

# Dossier d'analyse et de spécifications

<b>Rappel sujet</b>	<b>2</b>
<b>Adhérents</b>	<b>2</b>
Cas d'utilisation adhérents	2
Liste des services	3
public boolean inscrireAdherent(Adherent adh)	3
public void posterDemande(Demande demande)	3
public List<Demande> historiqueDemande(Adherent adh)	3
public List<Activite> catalogueActivites()	3
public Adherent connexionAdherent(String mail)	3
IHM adhérent	4
<b>Administrateur</b>	<b>9</b>
Cas d'utilisation administrateur	9
Liste des services	9
public List<Lieu> catalogueLieux()	9
public List<LatLng> latLngAdhEvent(Evenement event)	9
public List<LatLng> latLngLieux()	9
public List<Evenement> consulterTDB()	9
public List<Evenement> consulterTDBaPlanifier()	10
public void assignerLieuPAF(Evenement event, Lieu lieu, Float PAF)	10
IHM administrateur	10

## Rappel sujet

Vous êtes libre, demain après-midi et souhaitez faire une activité, un tarot ou une belote ? Un match de volley le samedi soir de la semaine prochaine ? COLLECT'IF, l'association événementielle locale, vous trouve des co-équipiers et met à votre disposition un local pour votre rencontre.

Pour devenir adhérentes de COLLECT'IF, les personnes intéressées doivent s'enregistrer sur le site . Les coordonnées GPS de l'adhérent sont alors automatiquement calculées. Un courriel confirmant l'inscription, ou l'informant en cas de problème, est alors envoyé au nouvel adhérent et au responsable de l'association. Une fois enregistré, l'adhérent peut s'identifier sur le site à l'aide de son adresse électronique. COLLECT'IF propose une liste d'activités sur son site (tarot, laser game etc...). Ces activités sont majoritairement gratuites mais certaines nécessitent une participation. Une fois identifié, l'adhérent peut poster une ou plusieurs demande(s) d'évènement en choisissant une activité, une date, un moment de la journée. Lorsque le nombre de demandes pour une même activité à une même date le permet (par exemple, 5 personnes souhaitant faire un tarot samedi soir), l'évènement est créé.

Le responsable de COLLECT'IF consulte quotidiennement un tableau de bord des événements créés et leur affecte un lieu disponible choisi dans un catalogue. Pour l'aider à choisir le lieu, il peut localiser les participants à l'évènement et les lieux du catalogue sur une carte. Une fois l'évènement planifié avec le lieu, un courriel est envoyé à chaque participant rappelant les informations de cet événement.

## Adhérents

### Cas d'utilisation adhérents

Un nouvel adhérent doit pouvoir s'inscrire sur le site en renseignant toutes les données nécessaires. Une fois l'inscription terminée, un mail de confirmation leur est envoyé avec leur numéro d'adhérent ou un mail d'échec leur demande de reprendre leur inscription.

Une fois enregistré, l'adhérent peut s'identifier sur le site à l'aide de son adresse électronique. Ici, pas de gestion de mot de passe.

Sur sa session l'adhérent peut alors faire une demande d'activité via un formulaire. Si cette demande abouti à la création d'un événement, l'adhérent est tenu au courant de celui-ci par mail contenant la liste des participants, la PAF, le lieu et la distance de l'adhérent au lieu.

L'adhérent possède également un historique de ses demandes indiquant celles qui ont abouti.

## Liste des services

`public boolean inscrireAdherent(Adherent adh)`

Retourne un booléen à true si l'inscription a fonctionné, false sinon.

Ajoute l'adhérent passé en paramètre si son adresse mail n'existe pas déjà dans la base et si son adresse est reconnue par l'API Google. Un message est envoyé pour préciser la nature de l'échec de l'inscription le cas échéant.

`public void posterDemande(Demande demande)`

Pas de retour.

Ajoute la demande passée en paramètre si elle n'est pas équivalente à une demande déjà présente dans la base de données et si sa date est bien postérieure à la date actuelle. De plus, si cette demande permet de remplir un événement, celui-ci est créé automatiquement, avec une valeur de Lieu et de Participation Aux Frais à null. Seul le responsable pour mettre à jour ces valeurs (voir assignerLieuPAF)

`public List<Demande> historiqueDemande(Adherent adh)`

Retourne une List de Demande contenant toutes les demandes faites par un Adherent passé en paramètre.

