

Polytech Nice-Sophia
Département des Sciences Informatiques
930 Route des Colles
06410 Biot
Téléphone : 04 92 96 50 50
Site web : www.polytechnice.fr

Rapport d'architecture

IsYolo 3000

Dimanche 21 février 2016

Enseignants référents :
Anne-Marie Pinna-Dery
Sébastien Mosser

Groupe ISA-BG

Alexandre Cazala
Tom Dall'Agnol
Pierre Massanes

Table des matières

Spécifications.....	2
Contexte.....	2
Scénarios.....	2
Scénario 1 - Permettre la vente de forfait et le contrôle aux remontées mécaniques.....	2
Scénario 2 - Rediriger les skieurs.....	2
Scénario 3 - Accès aux statistique d'afflux de skieur pour les commerçants.....	2
Scénario 4 - Service de notifications.....	3
Scénario 5 - Indiquer l'état des pistes et des remontées mécaniques.....	3
Scénario 6 - Adapter la stratégie marketing de la vente de forfait.....	3
Scénario 7 - Contraintes techniques.....	3
Vue Fonctionnelle.....	4
Cas d'utilisation.....	4
Diagrammes de Composants.....	9
Diagramme général.....	9
Diagramme de StationDeSki.....	10
Vue Développement.....	12
Diagramme de classe des objets métiers.....	12
Modèle relationnel de stockage.....	13
Explicitation du mapping objet - relationnel.....	14
Vue Déploiement	15
Exemples de scénarios d'utilisation	16
Achat d'un forfait	16
Abonnement aux données	16

I. Spécifications

1. Contexte

Suite à la situation d'enneigement de ces dernières années, la “*Société des Cimes*” (SdC) a lancé un [appel d'offre](#) dans le but de moderniser leur système d'informations afin de proposer des services toujours plus innovant à leurs clients.

2. Scénarios

De cet appel d'offre, nous pouvons en déduire les scénarios suivants correspondant aux fonctionnalités principales désirées par la SdC.

Scénario 1 - Permettre la vente de forfait et le contrôle aux remontées mécaniques de manière automatique

- Choix d'une offre
 - forfait
 - pack de forfait
 - recommandation de la SdC
- Paiement de forfait
 - en ligne (retrait de la carte CIME en station si non possédée)
 - vente en station directement
- Retrait de la carte CIME en station
- Passage de la carte CIME au portique
 - Si forfait “fidel’cimes”, retirer l'argent directement à l'utilisateur lors du premier passage.
- Ces actions sont effectuées par un client.

Scénario 2 - Rediriger les skieurs

- Visualisation de la fréquentation des remontées mécaniques depuis le poste de contrôle
- Accès aux écrans de la station afin de donner des directions pour rediriger l'afflux de visiteur
- Signalisation des pistes et remontées ouvertes et fermées
- Ces actions sont effectuées par un administrateur.

Scénario 3 - Accès aux statistique d'afflux de skieur pour les commerçants

- Création d'un compte commerçant
- Paiement d'un abonnement pour accéder aux données
- Accès au nombre de forfaits activés aujourd'hui. (Exemple : Nombre de forfaits activés aujourd'hui.)
- Accès au nombre de forfaits réservés sur une période future. (Exemple : Nombre de forfaits réservés pour la semaine 9.)

- Accès au nombre de forfait activé sur une période passée. (Exemple : en 2015 il y a eût 9.000.000 de forfaits activés. Et pendant la semaine 12 de cette année il y en a eût 10.000.)
- Ces actions sont effectuées par un commerçant de la station.

Scénario 4 - Service de notifications

- Sélection d'un sous-ensemble d'utilisateur
- Sélection de la méthode de notification (e-mail, sms)
- Envoi de la notification à ce sous-ensemble qui ont fourni les informations de contact suffisantes
- Envoi d'un message SMS automatiquement alertant les utilisateurs premium (possédant un forfait Fidéli'cime ou forfait saison) lors de chute de neige poudreuse repérées par la station météo.
- Les sélections et l'envoi de la notification sont effectués par un administrateur

Scénario 5 - Indiquer l'état des pistes et des remontées mécaniques

- Modification de l'état des pistes (ouverte/fermée, damée/non damée)
- Modification de l'état des remontées mécaniques (ouverte/fermée)
- Notification aux panneaux fixes le changement d'état des pistes et/ou des remontées
- Notification au bulletin HTML le changement d'état des pistes et/ou des remontées
- Les modifications sont faites par les pisteurs

Scénario 6 - Adapter la stratégie marketing de la vente de forfait

- Accès aux statistiques de ventes de chaque forfait et pack (quantité et date de vente)
- Changement des prix des forfaits
- Ajout de pack de forfait
- Ajout de type de forfait
- Ajout de recommandation pour les clients
- Ces actions sont effectuées par un webmaster

Scénario 7 - Contraintes techniques

- La base de données utilisée pour faire persister les données appartient à la famille SQL (implémentation HSQL par défaut);
- Le système d'information utilise principalement la technologie J2E pour la définition des composants fonctionnels et la persistance des données;
- Les services annexes (e.g., paiement, envoi de SMS) sont implémenté en .Net avec la version libre Mono;
- Les services J2E et .Net sont déployé sur deux serveurs différents (communication par web service entre les deux). Une interface en ligne de commande est disponible comme un troisième projet pour communiquer avec le système.
- Les panneaux d'affichage sont de simples pages HTML obtenue via JSF.

II. Vue Fonctionnelle

1. Cas d'utilisation

Nous avons modélisé ces scénarios sous forme de diagrammes de cas d'utilisation afin d'obtenir une première vision du comportement fonctionnel de notre application. Nous avons divisé notre système d'informations IsYolo3000 en plusieurs diagrammes pour des critères de lisibilité. Ainsi les fonctionnalités importantes sont représentées dans le diagramme de cas d'utilisation de haut niveau (Figure 1) et certaines d'entre elles seront détaillées dans d'autres cas d'utilisation dans la suite du rapport.

