

PLAN

1. Modèle du domaine	1
2. Maquettes IHM	2
A. Adhérent	2
B. Responsable Collect'if	4
C. Tableau des actions	5
a. Interface adhérent : authentification	5
b. Interface adhérent : menu principale	6
c. Interface responsable : tableau de bord	6
d. Interface Responsable : choix du lieu	7
3. Spécifications des services	8

1. Modèle du domaine

Pour la conception du domaine nous avons identifié 3 corps métiers.

Le premier, les ADHERENT, qui s'inscrivent sur l'application.

Les adhérents : ce sont les membres de l'association. Ils se définissent par : un identifiant, un nom, un prénom, une adresse mail, une adresse postale, et leur coordonnées GPS.

L'adhérent souhaite participer à des ACTIVITES, qui sont le second corps métier :

Lieu : les lieux où peuvent se dérouler les activités. Un lieu est caractérisé par : son nom, son adresse, sa description, et ses coordonnées GPS.

Activité : l'ensemble des activités proposées par l'association. Une activité se caractérise par : son nom, le nombre de participants, si elle est par équipe.

Le type de lieu : qui permet de savoir quelle activité peut se passer à quel endroit.

Pour participer à une activité, l'adhérent doit d'abord émettre une demande. Lorsqu'une activité reçoit suffisamment de demandes pour le même jour, un événement est planifié. Ainsi le dernier corps est celui des EVENEMENTS.

L'évènement : met en relation plusieurs adhérents souhaitant pratiquer la même activité le même jour. Il est défini par un lieu, une date et une activité.

La demande : elle est émise par un adhérent pour pouvoir participer à une activité. Elle se caractérise par : sa date d'émission, l'activité et la date désirée pour l'activité.

Mise en œuvre :

Nous avons fait le choix d'associer une demande à un événement dès sa création. Ainsi, si un événement correspondant à la demande existe déjà, elle lui est associée. Sinon un nouvel événement est créé.

Nous avons rajouté des attributs statut et nombre de participants à événement pour déterminer plus facilement l'état de celui-ci. Un événement planifié est un événement dont le lieu a été défini. Le nombre de participants permet de connaître l'avancement sans faire de jointure.

Pour distinguer les événements par équipe des événements sans équipe, nous avons utilisé l'héritage. Au vue de notre conception, pour l'évènement par équipe, tous les adhérents sont placés dans l'équipe A jusqu'à la finalisation de l'évènements, où les équipes sont créés.