`public List<Activite> catalogueActivites()`

Retourne une List d'Activite contenant toutes les activités disponibles dans la base de données.

`public Adherent connexionAdherent(String mail)`

Vérifie si le String mail est l'adresse mail de l'un des adhérents de la base. Retourne l'adhérent correspondant le cas échéant, null sinon.

## IHM adhérent

1

**Bienvenue sur le site du Collect'IF !**

Connexion

S'inscrire

→ Vers 1.1

→ Vers 1.2

Intention	Contrôle	Action/Evt	Réponse
Se connecter S'inscrire	Connexion S'inscrire	Clic Clic	Passage à la page 1.1 Passage à la page 1.2

1.1

**Connectez-vous**

Adresse e-mail :

Valider ✓

Pas encore membre ?  
Inscrivez-vous !

Vers 1.2

Vers 1.3 en  
cas de succès

Vers 1.1.2  
en cas  
d'échec

Intention	Contrôle	Action/Evt	Réponse
Se connecter	Valider	Clic	→ connexionAdherent(mail) // Si adresse mail non valide alors Passage à la page 1.1.2 // Sinon

S'inscrire	Inscrivez-vous	Clic	Passage à la page 1.1.1 Passage à la page 1.2
------------	----------------	------	--

1.2

## Inscription

Nom :

Prénom :

Mail :

Adresse :  Rue

Ville

Revenir à l'accueil du site  
 ↳ Vers 1

Valider ✓  
 ↳ Vers 1.2.1

Intention	Contrôle	Action/Événement	Réponse
S'inscrire	Valider	Clic	→ inscrireAdherent(adh) // Si inscription réussie alors Envoi mail de confirmation Passage à la page 1.1.1 // Sinon Envoi mail d'échec Passage à la page 1.1.2
Retour accueil	Retour à l'accueil du site	Clic	Passage à la page 1



Intention	Contrôle	Action/Event	Réponse
Retour accueil	Retour	Clic	Passage à la page 1