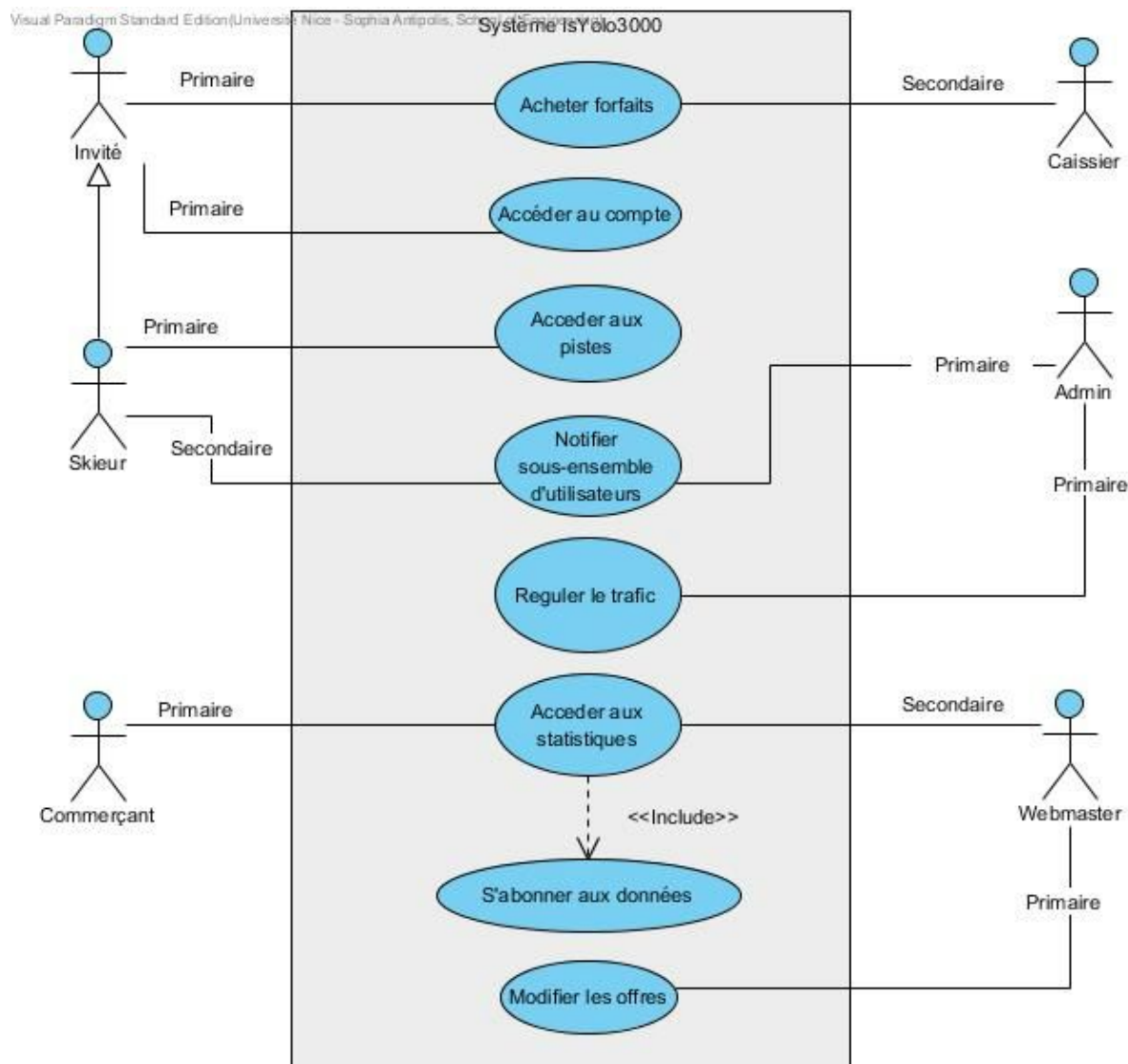


Figure 1 - Cas d'utilisation de haut niveau

Ce diagramme de cas d'utilisation de haut niveau du système IsYolo3000 (Figure 1) représente les principaux cas d'utilisation et les acteurs qui y sont associés. Les cas

d'utilisation les plus importants sont détaillés à la suite de ce schéma. Les rôles de chacun des acteurs sont les suivants :

- Le **Commerçant** peut s'abonner aux données fournies par le système (Figure 6) pour pouvoir accéder aux statistiques. Les statistiques dont dispose le Commerçant portent principalement sur l'afflux de personnes ce qui lui permet d'adapter la capacité d'accueil de son commerce.
 - Le **Webmaster** peut lui aussi accéder aux statistiques mais sans passer par un abonnement. Les statistiques qui l'intéressent portent sur le nombre de forfaits pour une période donnée ce qui lui permet de connaître quel forfait est le plus acheté à une période donnée. Son rôle est alors de modifier les offres proposées en fonction de ces données pour adapter sa stratégie marketing ;
 - **L'Administrateur** a pour rôle la régulation du trafic sur les pistes (Figure 4) et peut initier la notification d'information aux différents utilisateurs visés (Figure 5) ;
 - Le **Caissier** intervient lors de l'achat de forfait en caisse (Figure 2) pour faciliter l'achat d'une carte CIME ou de forfaits ;
 - **L'Invité** est l'utilisateur pouvant acheter un forfait (Figure 2) et accéder à son compte (Figure 3). Une fois que l'Invité a acheté un forfait il change de statut pour devenir Skieur ;
 - Le **Skieur** est donc un Invité titulaire d'un forfait et peut alors accéder aux pistes. Accéder aux pistes signifie que le Skieur peut passer les portiques menant aux pistes en présentant sa carte CIME liée à un forfait activé. Il existe un cas particulier pour accéder aux pistes, dans le cas où le Skieur possède un forfait -12 ans, le passage au portique enclenche un signal sonore indiquant au perchman qu'un enfant accède aux remontées mécaniques.
- Un Skieur peut aussi être notifié par le système via différents moyens de communication (Figure 5).