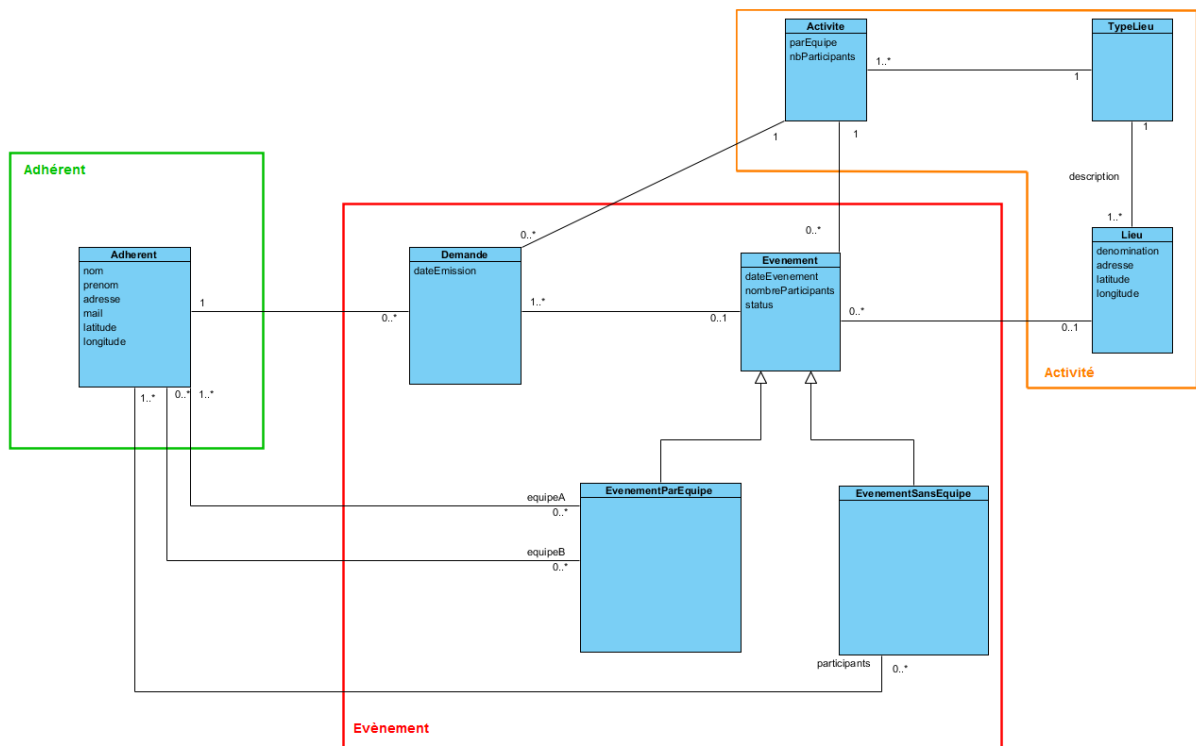


Figure 1 : schéma HTML du modèle.

2. Maquettes IHM

A. Adhèrent

A la connexion sur le site, l'adhèrent à le choix entre s'inscrire, ou se connecter grâce à son adresse mail. Il doit pour cela renseigner les champs, et valider en cliquant sur le bon bouton.

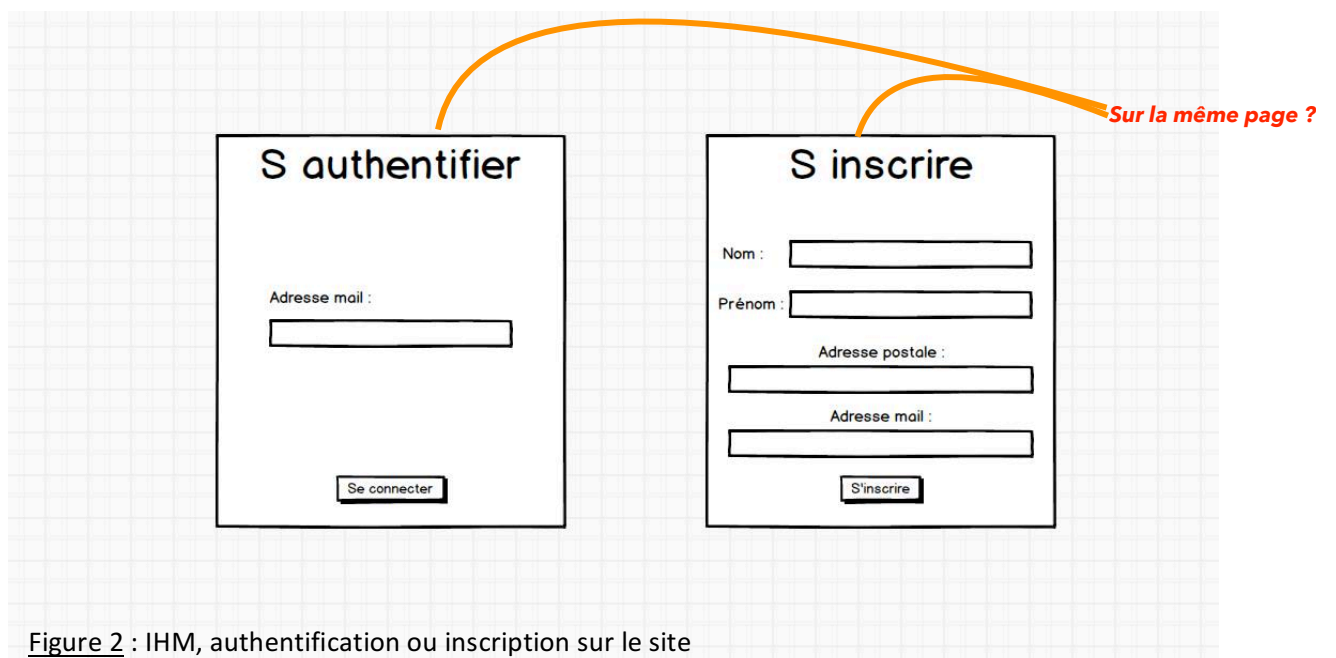


Figure 2 : IHM, authentification ou inscription sur le site

Une fois connecté sur le site, l'écran principale du site permet de créer une nouvelle demande en renseignant l'activité (menu déroulant), la date (calendrier), puis en validant par un clic sur le bouton valider.