1.1.2

Échec de connexion

Veuillez vérifier votre adresse e-mail ou inscrivez-vous

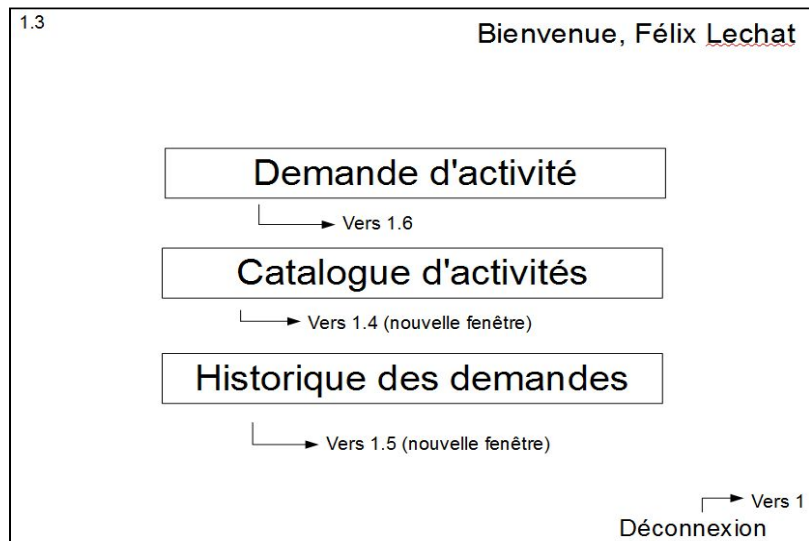
Inscription

Retour

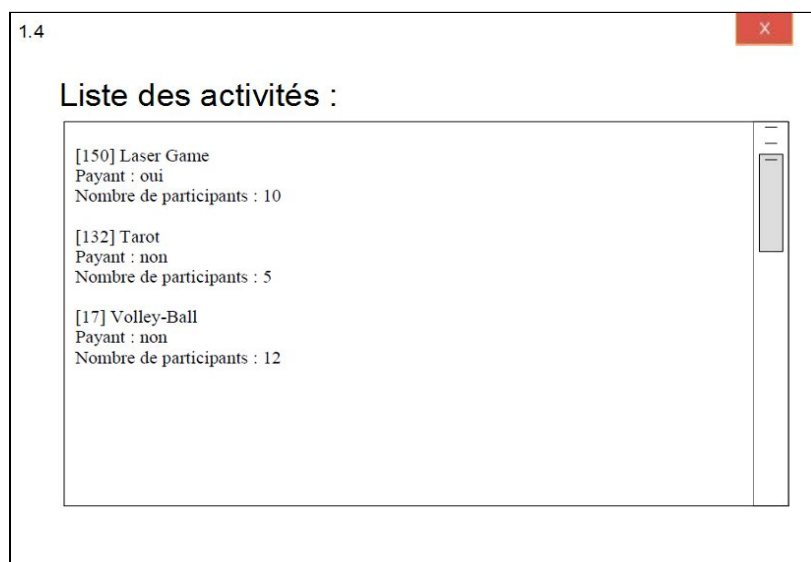
↳ Vers 1.2

↳ Vers 1

Intention	Contrôle	Action/Event	Réponse
Retour accueil	Retour	Clic	Passage à la page 1
S'inscrire	Inscription	Clic	Passage à la page 1.2



Intention	Contrôle	Action/Evt	Réponse
Demander une activité	Demande d'activité	Clic	Passage à la page 1.6
Consulter les activités	Catalogue d'activités	Clic	→ catalogueActivites()
Consulter l'historique	Historique des demandes	Clic	→ historiqueDemande(adh)
Se déconnecter	Déconnexion	Clic	Passage à la page 1



Intention	Contrôle	Action/Evt	Réponse
-----------	----------	------------	---------

Quitter	Croix fermeture	Clic	Fermeture de la fenêtre
---------	-----------------	------	-------------------------

1.5

**Historique des demandes :**

[124] Tarot  
Date: 12/03/2017  
Moment: soir  
Statut: validée

[87] Volley-ball  
Date: 01/02/2017  
Moment: après-midi  
Statut: non validée

[33] Lasergame  
Date: 20/01/2017  
Moment: après-midi  
Statut: validée

Intention	Contrôle	Action/Evt	Réponse
Quitter	Croix fermeture	Clic	Fermeture de la fenêtre

1.6

**Demande d'activité**

Nom activité:

Date :

Moment :

Revenir à la page principale [Vers 1.3](#)

**Valider ✓** [Vers 1.3](#)

Intention	Contrôle	Action/Evt	Réponse
Retour en arrière	Revenir à la	Clic	Passage à la page 1.3



Demander une activité	page principale  Valider	Clic	→ posterDemande(demande) // Si demande déjà postée ou date incorrecte alors Demande non créée // Sinon Création demande // Si demandes suffisantes alors Création événement
-----------------------	--------------------------------	------	---

## Administrateur

### Cas d'utilisation administrateur

Le compte administrateur est déjà créé. Il y accède en rentrant "rebarbatIF" à la place de son identifiant mail.

L'administrateur doit avoir accès à tous les événements créés. Afin qu'il puisse travailler au mieux, il doit également avoir accès à la liste de tous les événements en attente d'être gérés.

L'administrateur gère les événements en leur affectant un lieu et si l'événement est payant, un prix. Le prix étant dépendant de d'autres paramètres que celui de l'événement (ex: salle du lasergame), son estimation est laissée à la charge de l'administrateur. L'affectation du lieu est facilitée par une API Google qui lui indique les adresses des participants et des lieux possibles pour le type d'activité lié à l'événement sur une carte.

### Liste des services

`public List<Lieu> catalogueLieux()`

Retourne une List de Lieu contenant tous les lieux disponibles dans la base de données.

`public List<LatLng> latLngAdhEvent(Evenement event)`

Retourne une List de LatLng contenant les latitudes et longitudes de tous les adhérents qui participent à l'événement passé en paramètres.

L'objet LatLng (du package `com.google.maps.model.LatLng`) contient deux doubles (lat et lng), contenant respectivement la latitude et la longitude.

```
public List<LatLng> latLngLieux()
```

Retourne une List de LatLng contenant les latitudes et longitudes de tous les lieux de la base.

```
public List<Evenement> consulterTDB()
```

Retourne une List de Evenement qui contient tous les événements de la base de données.

```
public List<Evenement> consulterTDBaPlanifier()
```

Retourne une List de Evenement qui contient tous les événements de la base de données qui n'ont pas été planifiés (i.e le Lieu et le PAF ne sont pas encore définis).