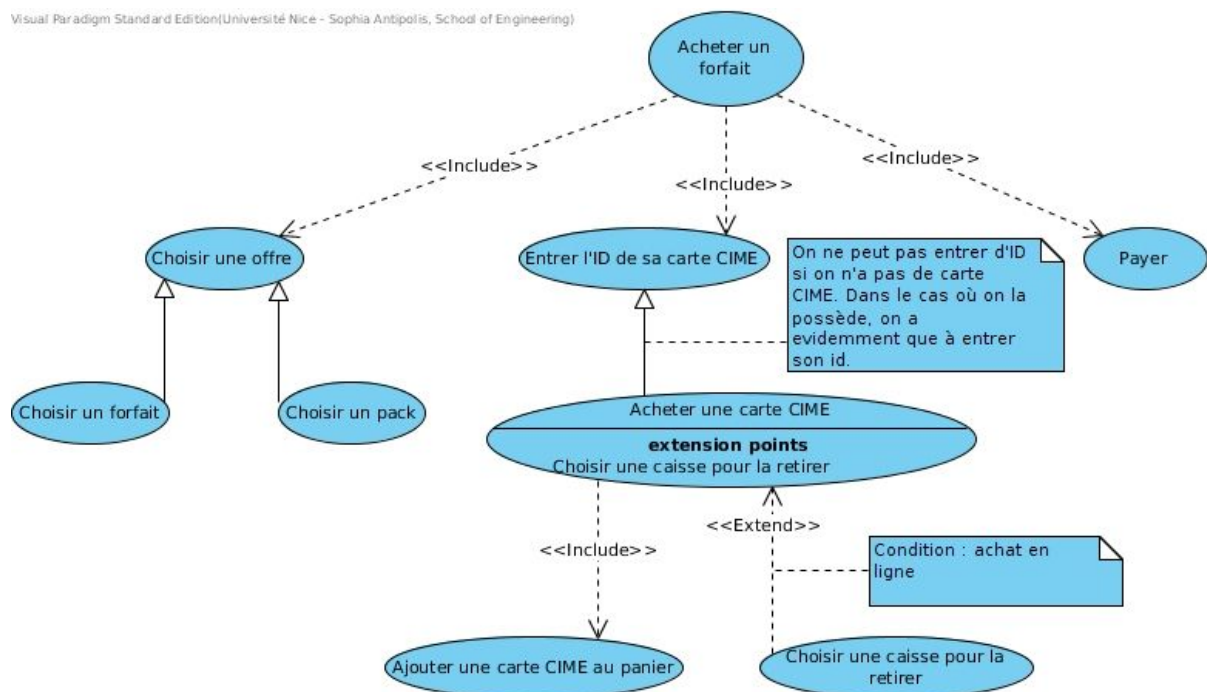


Figure 2 - Cas d'utilisation d'Acheter un forfait

Le diagramme ci-dessus reprend les actions à effectuer pour pouvoir obtenir un forfait à savoir :

- Choisir une offre, si l'utilisateur préfère acheter un seul forfait (il doit choisir lequel) ou alors s'il préfère choisir une offre économique (un pack) convenant à ses besoins (acheter plusieurs forfaits par exemple, ou obtenir la location des ski en même temps (à voir si la SdC propose ce genre de pack donc)).
- Il doit spécifier l'identifiant de sa carte CIME qu'il soit en ligne ou non. Dans le cas où il n'en a pas, il devra en acheter une et s'il est en ligne il devra choisir une caisse pour venir récupérer sa carte.
- Enfin, ce cas d'utilisation comprend aussi le paiement du forfait choisi.

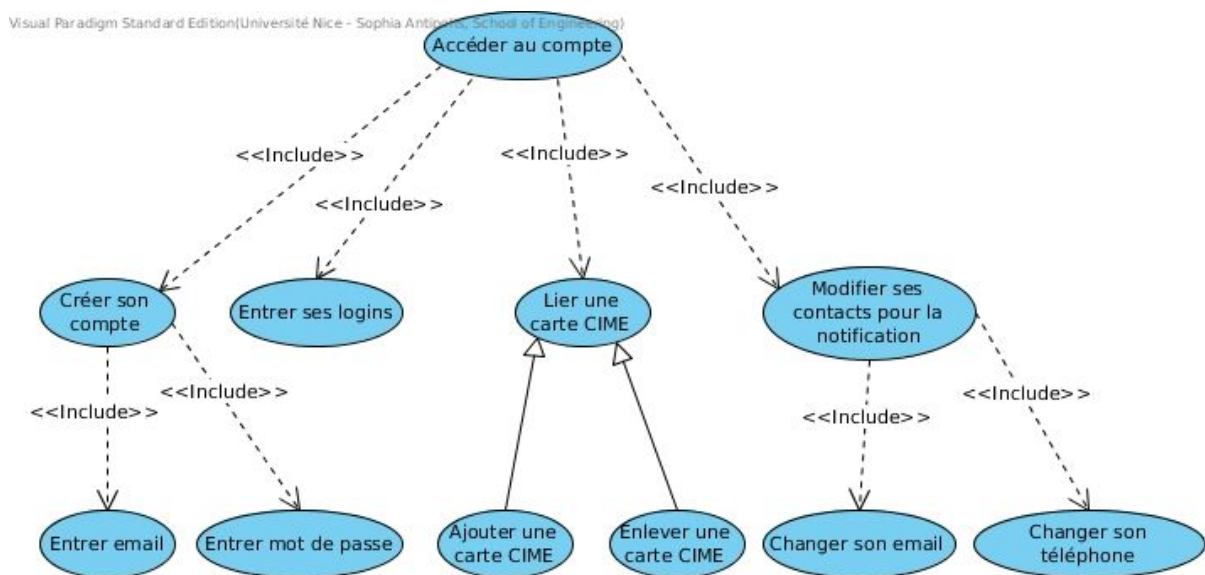


Figure 3 : Diagramme de cas d'utilisation de Accéder au compte

Concernant l'accès au compte (représenté par la Figure 3), diverses fonctionnalités lui sont incluses telles que :

- La création du compte imposant la saisie d'un mot de passe pour protéger le compte et d'une adresse email pour identifier le compte.
- L'accès au compte via ses logins choisis (email et mot de passe).

