

Travail de bachelor 2010

## Filière Informatique de gestion

### Informations interactives sur mobile pour le tourisme des remontées mécaniques



Etudiant-e : Frédéric Aebi

Professeur : Yann Bocchi

# Table des matières

<b>1</b>	<b>PREFACE .....</b>	<b>1</b>
<b>2</b>	<b>DECOMPOSITION DES TACHES .....</b>	<b>2</b>
2.1	Analyse.....	2
	Etat de l'art.....	2
	Logiciels existants.....	2
	Points positifs/négatifs d'un développement sur iPhone/Android.....	2
	Définition des fonctionnalités .....	2
2.2	Développement .....	2
	Recherche.....	2
	Modélisation de la base de données.....	3
	Conception d'une interface administrateur et d'un service web.....	3
	Développement du logiciel .....	3
	Test de l'application.....	3
<b>3</b>	<b>ETAT DE L'ART.....</b>	<b>4</b>
3.1	Les logiciels existants .....	5
	Plateforme iPhone .....	5
	Plateforme Android.....	6
<b>4</b>	<b>CAHIER DES CHARGES INITIAL .....</b>	<b>8</b>
4.1	Fonctionnalités de base .....	8
	Affichage du domaine skiable .....	8
	Menu des points d'intérêt.....	9
4.2	Fonctionnalités avancées.....	9
	Réalité augmentée.....	9
4.3	Autres fonctionnalités .....	10
	Planification d'un itinéraire .....	10
	Contact par téléphone .....	10
	Fonctionnalité d'alerte .....	10
4.4	Planification du développement.....	10
<b>5</b>	<b>CHOIX DES TECHNOLOGIES ET SPECIFICITES TECHNIQUES .....</b>	<b>11</b>
5.1	Plateforme de développement.....	11
	Pénétration des Smartphones dans le marché .....	11
	Parts de marchés des Smartphones .....	12
	Prochain système d'exploitation désiré par rapport au système actuel.....	13
	Choix définitif de la plateforme de développement.....	13
5.2	Système de gestion de base de données (SGBD).....	14
5.3	Couche d'accès aux données.....	14
5.4	Utilisation des APIs Google.....	16
5.5	Utilisation de l'API Topocoding.....	16

<b>6</b>	<b>DESCRIPTIF DE L'APPLICATION .....</b>	<b>17</b>
6.1	Interface administrateur .....	17
	Ajout, modification et suppression des points d'intérêt.....	17
	Gestion des alertes .....	23
6.2	Application mobile.....	25
	Accueil .....	25
	La carte du domaine skiable .....	26
	Liste des points d'intérêt.....	27
	Alerte rapide.....	31
	Statut des remontées mécaniques et des pistes de ski.....	32
	Météo .....	32
	Aide.....	33
<b>7</b>	<b>ADAPTATION DU CAHIER DES CHARGES INITIAL .....</b>	<b>34</b>
7.1	Suppression de la fonctionnalité de réalité augmentée .....	34
7.2	Amélioration de la fonctionnalité d'alerte .....	34
7.3	Remplacement de la planification d'un itinéraire par le statut des points d'intérêt... ..	35
7.4	Météo .....	35
<b>8</b>	<b>POTENTIEL DE L'APPLICATION .....</b>	<b>36</b>
8.1	iSummerSports.....	36
8.2	Utilisation possible dans le monde entier .....	36
8.3	Apport de l'application dans le milieu du tourisme .....	36
<b>9</b>	<b>EXTENSIONS POSSIBLES .....</b>	<b>37</b>
9.1	Globalisation de l'application .....	37
9.2	Planning et agenda des sauveteurs .....	37
9.3	Planification d'un itinéraire .....	38
9.4	Diversification des points d'intérêt .....	39
<b>10</b>	<b>CONCLUSION.....</b>	<b>40</b>
<b>11</b>	<b>REFERENCES.....</b>	<b>41</b>
<b>12</b>	<b>ANNEXES .....</b>	<b>42</b>
12.1	Accès aux ressources .....	42
12.2	Diagrammes .....	43
12.3	Tutorial sur le parsing d'un document XML en Objective-C pour iPhone .....	45
12.4	Icônes de l'application .....	49
12.5	Procès-verbaux.....	50
	Procès-verbal #1.....	50
	Procès-verbal #2.....	51
	Procès-verbal #3.....	52
	Procès-verbal #4.....	53
	Procès-verbal #5.....	54
	Procès-verbal #6.....	55
	Procès-verbal #7.....	56

<b>13</b>	<b>TABLE DES ILLUSTRATIONS.....</b>	<b>57</b>
<b>14</b>	<b>BIBLIOGRAPHIE .....</b>	<b>59</b>
	14.1 Bibliographie.....	59
	14.2 Sites Internet .....	59
<b>15</b>	<b>ACCORD DE CONFIDENTIALITE .....</b>	<b>60</b>

# 1 Préface

L'offre actuelle de services mobiles aux utilisateurs des remontées mécaniques des diverses stations valaisannes n'a de loin pas atteint son apogée. L'objectif de ce travail de bachelor est de combler une partie de ce manque en fournissant diverses informations à destination des Smartphones des usagés des pistes de ski.

