

Compte rendu TP DASI : Application Proact'IF

L'ensemble du travail réalisé ainsi que son versionnage est disponible sur GitHub à l'URL :
<https://github.com/baptiste-pauletto1/DASI>

Table des matières

I.	Introduction	2
II.	Modèle du domaine	2
a.	Diagramme des cas d'utilisation	2
b.	Diagramme d'états-transitions	3
c.	Diagramme de classe	4
III.	Maquettes IHM	5
a.	Enchaînement de fenêtres (états-transitions)	5
b.	Espace client	5
1.	Fenêtre d'accueil (connexion et inscription)	6
2.	Fenêtre de création de demande d'intervention	6
3.	Fenêtre d'historique	7
c.	Espace employé	8
1.	Fenêtre d'accueil (connexion)	8
2.	Fenêtre de consultation de l'intervention courante	9
3.	Fenêtre de clôture d'intervention	9
4.	Fenêtre du tableau de bord	10
IV –	Spécification des services	11
a.	Services liés aux clients	11
b.	Services liés aux employés	12
V-	Conclusion	12

I. Introduction

Proact'if est un service de conciergerie agile et écologique. En effet, ce service permet à des clients de faire des demandes d'intervention à leur domicile par des employés de Proact'IF lorsqu'ils ne sont pas chez eux. Parmi ces demandes d'interventions, on distingue 3 types : animal, incident et livraison. Chacun de ces types a des propriétés particulières, par exemple, si un animal est impliqué, il sera alors spécifié dans la demande d'intervention, alors que si c'est une livraison, le nom de l'entreprise responsable ainsi que l'objet attendu devront être précisés.

Cependant, ils ont tout de même des propriétés communes, telle qu'une courte description fournie par l'utilisateur ou l'identité du client à l'origine de cette dernière.

Dans ce compte-rendu nous allons alors nous attarder sur la réalisation de l'application nécessaire à ce service tout en fournissant la documentation concernant le travail déjà réalisé.

II. Modèle du domaine

Pour permettre une compréhension la plus complète du travail déjà réalisé, nous avons fait le choix de débiter par une modélisation des cas d'utilisation de l'application Proact'IF. En effet, c'est ce dernier qui a régi notre réflexion initiale et que nous avons souhaité mettre en place tout au long du projet.

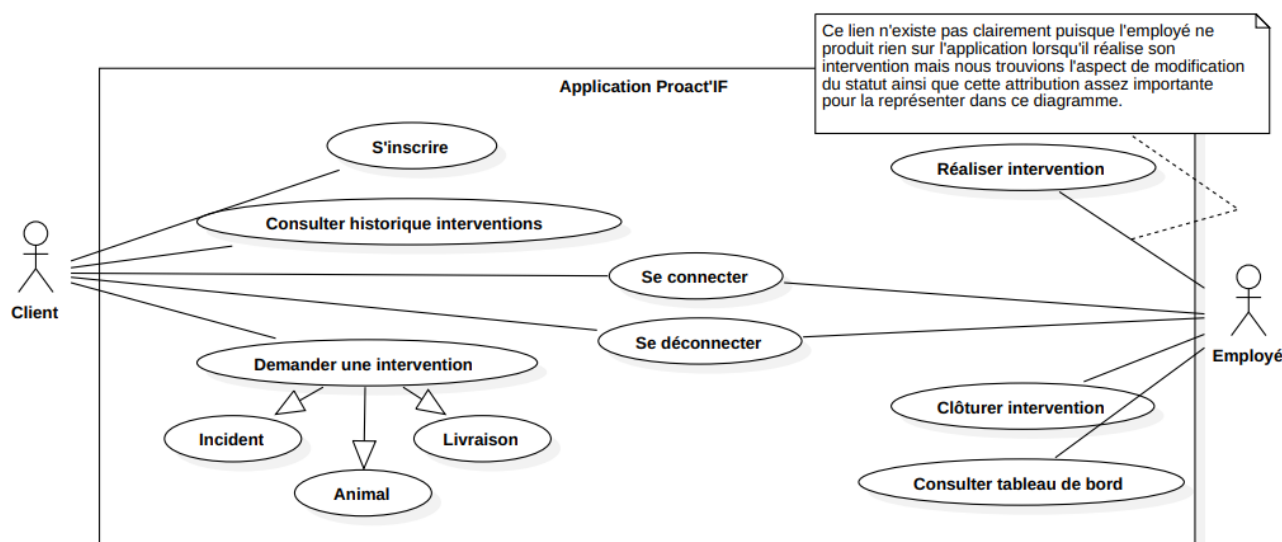
Nous distinguons alors deux types d'utilisateurs principaux : le client et l'employé, ceux-ci ont accès à des services radicalement différents, d'où l'importance de les différencier.