Plusieurs opérations peuvent être réalisées sur le compte une fois loggé ::

- Gérer ses cartes CIME
 - Associer une carte CIME à son compte
 - Retirer une carte CIME de son compte
- Modifier ses informations pour pouvoir être informé par l'administrateur de diverses informations. Dans le cas d'un membre premium, l'utilisateur pourra être informé par l'administrateur mais aussi en temps réel par la station météo dans le cas de chute de neige poudreuse.

Visual Paradigm Standard Edition (Université Nice - Sophia Antipolis, School of Engineering)

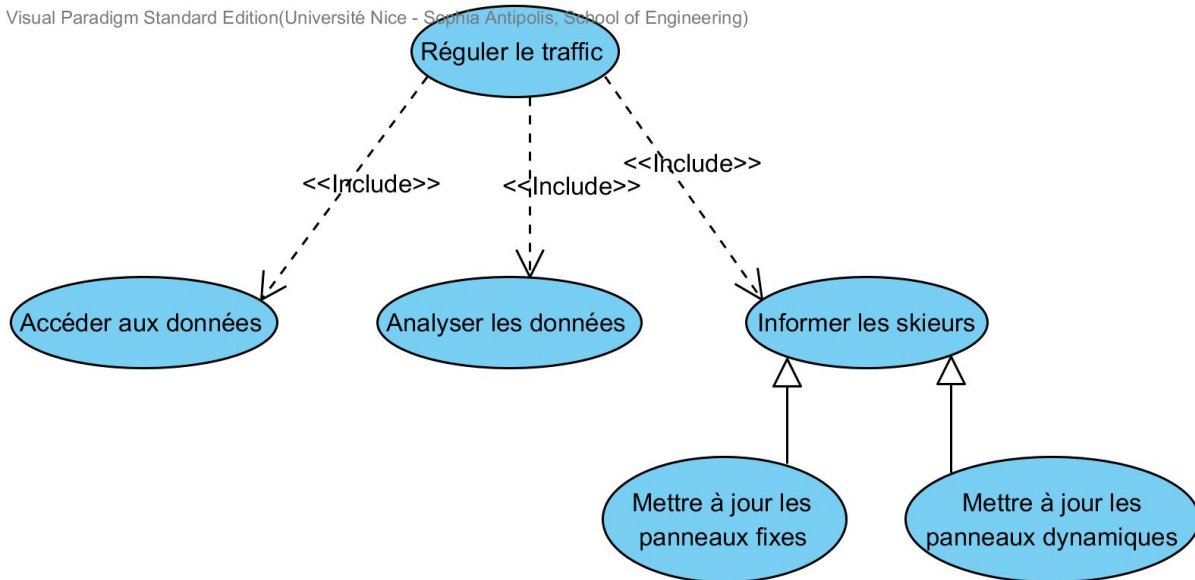


Figure 4 : Diagramme de cas d'utilisation de Réguler le trafic

Ce cas d'utilisation initié par l'Administrateur met en évidence la régulation du trafic de skieurs dans la station. Pour réguler le trafic il est d'abord nécessaire d'accéder aux données de la station pour connaître le nombre de skieurs présents sur chaque piste. Il faut alors analyser cette information pour pouvoir décider s'il faut informer les skieurs et comment. Pour informer les skieurs il existe plusieurs méthodes :

- Mettre à jour un ou plusieurs panneaux dynamiques affichant un message recommandant un itinéraire aux skieurs ;
- Mettre à jour les panneaux fixes dont le rôle est de donner l'état du domaine (quelle piste est ouverte et quelle piste est fermée par exemple).

Visual Paradigm Standard Edition (Université Nice - Sophia Antipolis, School of Engineering)

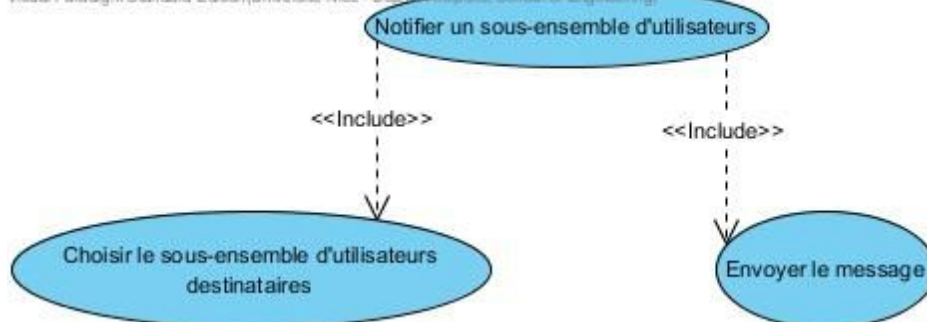


Figure 5 : Diagramme de cas d'utilisation de Notifier un sous-ensemble d'utilisateurs

Le système inclut un dispositif de notification des usagers initié par l'Administrateur. Le système de notification permet d'informer un groupe d'utilisateurs précis pour leur fournir différentes informations (proposition de réductions, conditions météo, etc). Pour effectuer cette action il faut d'abord choisir un sous-ensemble d'usagers à contacter (selon le forfait, l'âge, etc) et ensuite envoyer le message selon le moyen de communication souhaité (par exemple par SMS si la personne est au sein de la station, par mail si ce n'est pas le cas).