Imaginez-vous sur les pistes de ski d'une magnifique station de nos Alpes Suisses. Le ciel est dégagé, le soleil brille et la motivation de pratiquer votre sport de neige préféré est à son comble. Vous vous trouvez peut-être à un endroit qui vous paraît inconnu. Vous souhaiteriez connaître les détails de l'emplacement auquel vous vous trouvez actuellement. Quel est le degré de difficulté de la piste ? Quel est son dénivelé ? Quelles sont les possibilités qui s'offrent à vous au niveau des remontées mécaniques ? Est-ce que toutes les pistes sont ouvertes ? Y a-t-il un restaurant dans les environs ? Quelle sera la météo demain ?

Lorsqu'une personne se rend sur les pistes de ski, elle s'expose à des risques. En cas d'accident, il est important que les opérations de sauvetage aient lieu le plus rapidement possible. Or, la sécurité sur les pistes de ski n'est pas vraiment ce qui préoccupe le plus les pratiquants de sports de neige et souvent, ils ne savent pas comment agir dans ces situations. Appeler les secours est une chose, mais expliquer l'emplacement de l'accident au personnel de sauvetage est très compliqué pour un touriste ne connaissant pas le domaine skiable. Il serait donc intéressant que les secours puissent localiser l'appel téléphonique provenant du touriste. Ceci leur permettrait de se rendre plus rapidement sur les lieux de l'accident et ainsi de minimiser la durée de l'opération de sauvetage.

Tous ces services sont regroupés sur une application iPhone baptisée iSnowSports et développée durant les trois mois de ce travail de bachelor. Je vous souhaite d'ores et déjà beaucoup de plaisir lors de la lecture de ce document.

## 2 Décomposition des tâches

Suite aux deux premières séances (voir chap. 12.5 - Procès-verbaux) entre les différentes parties impliquées dans ce travail de bachelor, un cahier des charges bien précis doit être défini. Les tâches à réaliser se regroupent en deux grandes parties principales : l'analyse et le développement.

### 2.1 Analyse

#### Etat de l'art

Il est important d'analyser la situation actuelle. Est-ce que le logiciel que je prévois de développer répond à un réel besoin ? Quel est le nombre de personnes qui pourraient être intéressées par une telle application ? Comment est-ce que les touristes s'orientent et se retrouvent actuellement dans un domaine skiable ?

#### Logiciels existants

Quels sont les logiciels qui existent sur le marché d'aujourd'hui et qui contiennent des fonctionnalités similaires au produit que je désire développer ? Quels sont leurs points forts/faibles ? Il s'agit ici d'étudier ce qui a déjà été réalisé, afin d'être en mesure de prendre plus facilement des décisions par rapport aux fonctionnalités à implémenter.

#### Points positifs/négatifs d'un développement sur iPhone/Android

Le choix de la plateforme de développement est très important. Ce dernier se fera entre un développement sur iPhone ou sur Android. Il s'agit donc d'analyser le marché des Smartphones ainsi que les différentes possibilités qu'offrent ces derniers au niveau de la conception d'un tel logiciel. Il se peut que l'une ou l'autre plateforme ne permette pas le développement de certaines fonctionnalités. Il est également possible que des implémentations fonctionnent mieux sur iPhone que sur Android ou inversement.

#### Définition des fonctionnalités

Lorsque les trois premiers points de la partie « Analyse » seront traités, les fonctionnalités du logiciel pourront être définies. Grâce à ces informations, le développement pourra être organisé de manière adéquate.

### 2.2 Développement

Le développement de l'application se découpe en cinq parties : la recherche, la modélisation de la base de données, la conception d'une interface administrateur et d'un service web, le développement du logiciel et, finalement, le test de l'application.

#### Recherche

Que le développement se fasse sur iPhone ou sur Android, il faut tout d'abord s'informer sur la technologie en question.

Si l'idée d'un développement sur iPhone est retenue, il s'agira de se familiariser avec le langage de programmation Objective-C. Ce langage ne faisant pas partie intégrante des cours proposés par la filière informatique de gestion de la HES-SO Valais, des tutoriaux devront être suivis afin de concevoir le logiciel de manière à respecter la philosophie de ce langage.

Au contraire, si l'application est développée sur Android, la phase de préparation et de recherche sera certainement un peu moins exigeante, car le langage de programmation est Java. Ce dernier a été étudié en long et en large durant les trois années de bachelor HES. Il faudra tout de même s'acclimater avec les libraires nécessaires au développement mobile via le langage Java.

### **Modélisation de la base de données**

Suite aux décisions prises lors des différentes séances, il n'y aura pas de partenariat avec une station de sport de neiges (voir chap. 12.5 - Procès-verbaux). C'est-à-dire qu'il ne sera pas possible de pouvoir bénéficier d'une base de données existante pour la conception du logiciel.

Il faudra donc investir du temps pour réfléchir à tous les acteurs impliqués ainsi qu'aux différents cas d'utilisation de l'application. Un modèle physique de données doit également être réalisés. Ces diagrammes sont disponibles en annexe de ce document au chapitre 12.2 - Diagrammes.

Lorsque tous ces modèles auront été validés, la base de données pourra être déployée.

### **Conception d'une interface administrateur et d'un service web**

Afin de simplifier l'ajout, la modification et la suppression des données, une simple interface administrateur devra être conçue.

Ceci est indispensable, suite au concept de géolocalisation des différents points d'intérêt de l'application (pistes, remontées mécaniques et restaurants). La saisie de ces derniers dans la base de données sera simplifiée à travers les API (Application Programming Interface) de Google et de Topocoding. A travers ces APIs, un simple clic sur la carte du monde permet de retrouver les coordonnées géographiques ainsi que l'altitude du point sélectionné.