L'entreprise compte environ 5000 clients pour une quinzaine d'employés et ces derniers ne se déplacent qu'à vélo par souci écologique, ces informations ont été prises en compte pour le développement.

A propos du diagramme, le client pourra interagir avec notre application depuis son téléphone mobile ou un ordinateur via une application web et ses interactions seront gérées par des services avec un intitulé assez proche de celui disponible dans le diagramme ci-dessous.

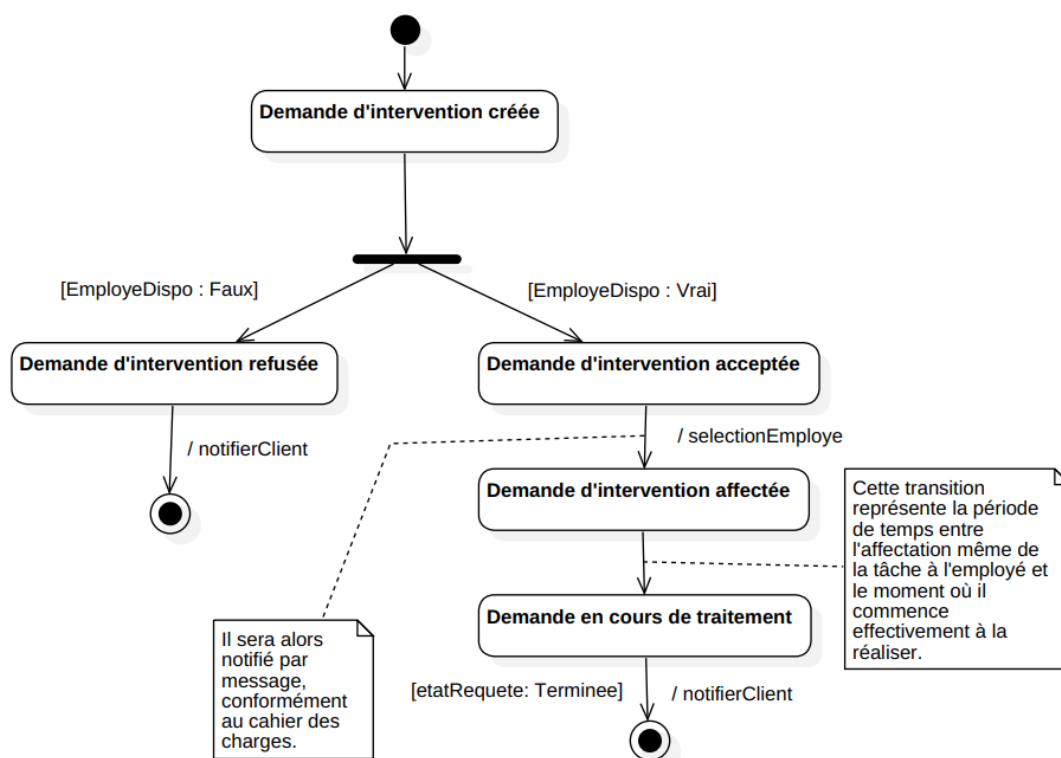
Certains des cas d'utilisation représentés ci-dessous pourraient être étendus, c'est pourquoi, tout au long de notre développement nous avons veillé à assurer la maintenabilité de ces derniers (nous pensons notamment à l'ajout d'autres types d'interventions ou de nouvelles actions d'échanges entre l'employé et le client demandant une intervention).