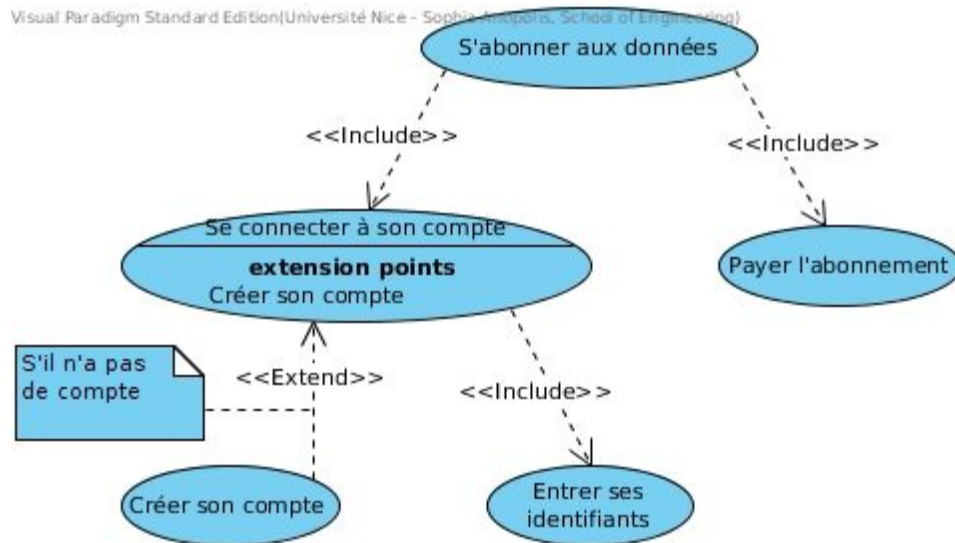


Figure 6 : Diagramme de cas d'utilisation de S'abonner aux données

Concernant le cas d'utilisation de s'abonner aux données, celui-ci concerne les commerçants. Les commerçants peuvent, moyennant finance, s'inscrire dans la liste des comptes commerçants pouvant lire certaines données pour pouvoir agir en conséquences de celles-ci. Par exemple, en cas de forte affluence dans le domaine, le commerçant peut juger préférable de réduire le nombre d'articles proposés pour ainsi libérer un peu de place dans son magasin et y accueillir plus de monde générant donc plus de profit pour celui-ci.

Pour cela, il faut donc d'abord qu'il se connecte à son compte (ou qu'il en crée un s'il n'en dispose pas). Enfin après avoir payé la cotisation, il disposera des droits de lecture sur ces données (il ne pourra en aucun cas les modifier).

2. Diagrammes de Composants

1. Diagramme général

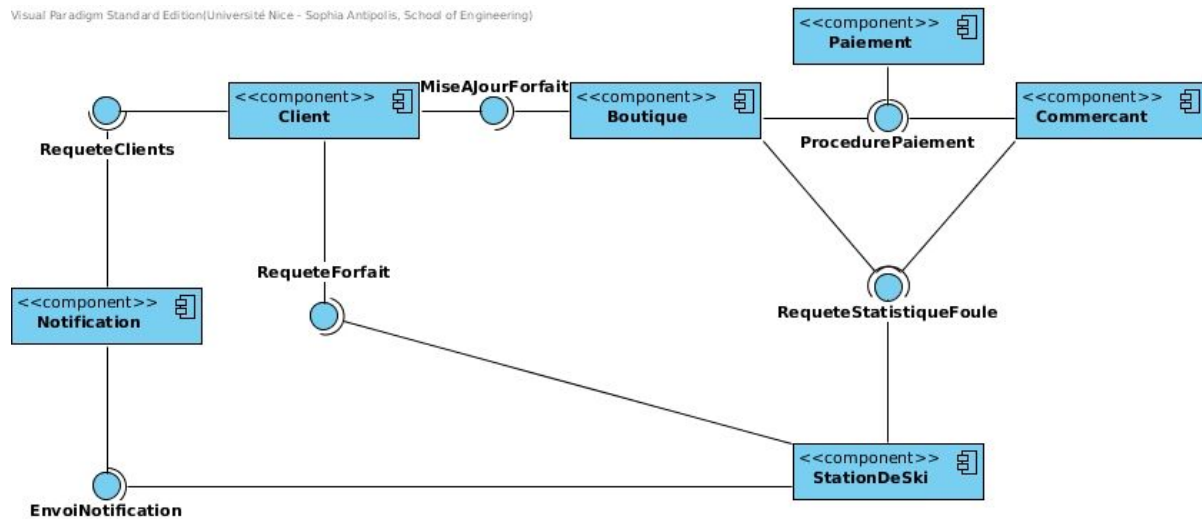


Figure 7 : Diagramme de composant général .Feat Père Noël

Le diagramme de composant ci-dessus (Fig.7) correspond à l'architecture générale de notre application.

Le composant **Client** contient tout ce qui a à voir avec les futurs skieurs ou skieurs actuels : compte personnel et connexion à celui-ci. Il expose 3 interfaces :

- *RequeteClients* permet de récupérer les contacts (téléphone/mail) des clients suivant des critères de sélection. Cela est utile pour pouvoir notifier des utilisateurs de diverses informations. Cette communication est faite par webservice.
- *RequeteForfait* permet à partir de l'identifiant de la carte Cime de retrouver le forfait associé.
- *MiseAJourForfait* permet de mettre à jour le forfait associé à une carte Cime.

Le composant **Boutique** représente la boutique en elle même ainsi que le webmaster qui s'occupera de la mettre à jour régulièrement afin d'adapter la stratégie de marketing d'après les statistiques de vente et d'affluence de la station.

Le composant **Paiement** est le module permettant de payer. Il sera développé en .Net et on communiquera avec par webservice (qui correspond à l'interface *ProcedurePaiement* ici).