Lorsque toutes les données seront saisies dans la base, un service web pourra être développé afin de rendre la communication avec l'interface mobile plus simple.

### **Développement du logiciel**

La phase de développement du logiciel représente certainement la partie la plus volumineuse de ce travail de bachelor. Il s'agit ici de développer les fonctionnalités définies et validées durant la phase d'analyse.

### **Test de l'application**

Une phase de test de l'application doit également être planifiée. Ceci afin de s'assurer d'éliminer tous les éventuels bugs qui pourraient être rencontrés lors de l'utilisation de l'application.

### 3 Etat de l'art

En ce jour, il existe déjà différents moyens de s'orienter sur des pistes de ski. Les stations sont équipées de panneaux nous indiquant la destination de chaque piste, ainsi que son degré de difficulté (piste verte, bleue, rouge, noire). Mais comment s'orienter sur une piste de ski en cas de mauvaise visibilité. Il se peut que nous passions complètement à côté d'un panneau indicateur et que nous rations la piste nous menant vers l'endroit désiré.

Chaque piste se doit d'être balisée de manière à pouvoir se situer en cas de brouillard. Ainsi, le balisage des pistes est différent du côté gauche que du côté droit. Grâce à ce système, le skieur peut constamment savoir s'il se trouve à gauche ou à droite de la piste et peut skier entre ces balises en toute sécurité même en cas d'intempéries et de mauvaise visibilité.

Ces informations sont très importantes pour chaque skieur. Or, ces derniers ne se préoccupent pas assez de ce genre d'informations et un grand pourcentage des skieurs ne connaît pas la signification de toute ces balises, panneaux, etc.

De même, il faut s'imaginer les pratiquants de sports de neige comme des voitures se déplaçant d'un point A vers un point B. Les GPS sont aujourd'hui équipés de manière à reconnaître les embouteillages sur les autoroutes etc. Ainsi, un automobiliste va peut-être choisir de prendre une route principale au lieu de rester sur l'autoroute en question et grâce à cela, le trafic sur le tronçon embouteillé pourra être rétabli plus rapidement.

Le même scénario pourrait avoir lieu sur les pistes de skis. Imaginez un skieur se trouvant sur une piste avec différentes possibilités au niveau des remontées mécaniques. Idéalement, il devrait pouvoir être mis au courant des temps d'attentes actuels de ces dernières et en fonction de ceux-ci, pouvoir opter pour une solution ou une autre.

A travers ce service, le temps d'attente auprès des différentes remontées mécaniques pourrait être réduit, l'exploitation de ces dernières optimisée et la satisfaction de la clientèle augmentée. Ce service serait disponible sous forme d'une application mobile permettant la géolocalisation du skieur et la mise à disposition d'informations par rapport au lieu actuel de ce dernier.

#### **Récapitulatif**

- ❖ *Les panneaux de signalisations sont très utiles en cas de beau temps, mais comment se situer en cas de brouillard ?*
- ❖ *Comment rendre les pratiquants de sports de neige sensibles au balisage des pistes de ski, ainsi qu'à sa signification ?*
- ❖ *Comment peut-on réduire le temps d'attente auprès des différentes remontées mécaniques ?*

## 3.1 Les logiciels existants

Il existe différents logiciels sur le marché qui contiennent déjà quelques fonctionnalités relatives aux besoins cités ci-dessus.

### Plateforme iPhone

#### Mobile Ski<sup>1</sup>

Cette application permet d'avoir accès aux cartes détaillées des stations de ski, y compris aux informations concernant les sentiers, les pavillons, les magasins de location de ski ou encore les écoles de ski. Elle permet de facilement trouver son chemin dans le dédale des sentiers inconnus. Elle permet d'enregistrer et d'afficher son parcours, de prendre des photos et des vidéos en cours de route et de mesurer ses performances. Par l'intermédiaire d'email, de Facebook TM ou Twitter TM, cette application permet également d'envoyer des commentaires et des photos à des amis.

Figure 1 : Mobile Ski



Source : Navionis - mai 2010

#### iTrailMap<sup>2</sup>

iTrailMap 3D combine des cartes de sentiers traditionnels avec des cartes interactives en 3D. L'application offre des cartes détaillées de l'ensemble du domaine skiable. L'application permet la localisation du skieur sur la montagne, l'enregistrement de son parcours et le téléchargement de ce dernier sur Internet afin d'en afficher le tracé dans Google Earth TM. Elle permet également d'afficher la distance parcourue.

Figure 2 : iTrailMap



Source : 3DSkiers - mai 2010

#### RealSki<sup>3</sup>

RealSki offre la possibilité d'accéder à différentes informations du domaine skiable (niveau et localisation des pistes, restaurants, activités spéciales) par le truchement de la réalité augmentée. Elle fournit aussi des informations météorologiques.

Figure 3 : RealSki



Source : RealSki - mai 2010

<sup>1</sup> Source : <http://www.navionics.com> - mars 2010

<sup>2</sup> Source : <http://www.3dskier.com/> - mars 2010

<sup>3</sup> Source : <http://www.rtp.com> - mai 2010

### SkiReport<sup>4</sup>

Grâce à la localisation par GPS, SkiReport offre aux skieurs la possibilité de trouver les zones de ski situées à proximité. Elle permet de mettre en évidence des points de chute de neige fraîche. SkiReport permet également d'accéder aux webcams des remontées mécaniques ainsi que l'obtention en temps réel d'informations météorologiques.