Il est aussi possible de consulter l'historique des demandes effectuées, avec un code couleur permettant de distinguer les demandes :

Vert : l'évènement est planifié, Jaune : la demande est en cours, Orange : la demande a abouti et est maintenant dépassée, Rouge : la demande n'a pas abouti.

ScrollBar dans cette case

planifie : par rapport au lieu

Nouvelle Demande	Historique des demandes
<div> <input type="text" value="Activité"/> </div> <div> <input type="text" value="Date :"/> </div> <div> <input type="button" value="Valider"/> </div>	<div> Demande n° 4 : <div> Activité : Football Date : 04/06/16 </div> </div>
	<div> Demande n° 3 : <div> Activité : Tarot Date : 01/03/16 </div> </div>
	<div> Demande n° 2 : <div> Activité : Volley Date : 24/02/16 </div> </div>
	<div> Demande n° 1 : <div> Activité : Tricot Date : 01/01/16 </div> </div>

Figure 3 : IHM, menu principale adhérent.

En cliquant sur l'icône **+** il est possible d'avoir plus de précisions sur la demande : visualiser les données sur l'évènement associé.

Nouvelle Demande	Historique des demandes
<div> <input type="text" value="Activité"/> </div> <div> <input type="text" value="Date :"/> </div> <div> <input type="button" value="Valider"/> </div>	<div> Demande n° 4 : <div> Activité : Football Date : 04/06/16 </div> </div>
	<div> Date de l'évènement : 14/05/16 Lieu : Stade des lumières <div> <input type="button" value="Mon équipe"/> <input type="button" value="Adversaires"/> </div> </div>
	<div> Demande n° 3 : <div> Activité : Tarot Date : 01/03/16 </div> </div>
	<div> Demande n° 2 : <div> Activité : Volley Date : 24/02/16 </div> </div>

Figure 4 : IHM, détail d'un évènement.

Trouver le statut : `getstatut(>=0 ou <0 + date ?? // -1 non complet (lieu donné mais pas assez de gens), 0 à planifier (lieu non défini), // 1 planifié(equipe complete et lieu donné)`

B. Responsable Collect'if

La page principale du site pour le responsable lui permet de gérer les événements. Elle affiche une liste d'événements triés par ordres décroissants en fonction de leur date.

// planifié non dépassé : vert, demande en cours : jaune, demande à aboutie et dépassée : orange, // demande non aboutie dépassé : rouge

Il est possible de filtrer les événements en utilisant la barre de filtre situé en haut de l'interface. Elle propose de filtrer les événements par date (calendrier), Activité (menu déroulant), lieu (menu déroulant), statut (menu déroulant). Pour valider le filtrage des événements, il faut cliquer sur le bouton Filtrer.

Statut	Activité	Lieu	Date	Filtrer
Evénement n° 5 :	Statut : A planifier			
Activité : Football	Date : 04/06/16	Lieu : ∅		
Evénement n° 4 :	Statut : Non complet	Participants : 4/5		
Activité : Tarot	Date : 01/03/16	Lieu : ∅		
Evénement n° 3 :	Statut : Planifié			
Activité : Volley	Date : 24/02/16	Lieu : Gymnase Colette Besson		

Figure 5 : IHM, menu principale responsable.




Lorsqu'un événement est au statut « à planifier », il est possible de cliquer sur  pour lui affecter un lieu. Il apparait alors une nouvelle fenêtre permettant d'afficher le lieu la position de tous les participants, ainsi que la position des lieux permettant la pratique de l'activité. La sélection du lieu se fait dans un menu déroulant. Il est cependant possible de passer le pointeur sur une icône pour afficher le nom du lieu correspondant.