a. Diagramme des cas d'utilisation



Maintenant que les bases des possibilités utilisateurs sont posées, nous allons nous intéresser aux entités régissant les interactions entre ces utilisateurs dans l'application, soit les demandes d'intervention. Le diagramme ci-dessous représente donc le cycle de vie d'une de ces demandes d'intervention. Le client en crée une et l'application se charge alors d'assigner automatiquement un employé à cette demande s'il y en a, puisque ces derniers possèdent des horaires de travail, comme toute personne, et peuvent être déjà en train de réaliser une intervention lorsqu'une nouvelle arrive, leur statut étant indisponible, ils ne seront alors pas pris en compte dans la recherche d'employés pour cette demande. S'il n'y en a pas, le client est notifié et est invité à réessayer plus tard. S'il y en a, l'employé est également notifié et commence tout de suite l'intervention. Au moment où il finit, il clôture cette demande via l'application et le client est à nouveau prévenu par message du succès ou de l'échec de l'opération.

b. Diagramme d'états-transitions



Légende

[variable : valeur] évaluation de condition
/ action service déclenché

L'opération de sélection de l'employé pour aller accomplir la tâche n'est pas faite par hasard, en effet, plusieurs facteurs entrent en jeu : tout d'abord, sa disponibilité et ensuite sa distance à vélo jusqu'au domicile du client demandeur.

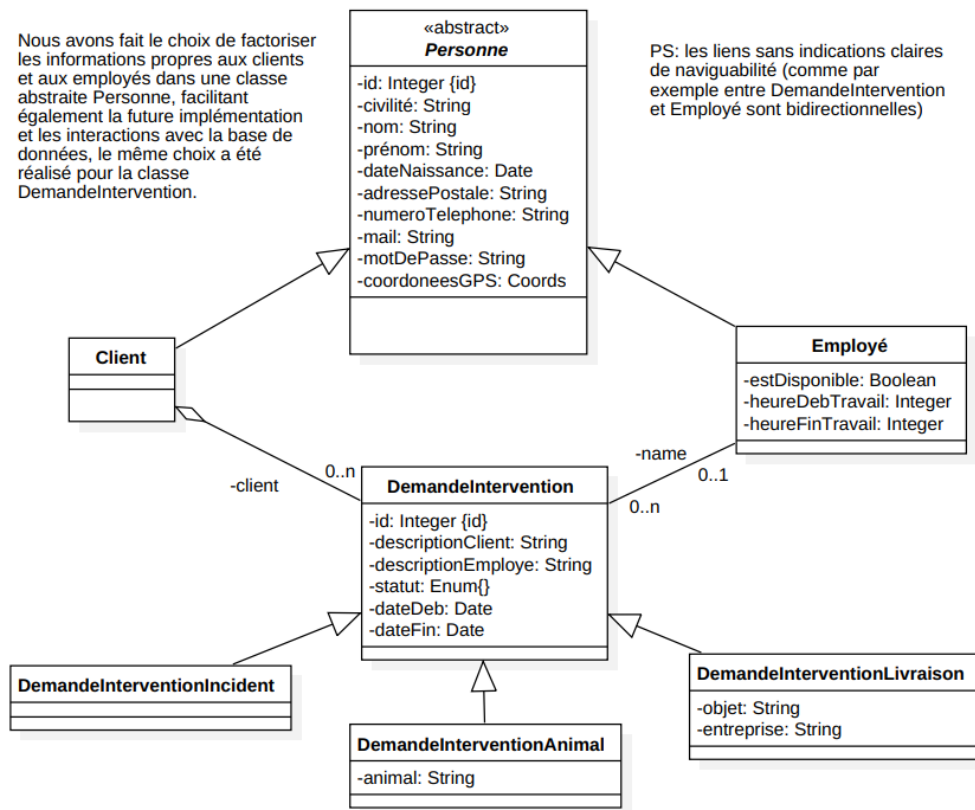
Ce calcul est réalisé par un utilitaire qui nous a été fourni et qui fait usage de l'API Google Maps.