**Figure 4 : SkiReport**



Source : SkiReport - mai 2010

### Snocator<sup>5</sup>

Snocator informe le skieur sur sa position actuelle sur une carte et fournit des informations sur la piste. Cette application permet de planifier la prochaine descente en utilisant des cartes interactives. Snocator fournit des informations sur les emplacements des restaurants et de buvettes sur les pistes. Ce logiciel permet l'accès aux différentes webcams de la station et fournit également des informations météorologiques.

**Figure 5 : Snocator**



Source : Snocator - mai 2010

## Plateforme Android

### Satski<sup>6</sup>

Satski propose une sélection de cartes des stations de ski pour le monde entier et permet de fournir diverses informations sur la station, les restaurants, les agences de locations de matériel de ski, etc. Les points d'intérêt (POI) s'affichent automatiquement sur la carte. Pour terminer, Satski fournit un dispositif d'urgence permettant d'envoyer automatiquement sa position GPS au service compétent des remontées mécaniques.

**Figure 6 : Satski**



Source : Satski - mai 2010

<sup>4</sup> Source : <http://www.skireport.com> - mai 2010

<sup>5</sup> Source : <http://www.snocator.com> - mai 2010

<sup>6</sup> Source : <http://www.satsportsgps.com> - mai 2010

### TrialMaps<sup>7</sup>

TrialMaps permet de télécharger différentes cartes de stations de ski du Canada, des Etats Unis et de l'Europe. L'application permet d'accéder au bulletin d'enneigement de la station de ski sur laquelle l'usagé se trouve.

Figure 7 : TrialMaps



Source : TrialMaps - mai 2010

### iSki Trail Maps<sup>8</sup>

iSki Trail Maps permet essentiellement d'acquérir le plan des pistes de ski. Elle offre aussi différentes informations sur la station de ski (téléphone, adresse du site Internet,...).

Figure 8 : iSki Trail Maps



Source : iSki Trail Maps - mai 2010

### ***Récapitulatif des fonctionnalités intéressantes***

- ❖ *Affichage des remontées mécaniques, des pistes (avec leur niveau), des restaurants, des magasins de location de matériel de sport de neiges, des écoles de sports de neiges.*
- ❖ *Webcam des remontées mécaniques (si disponibles).*
- ❖ *Géolocalisation du sauveteur le plus proche par rapport à l'emplacement courant d'une personne accidentée. Affichage du nom et du numéro de téléphone du sauveteur.*
- ❖ *Possibilité de consulter la météo.*
- ❖ *Possibilité de consulter le bulletin d'enneigement.*

<sup>7</sup> Source : <http://www.clearlogic.com> - mai 2010

<sup>8</sup> Source : <http://www.appbrain.com/>

## 4 Cahier des charges initial

La phase d'analyse a permis de définir un cahier des charges initial contenant les fonctionnalités à développer.

Le but fondamental de cette application est de fournir des informations à destination des Smartphones des usagés des pistes de ski. Ces informations sont représentées par les points d'intérêt suivants :

- Pistes
- Remontées mécaniques
- Restaurants
- Sauveteurs

L'utilisateur doit pouvoir interagir avec ces points d'intérêt. Il est donc important que l'application soit intuitive, simple d'utilisation et surtout performante en terme de mise à jour des données. Cette application sera particulièrement utilisée sur les pistes de ski. Il se peut donc que les technologies EDGE et 3G, permettant à l'utilisateur de se connecter à Internet via son téléphone portable, ne soient pas toujours disponibles. Il devient donc nécessaire de minimiser les requêtes vers la base de données et de sauvegarder certaines données directement sur le Smartphone. La solution est de trier ces dernières de la manière suivante :

- Données sensibles aux mises à jour
- Données insensibles aux mises à jour

Les données insensibles aux mises à jour représentent par exemple l'emplacement d'un point d'intérêt tel qu'une piste, une remontée mécanique, un restaurant, etc. Ces données sont entrées en principe une seule fois dans la base de données et restent inchangées. Ce sont donc typiquement des données qui peuvent être directement enregistrées sur le téléphone portable lors de la première exécution de l'application.

Les données sensibles aux mises à jour sont par exemple le nombre de personnes dans la file d'attente d'une remontée mécanique, ou encore l'emplacement d'un sauveteur sur les pistes de ski. Ce sont des données qui devront régulièrement être mises à jour.

### 4.1 Fonctionnalités de base

#### Affichage du domaine skiable

L'application doit permettre l'affichage global de la carte du domaine skiable, contenant tous les différents points d'intérêt avec la possibilité de trier ces derniers par types (pistes, remontées mécaniques, restaurants, sauveteurs). Un cadre apparaît lors d'un clic sur l'un des points d'intérêt. Le cadre contient les informations de base sur ce dernier ainsi qu'un bouton, permettant d'obtenir des détails supplémentaires.