Le composant **Commerçant** représente un commerçant de la station. Moyennant un abonnement, il a accès aux statistiques de fréquentation.

Le composant **StationDeSki** est le modèle de la station. Il sera détaillé plus loin (voir 2. Diagramme de StationDeSki). Il n'expose qu'une interface : *RequeteStatistiqueFoule* qui permet de récupérer les statistiques liées à la fréquentation d'utilisateur.

Le composant **Notification** s'occupe de notifier (par mail ou par sms par exemple) un groupe d'utilisateur défini selon des critères comme "Présents dans la station", "tranche d'âge" et "forfait en cours de validité" ainsi que leurs inverses. Il sera développé en .Net et on communiquera avec celui-ci par webservice (ici représenté par l'interface *EnvoiNotification*).

2. Diagramme de StationDeSki

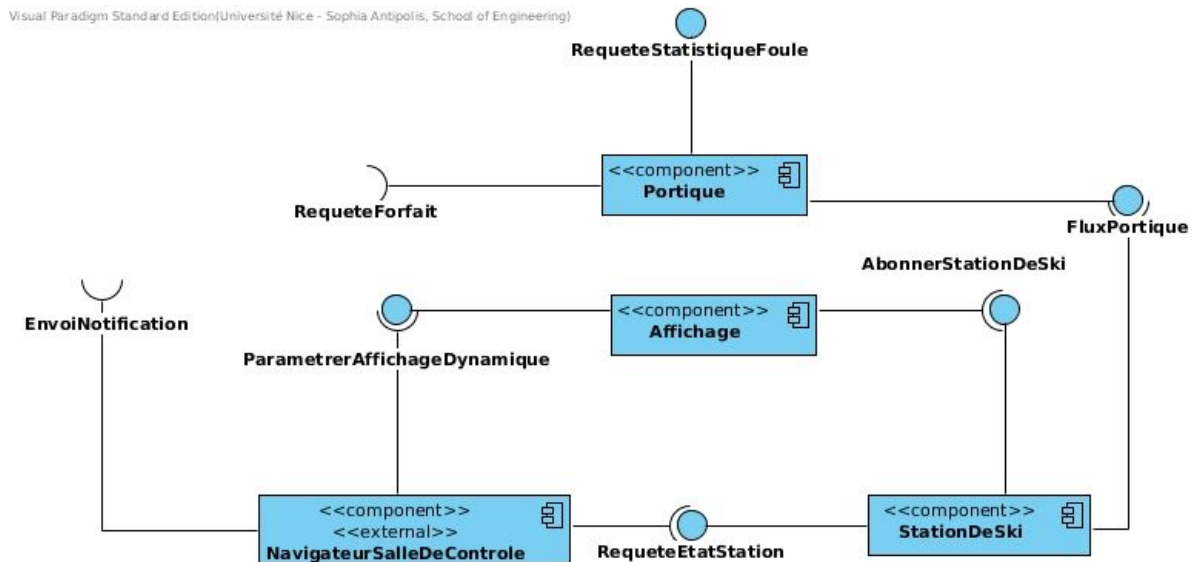


Figure 8 : Diagramme de composant de StationDeSki

Le composant **Affichage** s'occupe de l'affiche d'informations, que ce soit sur les panneaux d'affichages (fixes et dynamiques) que sur le bulletin des pistes en HTML. Il expose une interface : *ParametrerAffichageDynamique* qui permet de modifier le texte affiché sur un panneau d'affichage dynamique que l'on choisi.

Le composant **StationDeSki** est le modèle même de la station : pistes, remontées mécaniques, météo, tout ce qui concerne la station est ici. Ce sont les pisteurs qui modifieront l'état des pistes et des remontées mécaniques (ouvertes/fermées, damées/non damées), mais aussi les portiques afin de signaler le nombre de personne actuellement sur la remontée. Elle expose 2 interfaces :

- *RequeteEtatStation* qui permet d'avoir en temps réel l'état de la totalité de la Station de ski
- *AbonnerStationDeSki* qui permet de s'abonner à la station. Ainsi s'il y a une modification le bulletin HTML se remettra à jour et les panneaux fixes aussi. (On passe par un pattern Observer/Observable)

Le composant **Portique** est l'intelligence d'un portique sur lequel on va passer la carte CIME. Il va donc s'occuper de récupérer l'information de la carte, trouver le forfait associé, indiquer au portique de s'ouvrir ou non, indiquer au portique de sonner s'il y a un enfant de moins de 12 ans, marqué le forfait comme utilisé ce jour ci. Il expose 2 interfaces :

- *RequeteStatistiqueFoule* pour offrir la possibilité de récupérer les statistiques de passage sur une période donnée, mais aussi de le visualiser par type de forfait.
- *FluxPortique* pour obtenir en temps réel les passages de portique (et ainsi connaître le nombre de personne qui sont dans la remontée mécanique)

Le composant **NavigateurSalleDeContrôle** représente la personne se trouvant dans la salle de contrôle avec son logiciel pour gérer la station. A partir de là il visualise l'état de la station, choisi d'envoyer des notifications à un sous-ensemble d'utilisateurs, modifier le texte des affichages dynamiques.