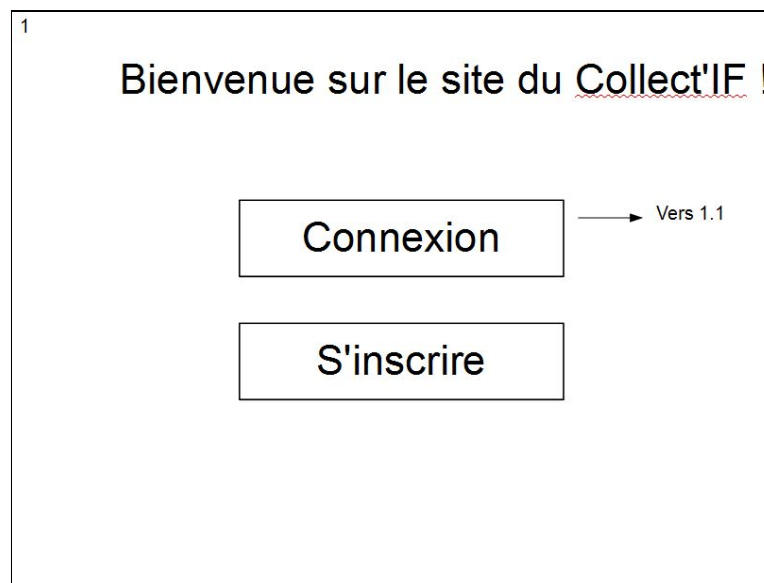
```
public void assignerLieuPAF(Evenement event, Lieu lieu, Float PAF)
```

Pas de retour.

Assigne le Lieu et la Participation Aux Frais (PAF) associés à un Evenement, tous trois passés en paramètres. Si un événement n'a pas de participation aux frais, le paramètre PAF doit prendre la valeur 0F.

Une fois le Lieu et le PAF assignés, un mail (console) est envoyé à tous les adhérents inscrits à cet événement, donnant les informations sur l'événement.

## IHM administrateur



Intention	Contrôle	Action/Evt	Réponse
Se connecter	Connexion	Clic	Passage à la page 1.1

1.1

## Connectez-vous

Adresse e-mail :

Valider ✓

Pas encore membre ?  
Inscrivez-vous !

→ Vers 1.3 en cas de succès  
→ Vers 1.1.2 en cas d'échec

Intention	Contrôle	Action/Événement	Réponse
Se connecter	Valider	Clic	→ connexionAdherent(mail) // Si adresse mail non valide alors Passage à la page 1.1.2 // Sinon si mail=="rebarbatIF" Passage à la page 2

1.1.2

## Échec de connexion

Veuillez vérifier votre adresse e-mail ou inscrivez-vous

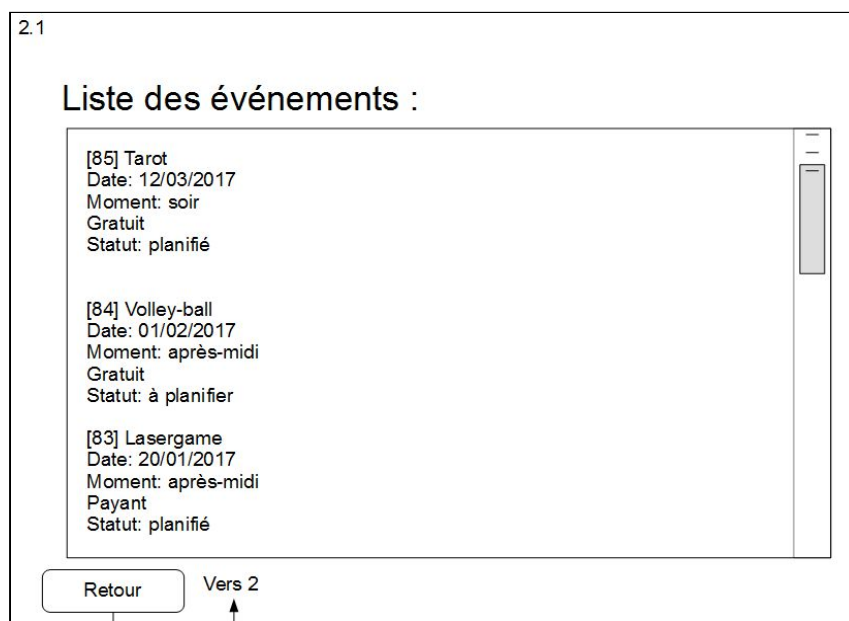
Inscription
Retour

→ Vers 1

Intention	Contrôle	Action/Événement	Réponse
Retour à l'accueil	Retour	Clic	Passage à la page 1