## Menu des points d'intérêt

Les points d'intérêt peuvent également être atteints par le biais d'un menu. Il s'agit ici d'une simple vue permettant de sélectionner un type de points d'intérêt (pistes, remontées mécaniques, restaurants, sauveteurs). L'utilisateur est ensuite redirigé vers un écran contenant la liste des points d'intérêt correspondant à sa sélection. Les types de points d'intérêt eux-mêmes peuvent être catégorisés de la manière suivante :

- Remontées mécaniques
  - Funiculaire
  - Télécabine
  - Télésiège
  - Téléski (arbalètes)
  - Téléski (assiettes)
  - Téléski (perches)
- Restaurants
  - Bar
  - Hôtel
  - Iglou
  - Pub
  - Restaurant
  - Self-Service
  - Snack
- Pistes
  - Verte
  - Bleue
  - Rouge
  - Noire

Il faut ensuite que l'utilisateur puisse trier les résultats par catégorie de type de points d'intérêt (par ex. que les restaurants de type « **Snack** »). Il est possible en tout temps de sélectionner un point d'intérêt de la liste des résultats pour obtenir ses détails. Cette vue détaillée permet également d'afficher le point d'intérêt sur le plan du domaine skiable par le biais d'un bouton « **Afficher sur la carte** ».

## 4.2 Fonctionnalités avancées

### Réalité augmentée

L'aspect « **Réalité augmentée** » fait partie des fonctionnalités avancées. Il s'agit ici de développer un écran qui se comporte de la manière suivante :

1. L'utilisateur est géolocalisé

2. La caméra du Smartphone s'active
3. Le Smartphone reconnaît l'exposition (en degrés de 0° à 359°) par rapport au point NORD (0°) et à la position actuelle du téléphone. A noter que cette fonctionnalité est uniquement applicable aux Smartphones équipés d'une boussole.
4. Par rapport aux coordonnées actuelles de l'utilisateur (point 1) et à l'exposition du Smartphone (point 3), les points d'intérêt sont superposés de manière dynamique au rendu de la caméra du Smartphone. Est affichée aussi la distance séparant l'utilisateur courant d'un point. Il est également possible de cliquer sur un point d'intérêt pour afficher ses informations spécifiques.

## 4.3 Autres fonctionnalités

### Planification d'un itinéraire

Une fonctionnalité intéressante serait également de pouvoir planifier un itinéraire sur la carte du domaine skiable. L'utilisateur pourrait indiquer le point de départ et d'arrivée sur la carte et l'itinéraire serait ensuite calculé et présenté de manière conviviale à l'utilisateur.

### Contact par téléphone

Certains points d'intérêt, tels que les restaurants et surtout les sauveteurs, doivent pouvoir être contactés par téléphone. Il faut donc optimiser l'application pour permettre aux utilisateurs de pouvoir appeler ces personnes à travers un simple clic.

### Fonctionnalité d'alerte

Cette fonctionnalité est liée à la sécurité sur, et à côté des pistes de ski. Elle permet à tout skieur d'alerter les sauveteurs en cas d'accidents sur les pistes du domaine skiable (skieur accidenté, avalanche, etc.). Le skieur sélectionne l'endroit ou la zone critique sur la carte ainsi que le type d'accident et confirme ensuite qu'il désire notifier les sauveteurs.

## 4.4 Planification du développement

No	Tâches à réaliser	Durée de développement
1.	Menu des points d'intérêt, affichage du détail des points d'intérêt	2 semaines
2.	Affichage du domaine skiable	1/2 de la tâche #1
3.	Planification d'un itinéraire	1/3 de la tâche #1
4.	Contact par téléphone et fonctionnalité d'alerte	1/4 de la tâche #1
5.	Réalité augmentée	Idem que la tâche #1

Les durées de développement ont toutes été estimées par rapport à la première tâche. Ainsi, on obtient une durée totale de développement de 43 jours.

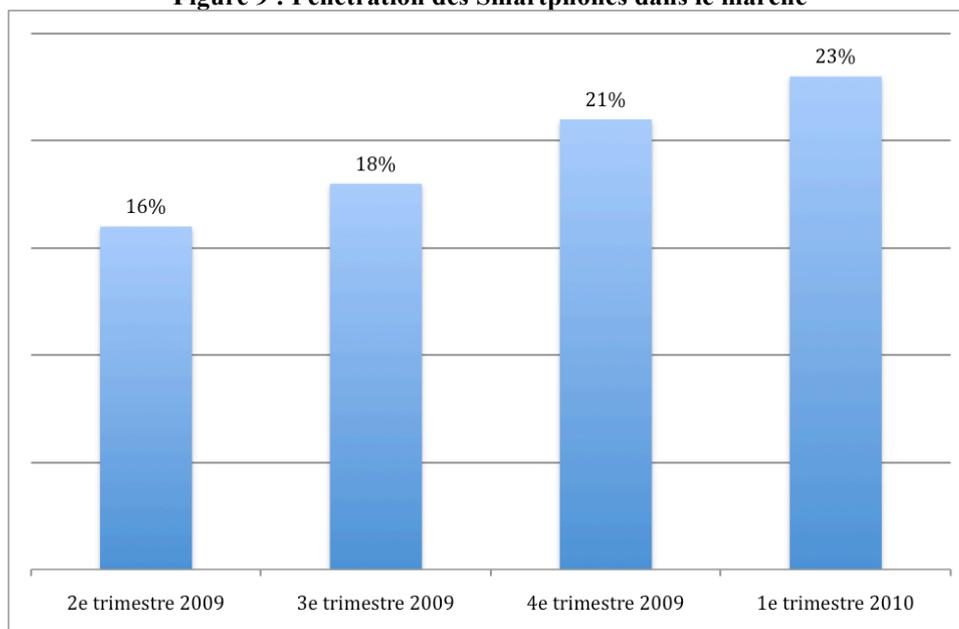
## 5 Choix des technologies et spécificités techniques

### 5.1 Plateforme de développement

#### Pénétration des Smartphones dans le marché

Le marché des Smartphones est en pleine croissance. Leurs capacités ont su convaincre les consommateurs de passer d'un simple téléphone mobile à un appareil plus sophistiqué. Les données de Nielsen, entreprise spécialisée dans les études de marché internationales, montrent qu'au 1<sup>er</sup> trimestre 2010, 23% des consommateurs mobiles sont en possession d'un Smartphone contre seulement 16% au 2<sup>e</sup> trimestre 2009.