III. Vue développement

1. Diagramme de classe des objets métiers



Figure 9 : Diagramme de classe des objets métiers

2. Modèle relationnel de stockage

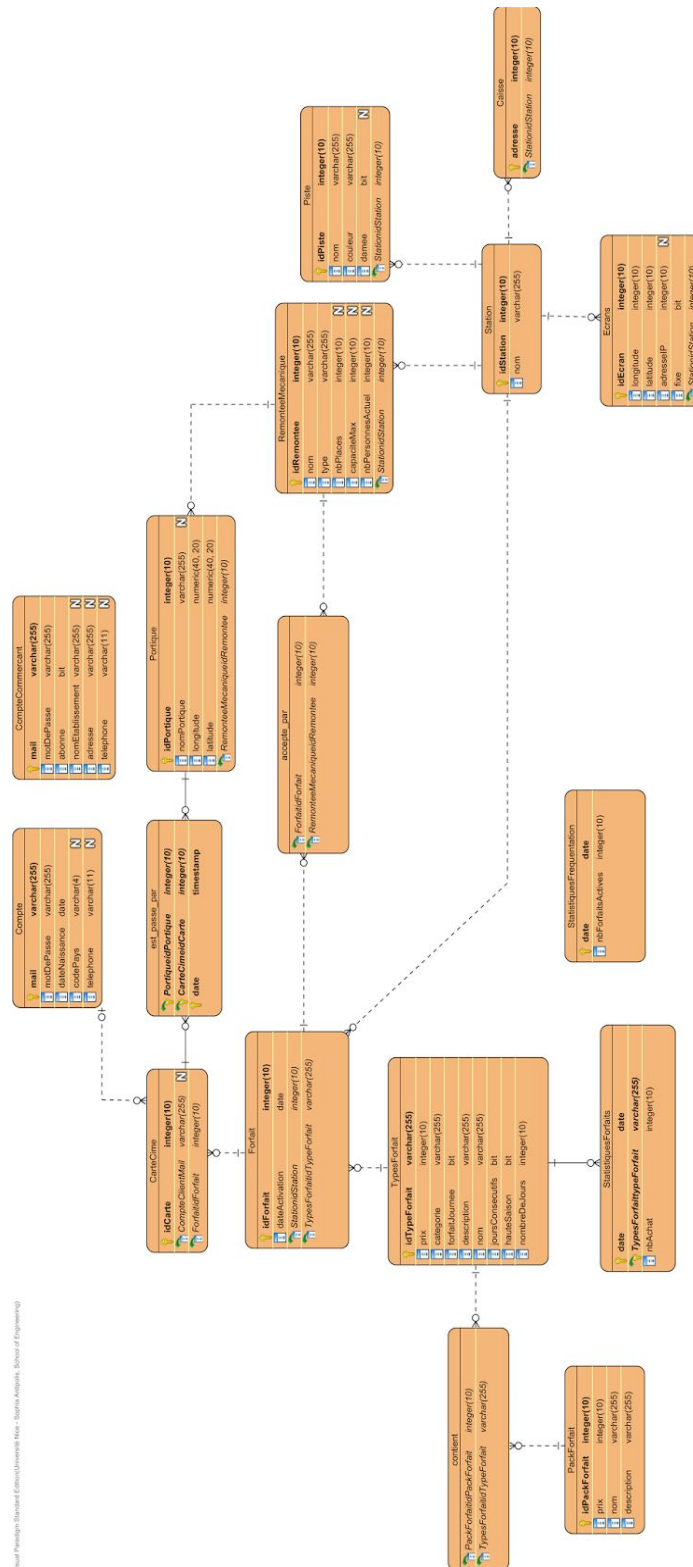


Figure 10 : Modèle relationnel de stockage

3. Explicitation du mapping objet - relationnel

Pour obtenir le modèle relationnel de stockage, nous sommes partis du diagramme de classe d'objets métiers puis nous avons ajouté des données à persister (les statistiques de fréquentation, les statistiques de vente).

Un forfait en fonction de son type ne permet pas forcément d'accéder à toutes les remontées mécaniques. Ainsi, nous avons besoin d'un lien entre **TypeForfait** et **RemonteMecanique** que nous avons choisi de représenter avec une table d'association (une remontée est accessible par plusieurs types de forfaits et un type de forfait permet d'accéder à plusieurs remontées différentes).

Nous avons deux types d'écrans : fixe et dynamique. Le contenu des deux est modifié en fonction du serveur :

- le panneau fixe ayant des DEL indiquant l'ouverture/fermeture des pistes/remontées mécanique sur lui nous avons représenté ça par des listes de booléen (pour "DEL allumée/éteinte").
- le panneaux dynamique affichant seulement un message texte il nous suffisait de le représenter avec un String.

Dans le modèle relationnel de stockage, nous avons transformé cet héritage en une simple colonne contenant un bit pour définir si le panneau était fixe ou non. Il n'y a pas besoin de stocker les lumières des panneaux fixes (car cela dépend de l'état de la station et pas du panneau en lui-même) et le texte du panneau dynamique (il change régulièrement et il n'y a pas d'intérêt de conserver un simple message informatif du type "==">").

Les deux types de comptes existant (**CompteCommercant** et **CompteClient**) n'ayant peu, voir rien à voir à part au niveau objet (ce sont tous les deux des comptes), nous avons décidé de faire deux tables séparées. Cela permet de ne pas les mélanger alors qu'ils n'ont pas accès aux mêmes choses. On pourrait ainsi les mettre sur deux base de données différentes pour optimiser les accès (et ne pas léser les skieurs tout comme les commerçant).

La boutique propose de vendre des **Pack** de forfaits. Ainsi un pack de forfait contient des forfaits et a un prix spécial. Nous avons choisi d'utiliser une table d'association pour cela car nous avons un pack qui est associé à plusieurs forfaits dont la taille n'est pas fixe.

Nous avons choisi de persister aussi les **StatistiqueDeVente** afin que le webmaster puisse y avoir accès et voir si sa nouvelle stratégie de marketing a fonctionné. Pour ça nous avons donc pour chaque type de forfait le nombre de vente qui ont été fait à une date précise. Cela permet de suivre l'évolution du nombre de vente en fonction du temps.

Nous utiliserons un pattern *Type-Object* pour les forfaits : nous aurons une classe **TypeForfait** et une classe **Forfait**. Nous avons fait ce choix afin que le webmaster qui s'occupe de la boutique puisse facilement faire de nouveaux types de forfaits via un éditeur de texte par exemple sans avoir à coder et hériter d'une classe abstraite "Forfait" : la taille

du système n'aurait pas cessé de grandir sans qu'il n'y ait aucun sens et on aurait dû faire une mise à jour du système dès que l'on souhaitait ajouter un autre type de forfait.