Il est important de préciser que nous ne prenons pas en compte la situation actuelle de l'employé, c'est-à-dire que si l'employé A termine une intervention proche d'une nouvelle intervention, il ne sera pas forcément affecté à cette dernière si un autre employé réside plus près de celle-ci.

En effet, nous réalisons le calcul depuis les coordonnées GPS du domicile de l'employé et non sa position courante.

c. Diagramme de classe

Nos trois objets métiers clairement identifiés sont donc : Employé, Client et Intervention



En plus des annotations d'ores et déjà présentes sur le diagramme, nous souhaitons insister sur plusieurs choix réalisés lors de la conception et puis mis en place dans l'application :

Héritage

Comme les classes *Employé* et *Client* ont un certain nombre d'attributs en commun (nom, prénom, adresse, mail ...) nous avons décidé de créer une classe mère abstraite *Personne*. Ainsi, on pourra directement hériter de ces attributs en commun sans redondance dans le code. La classe est abstraite car il ne sera pas possible d'instancier une *Personne*, elle doit forcément être soit un employé soit un client dans notre contexte. Il n'y a pas d'autres cas possibles, mais nous pouvons imaginer à l'avenir une nouvelle entité tel qu'administrateur maintenance ou un système de fidélité.

De même, il existe plusieurs types de demandes d'intervention et ces différents types ont grand nombre d'attributs identiques. On a alors procédé de la même façon : on crée une classe mère abstraite *DemandeIntervention* et les 3 types de demandes d'intervention en héritent, assurant à nouveau factorisation et extensibilité.

Lien logique entre les classes

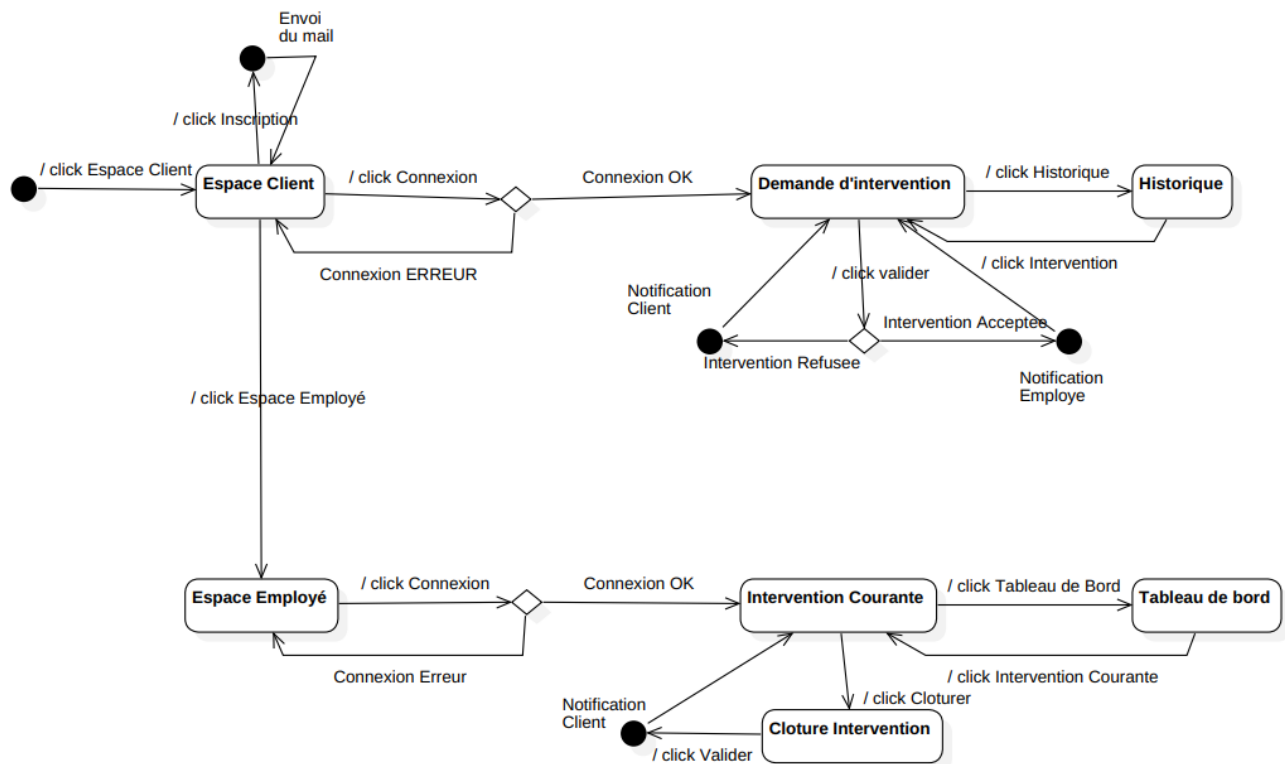
Une demande d'intervention ne peut pas exister sans être associée à un client. Un client peut alors émettre 0 ou plusieurs demandes, même si théoriquement il devrait être limité par une seule à la fois (il ne faudrait pas qu'il puisse forcer tous les employés à aller chez lui, en créant un grand nombre de demandes, par exemple).