Figure 9 : Pénétration des Smartphones dans le marché

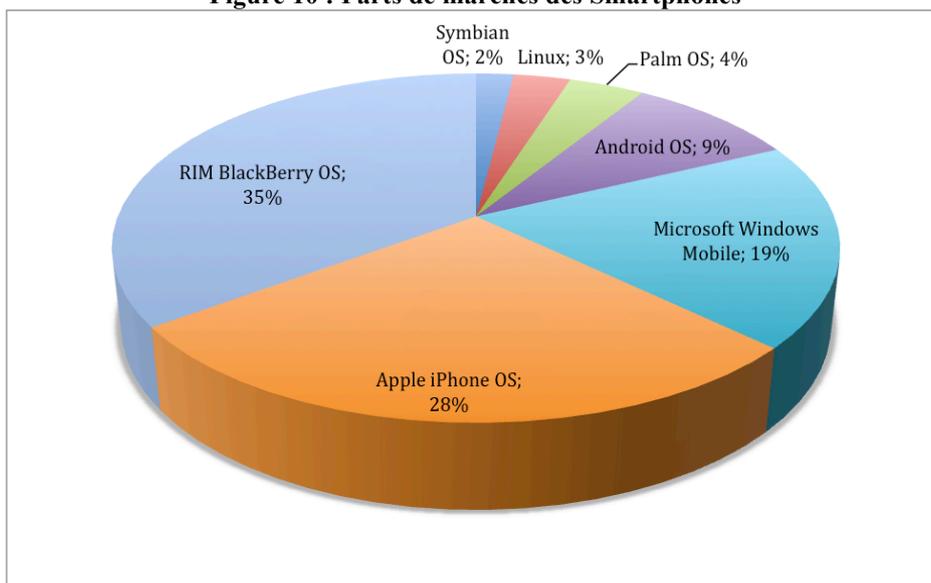


Source : [www.nielsen.com](http://www.nielsen.com)

## Parts de marchés des Smartphones

Les trois leaders mondiaux au niveau des parts de marché des Smartphones sont RIM BlackBerry, Apple, avec l'iPhone, et Android, grâce à son système d'exploitation. Les parts de marché d'Apple et d'Android ont augmenté de 2% entre le dernier trimestre 2009 et le 1<sup>er</sup> trimestre 2010. Pendant ce temps là, RIM BlackBerry et Windows Mobile ont eux perdu 2% de part de marché.

Figure 10 : Parts de marchés des Smartphones



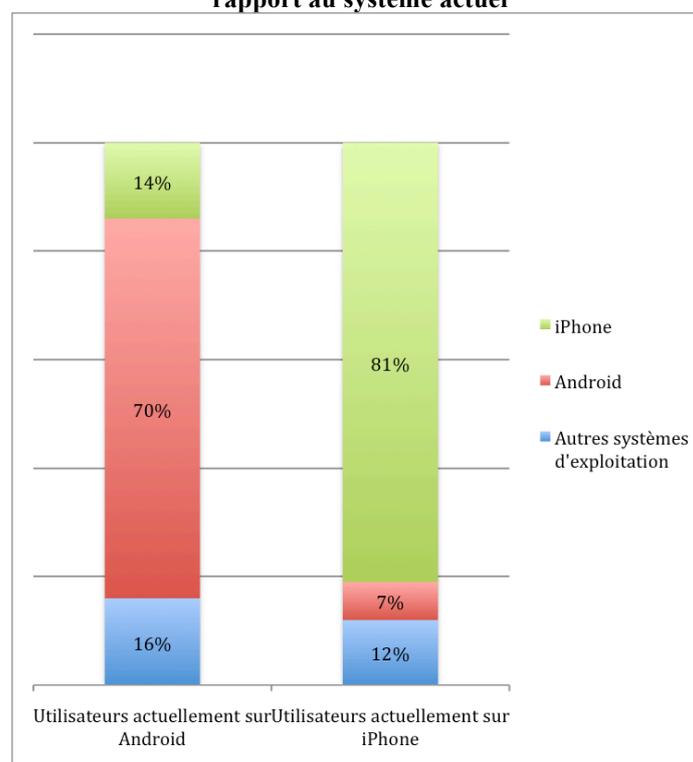
Source : [www.nielsen.com](http://www.nielsen.com)

## Prochain système d'exploitation désiré par rapport au système actuel

Le système de fidélisation est définitivement ce qui distingue l'iPhone et Android par rapport à ses concurrents car 80% des utilisateurs d'iPhone veulent que leur prochain appareil reste un iPhone. De même, 70% des utilisateurs actuellement sur Android sont prêts à réutiliser le même système d'exploitation sur leur prochain appareil mobile.

