et couplé avec le langage C#, il nous permet d'ajouter une part de « programmation » à nos pages web (événements suite à une interaction avec nos boutons, adaptation de la page à son chargement en fonction de certaines variables ...).

- JS : Sencha Ext : Ext une bibliothèque JavaScript permettant de créer des applications web interactives grâce à un certain nombre de composants visuels.

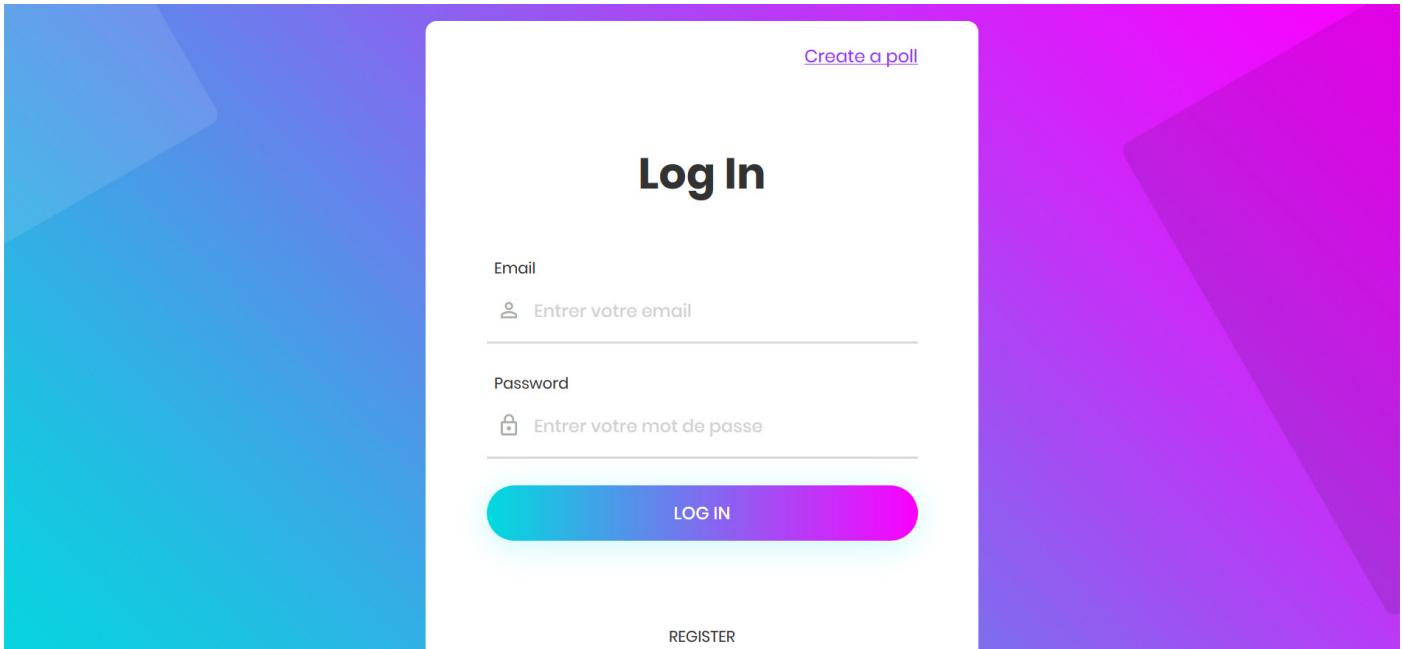
Notre choix s'est porté vers cette bibliothèque puisque nous souhaitons ajouter une représentation graphique aux résultats des sondages de notre application web.

## BACK-END : NOS MAQUETTES

# Nos maquettes

## Page de connexion

L'utilisateur, une fois son compte validé pourra se connecter à l'application web grâce aux identifiants qu'il aura précédemment défini.