En revanche, une demande peut exister sans être associée à un employé (dans le cas où aucun employé est disponible pour faire une intervention). D'autant plus qu'à la suite de la création de celle-ci, cette entité existe sans être associée à un employé, en effet, l'affectation par le système ne sera pas forcément instantanée si un grand nombre d'utilisateurs font appel à leurs services en même temps.

III. Maquettes IHM

a. Enchaînement de fenêtres (états-transitions)

Après avoir vu les composantes ainsi que la modélisation de l'application et avant de présenter chacune des IHM qui seront à réaliser, nous souhaitons proposer un enchaînement de fenêtres complet afin de permettre à l'équipe qui prendra en charge la suite du projet d'avoir une ligne directrice.



La légende de cet enchaînement est la même que le précédent diagramme états-transitions avec simplement les commentaires (informations sans notations particulières) en plus.

b. Espace client

Dans cet espace, le client a accès à 3 pages :

- La première qui est une page d'accueil et depuis laquelle il aura la possibilité de se connecter ainsi que de s'inscrire s'il ne l'est toujours pas.
- La page de création de demande d'intervention où il pourra choisir le type d'intervention qu'il souhaite et qui adaptera alors ses champs afin de permettre de rentrer les informations nécessaires au bon fonctionnement de l'application (type = Animal, champ animal...). De plus, un espace sera lui disponible constamment où le client pourra apporter des détails.
- La page de l'historique des demandes faites par le client connecté, une liste sera alors dressée chronologiquement en plaçant les demandes les plus récentes en premier. Des brèves informations telles que la date ou encore le type et le statut seront affichées et si l'utilisateur vient à cliquer sur une de ces lignes, celle-ci s'étendra alors et permettra d'accéder à chacun des attributs qui la composent. On pourra alors savoir quel employé a pris en charge la demande, à quelle heure et le commentaire laissé par l'employé.

1. Fenêtre d'accueil (connexion et inscription)

PROACT'IF Le service de conciergerie agile et écologique

Espace Client **Espace Employés**

Accéder à mon espace client : Pas encore inscrit ?