Parmi les utilisateurs d'Android, 14% seraient tentés de passer à l'iPhone pour seulement 7% qui passeraient de l'iPhone à Android.

**Figure 11 : Prochain système d'exploitation désiré par rapport au système actuel**



Source : [www.nielsen.com](http://www.nielsen.com)

## Choix définitif de la plateforme de développement

Les statistiques ci-dessus montrent que l'avenir de l'iPhone dans le marché des Smartphones est très prometteur. Bien que Android a drastiquement augmenté sa part de marché, l'iPhone reste dans la compétition. La sortie de l'iPhone 4 va certainement à nouveau bouleverser le monde des Smartphones puisque, déjà avant sa sortie, toutes les préventes étaient saturées.

L'avantage que possède Apple sur Android est que cette entreprise ne produit pas que des Smartphones mais également des ordinateurs portables, fixes, ainsi que d'autres appareils comme par exemple l'iPad. Ces produits attirent énormément l'attention des consommateurs. Leur diversité a déjà fasciné passablement de personnes qui, aujourd'hui, n'achèteraient plus jamais autre chose que des produits Apple. L'achat d'un ordinateur portable Mac pousse à la consommation d'un iPhone et vice versa. Par exemple, il est impossible de développer une application iPhone sans être en possession d'un ordinateur Mac, car le logiciel de

développement Xcode n'a pas été conçu pour un autre système d'exploitation que Mac OS X. Ceci rend la tâche des concurrents de Apple très difficile.

Ce que je conclus de cette analyse est que Apple reste une entreprise très fermée au niveau de sa politique de vente. Leur manière d'agir reste toutefois compréhensible car, actuellement, leur succès ne les oblige en rien à s'ouvrir davantage. L'iPhone est très apprécié par ses utilisateurs et ceci est un fait qui ne changera pas de sitôt.

Ces résultats m'ont donc poussé à choisir l'iPhone comme plateforme de développement pour l'application mobile. Ce qui m'a le plus encouragé à prendre cette décision est le fait que je n'ai encore jamais développé ni en C, ni en Objective-C et que je voulais apprendre une nouvelle technologie durant mon travail de bachelor.

## 5.2 Système de gestion de base de données (SGBD)

Suite au choix de la plateforme de développement, la décision du système de gestion de base de données a dû se faire entre SQLite et MySQL. SQLite permet de pouvoir directement questionner une base de données à travers des bibliothèques disponibles en Objective-C, ce qui n'est pas le cas pour MySQL. Par contre, il est aussi très simple depuis une application iPhone d'interpréter du XML.

J'ai donc décidé de choisir MySQL comme système de gestion de base de données et de créer ensuite un service web en XML. C'est à travers ce dernier que l'application iPhone pourra consulter les données se trouvant dans la base. Les bases de données MySQL peuvent aussi très facilement être administrées à travers l'outil phpMyAdmin.

## 5.3 Couche d'accès aux données

Figure 12 : Couche d'accès aux données - Table restaurant



T_RESTAURANT	
idRestaurant	INT
idRestauranttype	INT
restaurantLatitude	DOUBLE
restaurantLongitude	DOUBLE
restaurantAltitude	INT
restaurantHasTerrasse	SMALLINT
restaurantPhone	VARCHAR(30)

Une couche d'accès aux données, ou DAO (Data Access Object) en Anglais, a été créée pour accéder aux données de la base de manière rapide et efficace. Chaque table figurant dans la base de données correspond à un objet bien précis.

Voici à votre gauche la représentation de la table T\_RESTAURANT. Elle contient six champs, dont une clé primaire et une clé étrangère. Dans cette table, des données seront insérées, lues et supprimées à travers une interface administrateur que vous aurez l'occasion de découvrir tout à l'heure.

Chaque classe objet est composée des attributs, setteurs et getteurs correspondants au contenu de la table se trouvant dans la base de données. Les classes objet sont développées dans le même langage que l'interface administrateur (voir chap. 6.1 - Interface administrateur), c'est-à-dire en PHP.

Le fichier correspondant à la table T\_RESTAURANT de la base de données représentée sur la Figure 12 a donc été créé de la manière suivante :

Figure 13 : Couche d'accès aux données - Classe restaurant

```

<?php
DEFINE("RESTAURANT", "T_RESTAURANT") ;

class Restaurant {

    // Attributes
    private $idRestaurant;
    private $idRestauranttype;
    private $restaurantLatitude;
    private $restaurantLongitude;
    private $restaurantAltitude;
    private $restaurantHasTerrasse;
    private $restaurantPhone;

    // Constructor
    function Restaurant() {}

    // Setters
    public function setIdRestaurant($pValue = 0)           { $this->idRestaurant = $pValue; }
    public function setIdRestaurantType($pValue = 0)      { $this->idRestauranttype = $pValue; }
    public function setRestaurantLatitude($pValue = 0)    { $this->restaurantLatitude = $pValue; }
    public function setRestaurantLongitude($pValue = 0)   { $this->restaurantLongitude = $pValue; }
    public function setRestaurantAltitude($pValue = 0)   { $this->restaurantAltitude = $pValue; }
    public function setRestaurantHasTerrasse($pValue = 0) { $this->restaurantHasTerrasse = $pValue; }
    public function setRestaurantPhone($pValue = 0)       { $this->restaurantPhone = $pValue; }

    // Getters
    public function getIdRestaurant()                     { return $this->idRestaurant; }
    public function getIdRestaurantType()                 { return $this->idRestauranttype; }
    public function getRestaurantLatitude()               { return $this->restaurantLatitude; }
    public function getRestaurantLongitude()              { return $this->restaurantLongitude; }
    public function getRestaurantAltitude()              { return $this->restaurantAltitude; }
    public function getRestaurantHasTerrasse()            { return $this->restaurantHasTerrasse; }
    public function getRestaurantPhone()                  { return $this->restaurantPhone; }

    // Methods
    // ...
}
?>