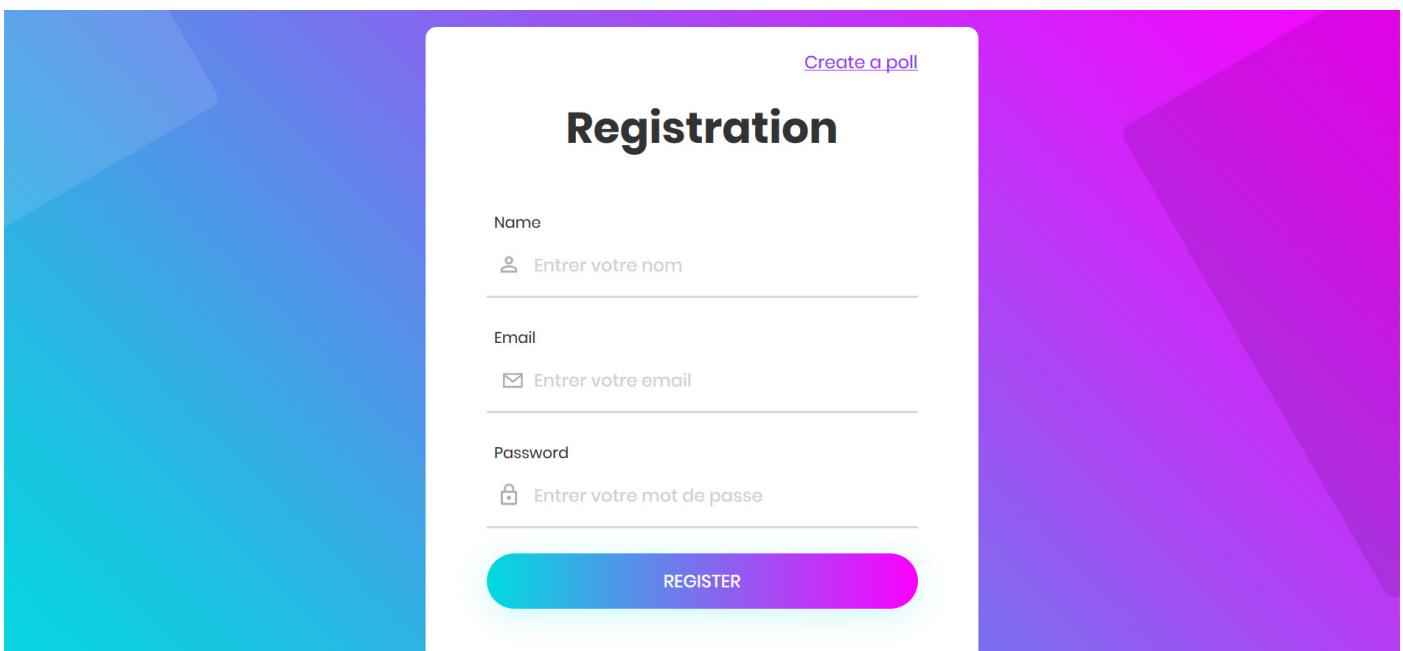


The image shows a login page with a white background and rounded corners. At the top right, there is a link labeled "Create a poll". Below it, the word "Log In" is displayed in a large, bold, black font. There are two input fields: one for "Email" with a placeholder "Entrer votre email" and one for "Password" with a placeholder "Entrer votre mot de passe". Both fields have small icons next to them. A blue button labeled "LOG IN" is centered below the password field. At the bottom left of the form, there is a link labeled "REGISTER". The background of the entire page has a gradient from blue to purple.

Figure 28 : page de connexion à l'application web

## Page d'inscription

C'est sur cette page qu'un utilisateur jusqu'alors anonyme peut se créer un compte utilisateur et ainsi accéder à la suppression des sondages qu'il aura créé. En cliquant sur « Register », un compte sera créé et devra être validé par l'administrateur.



The image shows a registration page with a white background and rounded corners. At the top right, there is a link labeled "Create a poll". Below it, the word "Registration" is displayed in a large, bold, black font. There are three input fields: one for "Name" with a placeholder "Entrer votre nom", one for "Email" with a placeholder "Entrer votre email", and one for "Password" with a placeholder "Entrer votre mot de passe". Each field has a small icon next to it. A blue button labeled "REGISTER" is centered below the password field. The background of the entire page has a gradient from blue to purple.

Figure 29 : page d'inscription à l'application web

## Page de création de sondage

C'est la page principale, la page sur laquelle un utilisateur, qu'il soit connecté ou non, peut créer un sondage de maximum dix choix. On peut d'ailleurs voir que différentes options s'offre d'ailleurs à lui (contrôle par ip ou non et choix multiple ou non)

The screenshot shows a white rectangular form on a blue and purple gradient background. At the top, it says "CREATE A NEW POLL". Below that is a section titled "WHAT WEATHER IS IT ?" containing four options: "Sunny", "Cloudy", "Rainy", and "Stormy", each with a horizontal line underneath. Below these options is a blue rounded rectangle button labeled "NEW ANSWER". Underneath the "NEW ANSWER" button are two dropdown menus: "IP duplication" and "One choice". To the right of the dropdowns is a blue rounded rectangle button labeled "CREATE".