IV. Vue Déploiement

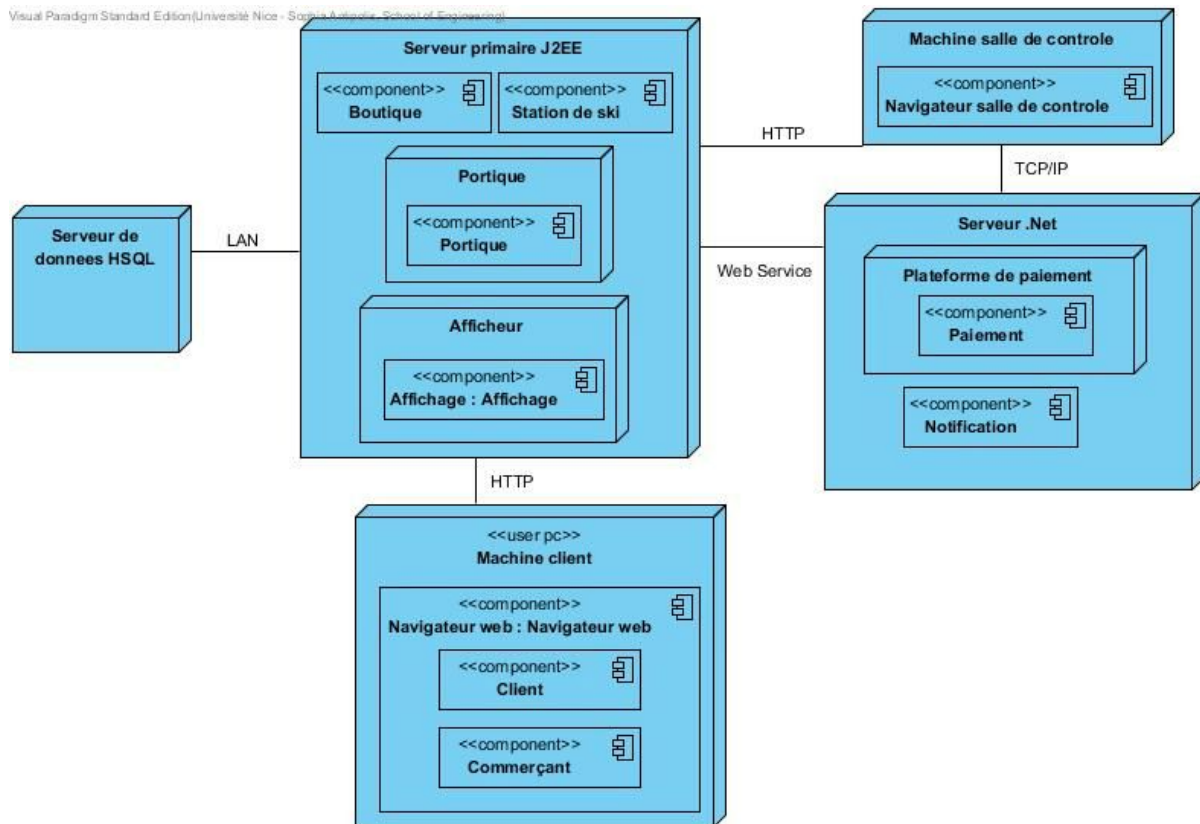


Figure 11 : Diagramme de déploiement du système IsYolo3000

Physiquement nous disposons de 5 entités.

Nos données sont stockées sur notre serveur de données HSQL. Ce serveur communique avec notre serveur primaire en J2EE en LAN.

Sur ce même serveur primaire, nous déploierons nos composants de Boutique et de station de ski tel que spécifié dans l'appel d'offre. Les Portiques et Afficheurs seront eux aussi déployés sur le serveur primaire.

Le contexte technologique détaillé dans l'appel d'offre imposait que les modules de notification et de paiement soient déployés sur le serveur en .Net. Enfin Le client quant à lui utilisera son navigateur web que ce soit un commerçant ou un client standard.

Exemples de scénarios d'utilisation

Au préalable, pour en faciliter la lecture (et l'écriture) nous établissons certaines règles de typographie :

- Les composants seront représentés en **gras**
- Les interfaces seront quant à elles représentées en *italique*

Achat d'un forfait

L'utilisateur se connecte à son compte client depuis son **NavigateurWeb** qui est donc un composant externe.

Pour pouvoir se procurer son forfait, on se connecte à **Client**. Puis via le navigateur, on transmet notre token de compte à la Boutique via son interface. Et enfin **Boutique** communiquera avec la plate-forme de **Paiement** via *ProcedurePaiement*. Une fois confirmé, la boutique mettra à jour les forfaits du compte associé au token envoyé.

La figure 12 ci-dessous représente ce qui vient d'être expliqué.

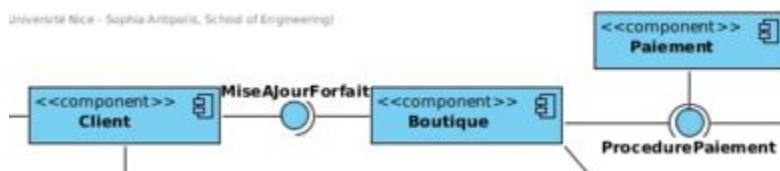


Figure 12 : Abstraction des composants inutiles à l'achat d'un forfait

Abonnement aux données

Le commerçant se connecte à son compte depuis son navigateur web. Ces actions s'effectuent dans le composant **Commerçant**.

Pour pouvoir accéder aux données le **Commerçant** doit passer par le module **Paiement** via l'interface *ProcedurePaiement* et ainsi acheter un abonnement. Une fois l'abonnement obtenu, le **Commerçant** peut librement accéder aux statistiques de fréquentation de la station grâce à l'interface *RequeteStatistiqueFoule*.