Login : Civilité : Adresse Postale :

Mot de passe : Nom :

Prénom :

Date de naissance : Téléphone :

Mail : (+33)

Mot de passe :

Intention	Contrôle	Action	Réponse
Se connecter	Bouton : Connexion	Clic	Service authentifierClient Si retour != null Ouverture page Demande d'intervention Sinon Message : "Erreur de connexion"
S'inscrire	Bouton : Inscription	Clic	Service inscrireClient Message : "Inscription prise en compte, vérifiez votre boîte mail"

2. Fenêtre de création de demande d'intervention

PROACT'IF Espace client

Demande d'intervention **Historique** **Déconnexion**

Type : ▼

Animal : si type = Livraison

xor Objet : Entreprise de Livraison:

xor Incident :

Description :

Par souci de lisibilité, nous avons déplacé les ICARs de la fenêtre précédente ici :

Intention	Contrôle	Action	Réponse
Demander l'intervention	Bouton : Valider	Clic	Service faireDemandeIntervention Message : "Demande prise en compte, vous serez informé de son statut par sms"
Consulter l'historique	Menu : Historique	Clic	Ouverture page Historique
Se déconnecter	Bouton : Déconnexion	Clic	Renvoie vers la page d'accueil

3. Fenêtre d'historique

PROACT'IF
Espace client

Demande d'intervention
Historique
Déconnexion

Date : 23/03/2019 Type : Animal Statut : En Cours

Date : 18/01/2019 Type : Livraison Statut : Finie : Succès

Intention	Contrôle	Action	Réponse
Faire une demande d'intervention	Menu : Demande d'intervention	Clic	Envoie vers la page de demande d'intervention
Se déconnecter	Bouton : Déconnexion	Clic	Renvoie vers la page d'accueil

c. Espace employé

Dans cet espace, l'employé a accès à 4 pages :

- La page de connexion, identique à celle d'un utilisateur lambda, mais il n'est pas possible de s'inscrire en tant qu'employé, en effet, ces derniers sont ajoutés en dur dans la base de données puisqu'ils font partie d'un processus différent de celui de l'inscription habituelle.
- La page de consultation de l'intervention courante associée à l'employé connecté où il pourra retrouver toutes les informations propres au client ayant fait appel à une intervention. Il pourra également consulter le commentaire laissé lié à cette intervention en particulier.
- La page de clôture de l'intervention courante, l'employé pourra alors spécifier comment s'est déroulé cette dernière (succès/échec), modifier l'heure s'il le souhaite et ajouter un commentaire qui sera ensuite envoyé au client qui avait fait cette demande.
- Le tableau de bord de toutes les interventions du jour avec un affichage de carte type Google Map et des points permettant de connaître les informations propres à une intervention. Les interventions en cours et celles terminées de la journée, seront affichées.

1. Fenêtre d'accueil (connexion)

PROACT'IF Le service de conciergerie agile et écologique

Espace Client **Espace Employés**

Accéder à mon espace employé :

Login :

Mot de passe :

Connexion

Intention	Contrôle	Action	Réponse
Se connecter	Bouton : Connexion	Clic	Service authentifierEmploye Si retour != null Ouverture page Intervention Courante Sinon Message : "Erreur de connexion"

2. Fenêtre de consultation de l'intervention courante

PROACT'IF
Espace Employés

Intervention Courante
Tableau de bord
Déconnexion

Client :
Adresse :

Type :
Heure/Date :

Mail :
Telephone :

Animal :

si type = Livraison
Entreprise de Livraison:

xor Objet :
xor Incident :

Description :

Clôturer l'intervention

Intention	Contrôle	Action	Réponse
Clôturer l'intervention	Bouton : Clôturer l'intervention	Clic	Ouverture page Clôture d'intervention
Consulter le tableau de bord	Menu : Tableau de Bord	Clic	Ouverture page Tableau de bord
Se déconnecter	Bouton : Déconnexion	Clic	Renvoie vers la page d'accueil

3. Fenêtre de clôture d'intervention

PROACT'IF
Espace Employés

Intervention Courante
Déconnexion

Heure de fin :

Statut :