Intention	Contrôle	Action/Evt	Réponse
Consulter les événements	Liste des événements	Clic	→ consulterTDB() Passage à la page 2.1
Consulter les événements à traiter	Liste des événements à traiter	Clic	→ consulterTDBaPlanifier() Passage à la page 2.2
Se déconnecter	Déconnexion	Clic	Passage à la page 1



Intention	Contrôle	Action/Evt	Réponse
Retour en arrière	Retour	Clic	Passage à la page 2

2.2

Liste des événements à traiter :

[84] Volley-ball  
Date: 01/02/2017  
Moment: après-midi  
Gratuit

[81] Tarot  
Date: 02/02/2017  
Moment: soir  
Gratuit

[80] Go  
Date: 28/01/2017  
Moment: après-midi  
Gratuit

Traiter

→ Vers 2.2.1

Traiter

Traiter


Retour      Vers 2

Intention	Contrôle	Action/Evt	Réponse
Retour en arrière	Retour	Clic	Passage à la page 2
Traiter un événement	Traiter	Clic	→ latLngLieux() → latLngAdhEvent(Evenement event) Passage à la page 2.2.1

2.2.1

Affecter PAF et lieu

PAF



[81] Tarot  
Date: 02/02/2017  
Moment: soir  
Gratuit

Nouvelle fenêtre 2.2.2  
Afficher liste des lieux

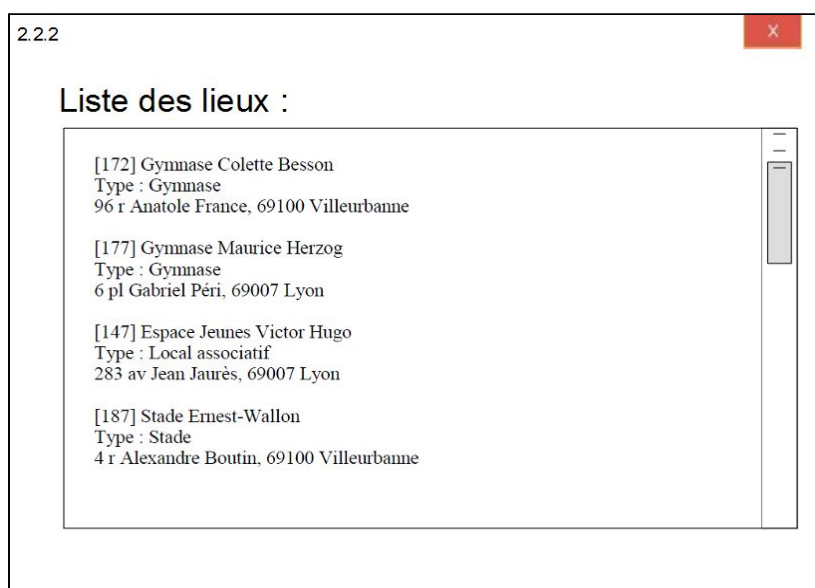
Sélectionner lieu :

Lieu	
Gymnase Colette Besson	✓
Gymnase Maurice Herzog	✓
Espace Jeunes Victor Hugo	
Stade Ernest-Wallon	

Vers 2.2      Vers 2.2

Retour      Valider

Intention	Contrôle	Action/Evt	Réponse
Retour en arrière	Retour	Clic	Passage à la page 2.2
Planifier l'événement	Valider	Clic	→ assignerLieuPAF(Evenement event, Lieu lieu, Float PAF) // Si absence d'un paramètre alors Validation impossible //Sinon Envoi d'un mail de modalités d'événement à tous les adhérents concernés Passage à la page 2.2



Intention	Contrôle	Action/Evt	Réponse
Quitter	Croix fermeture	Clic	Fermeture de la fenêtre