Figure 6 : IHM, sélection du lieu pour un événement. ( = lieu ,  = participant)

Enfin, à tout moment, il est possible d'accéder à la liste des participants à l'activité en cliquant sur l'icône **+** d'un évènement.

Si des équipes sont disponibles pour l'évènement, alors elle affiche les participants par équipe. Sinon elle affiche la liste des participants.

evenement.id

Statut	Activité	Lieu	Date	Filtrer
Évènement n° 5 :	Statut : A planifier			
Activité : Football	Date : 04/06/16	Lieu : ∅		
Évènement n° 4 :	Statut : Non complet	Participants : 4/5		
Activité : Tarot	Date : 01/03/16	Lieu : ∅		
Évènement n° 3 :	Statut : Planifié			
Activité : Volley	Date : 24/02/16	Lieu : Gymnase Colette Besson		
Manvussa Gérard Akepourlui Yann Héantie Anne	Talo Armand Hique Cyril Diotte Kelly	VS	Kilo Sandy Gino Laurent Dasso Richard	Cékilépamor Sylvie Golo Thierry Chaimoi Vivien

Figure 7 : IHM, détails des participants à un évènement.

Design sans equipe ???

C. Tableau des actions

La liste des actions principales possible en interaction avec l'IHM, et le lien avec nos services métiers.

A noter que les messages de validation et d'erreur ne sont pas détaillés. Nous proposons de le faire sous forme de pop-up, mais cela n'a pas d'influence cruciale sur le fonctionnement de l'application. Ainsi le choix de cette implémentation est laissé libre.





a. Interface adhérent : authentification

ATTENTION : Mettre les ICARs sous chaque IHM

Action	Service
Initialisation	Afficher page avec tous les champs vides
Clic sur Se connecter	Si l'adresse mail existe déjà, renvoyer un message d'erreur, sous forme de pop-up par exemple*. Sinon accès au menu principale. Utiliser : selectAdherentByMail
Clic sur S'inscrire (uniquement si mail valide)	Affichage une confirmation d'adhésion. Sous forme de pop-up par exemple*. Utiliser : creerAdherent Message erreur si infos (mail...) non valide
Perte de focus sur le champ mail	Vérification via une requête Ajax si le mail existe. Valider ou non adresse mail par un signal de couleur par exemple*. Utiliser : mailExist








Définir gestion des erreurs, messages, ... ()*

b. Interface adhérent : menu principale

Action	Service
Initialisation	Affichage de la page avec liste de toutes les demandes ordonnées par ordre chronologique (du plus récent en haut au plus ancien en bas) avec une Scroll bar sur les demandes. Utiliser : selectDemandeByIdAdherent
Clic sur 	Affichage de la liste des activités dans la comboBox. Utiliser : AfficheActivites
Clic sur 	Affichage d'un dateChooser pour sélectionner une date
Clic sur 	Vérifier si tous les champs sont remplis, puis créer la demande. Valider la création par un message, dans une fenêtre pop-up par exemple*. Obligatoire car non géré (à vérifier) Utiliser : CreerDemande
Clic sur 	Affiche sous la demande les caractéristiques de l'évènement associé : sa date, le lieu, les participants sous forme de comboBox, en distinguant les équipes s'il y en a. JavaScript Utiliser : AffichageDetails



Manque le bouton - qui masque les détails

c. Interface responsable : tableau de bord

Action	Service
Initialisation	Affichage de la liste de tous les évènements par ordre chronologique décroissant. Si status = A planifier boutons disponibles = choisir lieu, détails, supprimer Sinon boutons disponibles = détails et supprimer Utiliser : selectEvenement
Clic sur 	Affichage des 3 statuts en combo box, et d'un champ tous. Utiliser : AfficheStatut
Clic sur 	Affichage de la liste des activités dans une comboBox avec une scrollbar pour limiter la taille Utiliser : AfficheActivites
Clic sur 	Affichage de la liste des lieux en combo box avec une scrollbar pour limiter la taille Utiliser : AfficheLieux
Clic sur 	Affichage d'un dateChooser.
Clic sur 	Affiche la liste de tous les évènements correspondant aux filtre choisis, par ordre chronologique décroissant. Utiliser : AffichageDemandesFiltrees
Clic sur 	Ouvrir la fenêtre de sélection du lieu. Une fois validé, le lieu ne peut pas changer
Clic sur 	Affichage sous l'évènement des adhérents si c'est le cas les équipes. JavaScript Utiliser : AffichageDetails