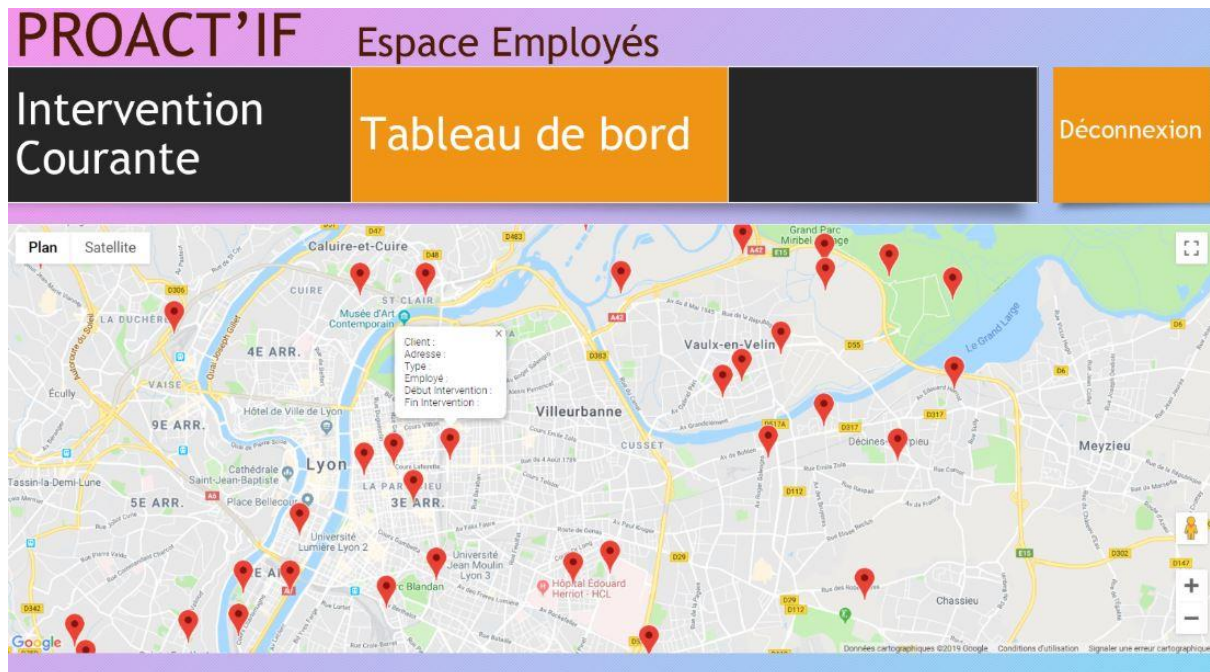
Commentaire :

Valider

Par souci de lisibilité, nous avons déplacé les ICARs de la fenêtre précédente ici :

Intention	Contrôle	Action	Réponse
Confirmer la clôture	Bouton : Valider	Clic	Service clôturéDemandeIntervention
Se déconnecter	Bouton : Déconnexion	Clic	Renvoie vers la page d'accueil

4. Fenêtre du tableau de bord



Intention	Contrôle	Action	Réponse
Voir l'intervention courante	Menu : Intervention Courante	Clic	Renvoie vers la page Intervention courante
Se déconnecter	Bouton : Déconnexion	Clic	Renvoie vers la page d'accueil

IV – Spécification des services

Nous avons regroupé l'intégralité des spécifications de services dans le tableau ordonné ci-dessous. Dans la réalisation de ce dernier, nous avons tâché d'être le plus précis possible sans non plus rentrer dans des détails d'implémentation qui pourrait déranger la future création d'IHM. D'autre part, nous avons essayé de rester le plus cohérents possible avec les différents descriptifs établis auparavant et le descriptif des cas d'usages fourni initialement.