Figure 30 : "index" ou page de création d'un sondage

## Page de vote

Lorsqu'un sondage est créé, l'utilisateur est automatiquement redirigé sur la page de vote, il peut ainsi choisir de voter ou de partager son sondage (un lien lui sera alors donné de façon à ce qu'il puisse le communiquer à qui il veut).

The screenshot shows a white rectangular form on a blue and purple gradient background. At the top, it says "WHAT WEATHER IS IT ?". Below that is a list of four options with radio buttons: "Sunny", "Cloudy", "Rainy", and "Stormy", each with a horizontal line underneath. At the bottom left is a blue rounded rectangle button labeled "VOTE", and at the bottom right is a blue rounded rectangle button labeled "SHARE".

Figure 31 : page de vote pour l'edit sondage

## Page de résultat

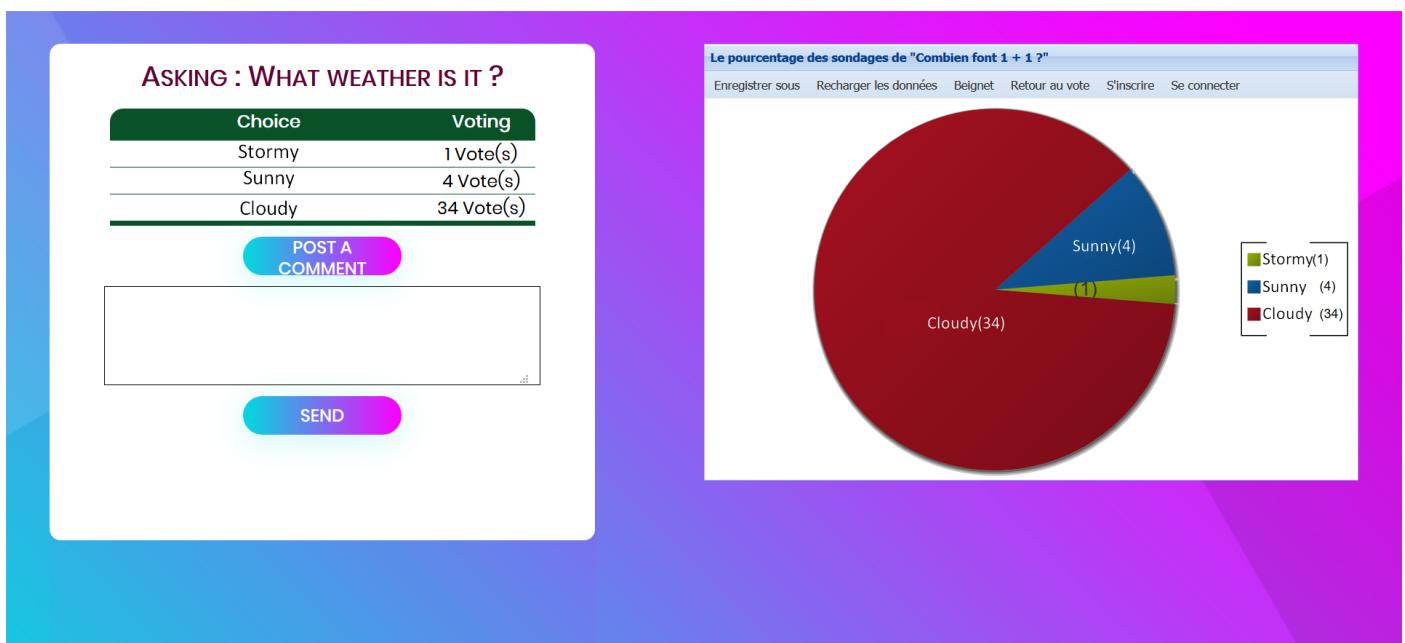


Figure 32 : page de résultats dudit sondage

La page de résultats fait suite à la page de vote, elle permet de voir combien de personnes ont voté pour chacun des choix ainsi qu'une représentation graphique sous forme de « camembert » ou « tarte » (Pie chart en anglais). Les utilisateurs, qu'ils soient connectés ou non, auront aussi la possibilité de commenter les résultats du sondage.