Si événement en équipe : première moitié des adhérents = équipe 1, deuxième moitié = équipe 2

d. Interface Responsable : choix du lieu

Action	Service
Initialisation	Affichage de la barre latérale avec les informations complétées. Affichage sur la carte des adhérents concernés et des différents lieux disponibles. Utiliser : getEvent
Clic sur 	Affichage de la liste des lieux en combo box avec scrollbar pour limiter la taille Utiliser : AfficheLieux
Survol de la carte	Affichage des informations relatives aux points sélectionnés, en particulier pour les lieux. <i>Voir si c'est faisable</i>
Clic sur 	Ferme la fenêtre et actualise les évènements de la page tableau de bord. Utiliser : finaliserEvenement

3. Spécifications des services

Methods	Evenements créer automatiquement quand demande (si besoin)
Modifier and Type	Method and Description
static List<Adherent>	AffichageDetail (long noEvent) Permet d'obtenir la liste des adhérents participants à un événement
static List<Evenement>	AffichageEvenementsFiltres (String statut, String activite, String lieu, Date date) Service permettant d'obtenir une liste d'événements correspondants aux différents filtres Statut prend les valeurs A planifier Non complet ou Planifié, sa valeur par défaut est "Statut" ou n'importe quel autre String Activite prend comme valeur le nom d'une activité et sa valeur par défaut est "Activité" Lieu prend comme valeur le nom d'un lieu et sa valeur par défaut est "Lieu" Date prend comme valeur la date de l'événement recherché et sa valeur par défaut est 01/01/1970 Si les paramètres ont leurs valeurs par défaut alors ils ne sont pas pris en compte dans la requête Si tous les paramètres sont par défaut, tous les événements sont retournés
static List<String>	AfficheAdherentLike (String nom) Service qui renvoie une liste des noms des adhérents dont le nom commence par "nom"
static List<String>	AfficheNomActivites () Service renvoyant une liste de String des noms d'activité Ce service est utilisable pour afficher les activités dans un menu déroulant par exemple
static List<String>	AfficheNomLieux () Service renvoyant une liste de String des noms de Lieux Ce service est utilisable pour afficher les lieux dans un menu déroulant par exemple
static List<String>	AfficheStatut () Service renvoyant une liste de String des statuts Ce service est utilisable pour afficher les statuts dans un menu déroulant par exemple
static void	creerAdherent (String nom, String prenom, String adresse, String mail) Service de création d'un adhérent grâce à son nom, prénom, adresse, mail
static void	CreerDemande (String activite, Date dateEvenement, int idAdherent) Service permettant la création d'une demande Il vérifie tout d'abord si un événement existe déjà et correspond à la demande Si c'est le cas l'adhérent est ajouté à la liste des participants Sinon l'événement correspondant est créé et l'adhérent est le premier participant
static void	finaliserEvenement (String lieu, long idEvt) Service permettant de finaliser un événement, c'est à dire de lui attribuer un lieu et d'envoyer les mails Il affecte tout d'abord le lieu à l'événement puis change le statut de l'événement à "Planifié" (valeur 1) Ensuite il envoie un mail à chacun des participants de l'événement
static Evenement	getEvenement (long noEvenement) Permet d'obtenir l'événement grâce à son identifiant
static boolean	MailExist (String mail) Service permettant de savoir si un Mail existe
static List<Lieu>	obtenirLieux (Evenement evenement) Service permettant d'obtenir une liste de lieux susceptibles d'accueillir l'événement passé en paramètre
static Adherent	selectAdherentByMail (String mail) Service de connexion qui permet de sélectionner un adhérent grâce à son mail, si le retour est null c'est que l'adresse mail est invalide Sinon la session sera identifiée par l'adhérent retourné.
static List<Demande>	selectDemandeByIdAdherent (int idAdh) Service retournant l'ensemble des demandes effectuées par un adhérent dont l'id est passé en paramètre Les demandes sont triées par ordre chronologique
static void	supprimerEvenement (int idEvenement) Service non utilisé permettant la suppression d'un événement