a. Services liés aux clients

Intention	Action	Succès/Echec	Paramètre	Réponse
Le client veut s'inscrire.	On calcule les coordonnées GPS du client à partir de son adresse à l'aide de la classe utilitaire GeoTest puis on persiste le client dans la base de données.	S : le client reçoit un mail de confirmation de l'inscription. E : le client reçoit un mail lui informant de l'échec de l'opération d'inscription. Il est invité à renouveler son inscription ultérieurement.	Objet de type Client.	void
Le client se connecte.	A partir des paramètres fournis, on recherche dans la base de données le client correspondant à l'aide d'une requête JPQL.	S : l'objet Client correspondant est renvoyé par le service. E : un objet null est renvoyé par le service.	String adresseMail, String motDePasse	Objet de type Client
Le client fait une demande d'intervention.	On calcule et enregistre la date au moment de la création de la demande. On recherche l'employé disponible (pas d'intervention en cours) le plus proche géographiquement du client à l'aide des coordonnées GPS et si la demande a été émise pendant ses horaires de travail. La demande d'intervention passe au statut "EN_COURS".	S : un employé correspondant aux critères a été trouvé. Il est alors associé à cette demande d'intervention et il devient non disponible. Cet employé est notifié par message de la demande d'intervention. E : Aucun employé est disponible (soit tous en train de faire une intervention ou aucun disponible pendant ces horaires). La demande d'intervention devient alors "REJETEE". Le client est notifié par message et invité à réessayer ultérieurement.	Objet de type Client et objet de type DemandeInterventionAnimal, DemandeInterventionIncident ou DemandeInterventionLivraison	void
Le client veut consulter son historique de demande d'intervention.	On récupère toutes les demande d'intervention qui sont liées aux clients par une requête JPQL.	S : la liste des demandes d'intervention correspondante est renvoyée. E : null est renvoyé	Objet de type Client	Liste d'objets de type DemandeIntervention

b. Services liés aux employés

Intention	Action	Succès/Echec	Paramètre	Réponse
L'employé se connecte.	A partir des paramètres fournis, on recherche dans la base de données l'employé correspondant à l'aide d'une requête JPQL.	S : l'objet Employé correspondant est renvoyé par le service. E : un objet null est renvoyé par le service.	String adresseMail, String motDePasse	Objet de type Employé
L'employé veut clore son intervention.	La demande d'intervention associée à l'employé connecté (fourni en paramètre) voit son statut modifié à "FINIE_SUCCES" ou "FINIE_ECHEC" et la description de l'intervention par l'employé est associée à cette demande. Le client reçoit ensuite une notification par mail.	S : le client est informé par mail de la réussite de l'intervention accompagnée d'une description de l'employé ayant réalisé cette intervention. E : le client est informé par mail de l'échec de l'intervention et est invité à contacter Proact'if pour plus d'informations.	Objet de type Employe, boolean succes, String descriptionEmp	void
L'employé veut consulter l'intervention en cours.	Recherche et renvoie l'intervention associée à l'employé connecté (en paramètre) par une requête JPQL vers la base de données.	S : l'intervention courante est renvoyée E : un objet null est renvoyé	Objet de type Employe	Objet de type DemandeIntervention
L'employé veut consulter le tableau de bord affichant toutes les interventions du jour	Recherche toutes les demandes d'intervention du jour (finies ou non) par une requête JPQL.	S : une liste de demande d'intervention est renvoyée E : une liste null est renvoyée	void	Liste d'objets de type DemandeIntervention

V- Conclusion

En conclusion, un bon cahier des charges est crucial au développement d'une application d'un système d'information. En effet, chaque fonctionnalité doit être personnalisée en fonction d'un besoin spécifique de l'entreprise. Dans ce compte-rendu, nous définissons les besoins à partir d'un diagramme des cas d'utilisation, les objets métiers dans un diagramme de classe et puis nous avons traduit ces besoins en fonctionnalités dans les spécifications de service et différentes IHM réalisées.

Finalement, nous espérons que l'équipe qui sera chargée de réaliser la seconde partie de ce projet sera attentive et appliquée afin de permettre la création d'une application qui pourrait se révéler très utile.