- Html : HyperText Markup Language (HTML) est ce qui nous a permis de structurer nos pages - (grâce à différents éléments qui s'imbriquent entre eux).

Grâce à l'ajout d'une balise « runat="server" » à l'élément, il nous a été rendu possible d'exercer certaines actions par l'intermédiaire de C# sur notre page (comme par exemple l'affichage ou non d'un élément).

- Css : Content Style Sheet (CSS) est ce qui nous permet de gérer l'apparence de nos différents éléments HTML.

C'est grâce au CSS que nous avons pu placer les différents éléments de la page où nous le souhaitions, que nous avons pu styliser nos boutons, nos différents formulaires ... Nous nous sommes servis de plusieurs librairies, notamment de BootStrap et de font-awesome pour certaines fonctionnalités qu'ils apportaient.

## Problèmes rencontrés

### Graphique

Nous avons eu du mal à choisir et maîtriser un framework et à l'intégrer à notre projet. Nous étions d'abord partis sur D3.JS puis, n'ayant pas réussi, nous avons ensuite chercher d'autres alternatives infructueuses.

Nous nous sommes finalement rabattus sur Ext JS, que nous avons finalement su maîtriser.

### Temps réel

La mise en place d'un graphique allait de paire avec la gestion en temps réel de son affichage sur la page de résultats. Actuellement, nous cherchons à mettre en place SignalR de façon à ce que chacun des clients soit mis au courant automatiquement au moindre nouveau vote attribué à un sondage qu'ils consultent.

Malheureusement, ce n'est pas encore le cas et nous travaillons toujours sur cette fonctionnalité.

### Conclusion

Ce projet nous a permis de découvrir un nouveau langage de programmation, d'utiliser des frameworks que nous n'avions encore jamais utilisé et bien sûr il nous a permis de nous confronter à la dure réalité d'un projet : des imprévus peuvent surgir à tout moment

# Nos figures

---

Figure 1 : maquette du formulaire de création d'un sondage

Figure 2 : maquette du vote pour un sondage

Figure 3 : maquette des résultats d'un sondage

Figure 4 : modèle de conception de notre projet, notre choix s'est porté sur un modèle en cascade

Figure 5 : Tâches à accomplir au sein de notre projet et estimation en temps de ces dernières

Figure 6 : Diagramme de Gantt représentant graphiquement le déroulement de notre projet en fonction du temps

Figure 7 : architecture fonctionnelle des possibilités et raccourcis d'un utilisateur enregistré sur les différents menus auxquels il a accès

Figure 8 : architecture fonctionnelle des possibilités et raccourcis d'un utilisateur enregistré sur les différents menus auxquels il a accès

Figure 9 : architecture fonctionnelle des possibilités et raccourcis d'un administrateur sur les différents menus auxquels il a accès

Figure 10 : Schéma représentant le scenario de gestion d'inscription

Figure 11 : Schéma représentant le scenario de gestion de connexion

Figure 12 : Schéma représentant le scenario de création de sondage

Figure 13 : Schéma représentant le scenario d'interaction sondage

Figure 14 : Schéma représentant le scenario d'administration

Figure 15 : Diagramme de cas d'utilisation général

Figure 16 : Schéma définit les différentes classes de l'application « Strawpoll »

Figure 17 : diagramme de séquence pour l'inscription de l'utilisateur

Figure 18 : diagramme de séquence d'inscription d'un utilisateur

Figure 19 : diagramme de séquence du cycle de vie d'un sondage

Figure 20 : diagramme de séquence de la création d'un sondage

Figure 21 : diagramme de séquence de connexion d'un utilisateur

Figure 22 : diagramme de séquence pour l'inscription de l'utilisateur

Figure 23 : diagramme de séquence du cycle de vie d'un sondage

Figure 24 : diagramme d'état-transition pour un utilisateur et pour un sondage

Figure 25 : diagramme d'état-transition

Figure 26 : diagramme de packages

Figure 27 : architecture applicative

Figure 28 : page de connexion à l'application web

Figure 29 : page d'inscription à l'application web

Figure 30 : "index" ou page de création d'un sondage

Figure 31 : page de vote pour l'edit sondage

Figure 32 : page de résultats dudit sondage