```

Chaque objet contient également les méthodes permettant une interaction avec la base de données :

```
public function readObject( $array = array() )
```

Cette méthode interroge la base de données par rapport à une série d'arguments passés en paramètre. Elle retourne un objet du type de la classe dans laquelle elle est définie.

```
public static function readArray( $array = array() )
```

Cette méthode interroge la base de données par rapport à une série d'arguments passés en paramètre. Le nombre d'arguments peut varier en fonction de l'objet en question. La méthode retourne un tableau contenant les informations correspondantes aux arguments de la méthode.

```
public function writeObject()
```

Cette méthode permet l'insertion ou la mise à jour d'un enregistrement dans la base de données. Si l'identifiant de l'objet appelant la méthode est nul, la méthode exécutera une nouvelle insertion de cet objet dans la base de données ; dans le cas contraire, une mise à jour sera effectuée sur l'enregistrement correspondant à l'identifiant de l'objet.

```
public static function deleteObject( $id = 0 )
```

Cette méthode supprime un enregistrement de la base de données par rapport à un identifiant passé en paramètre.

```
private static function buildQuery( $array = array() )
```

Cette fonction est appelée au début des méthodes « **readObject** » et « **readArray** » et permet de construire la requête qui va être envoyée vers la base de données par rapport à une série d'arguments passés en paramètres. Elle retourne une chaîne de caractères contenant la requête. Elle est utilisée uniquement dans la classe elle-même. C'est pour cette raison qu'elle est privée.

## 5.4 Utilisation des APIs Google

Google a lancé l'API Google Maps en Juin 2005 pour permettre aux développeurs d'intégrer Google Maps dans leurs sites Web. Pour l'application que je dois développer à travers ce travail de bachelor, il est impératif d'utiliser cet API pour la localisation des points d'intérêt sur la carte du monde.

Tous les points se trouvant sur une carte Google sont définis par leur latitude et leur longitude. Lors de l'administration des points d'intérêt (voir chap. 6.1 - Interface administrateur), l'utilisateur doit indiquer sur des cartes Google à quel endroit ces derniers se trouvent. Grâce à l'API, la latitude et la longitude peuvent être récupérées et enregistrées dans la base de données.

Il existe également une API appelée Google Weather. Cette dernière fonctionne sur le même principe que l'API Google Maps sauf qu'elle a été conçue pour recueillir des informations sur les conditions météorologiques d'un lieu.

## 5.5 Utilisation de l'API Topocoding

Ce service est une extension de l'API Google décrite ci-dessus. Il permet de récupérer l'altitude d'un point d'une carte Google à partir de ses coordonnées géographiques (latitude, longitude).

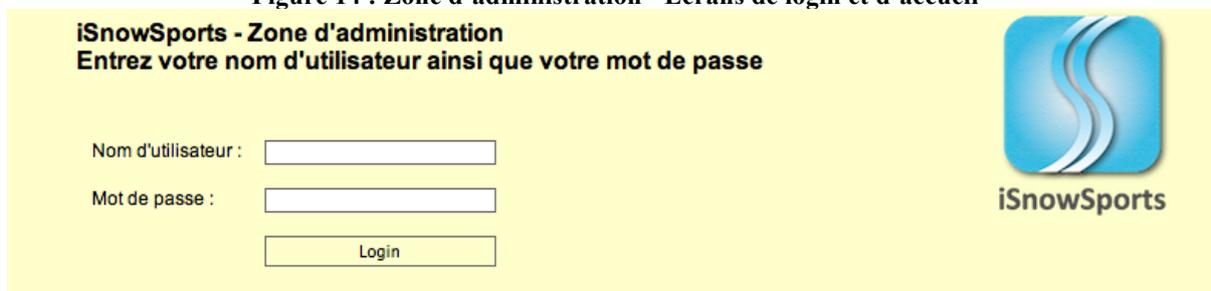
## 6 Descriptif de l'application

### 6.1 Interface administrateur

La présence d'une interface administrateur est indispensable pour ce type d'applications. Celle-ci doit permettre l'insertion, la modification et la suppression des points d'intérêt. Ces données seront ensuite utilisées par l'application mobile pour afficher les points d'intérêt de manière ergonomique sur un Smartphone. Voici ci-dessous les écrans de login et d'accueil.

Les informations concernant le « **nom d'utilisateur** » et le « **mot de passe** » à utiliser pour se connecter à ce service, sont disponibles en annexe de ce document au chapitre 12.1 - Accès aux ressources.

Figure 14 : Zone d'administration - Ecrans de login et d'accueil



**iSnowSports - Zone d'administration**  
Entrez votre nom d'utilisateur ainsi que votre mot de passe

Nom d'utilisateur :

Mot de passe :

  
iSnowSports



**iSnowSports - Zone d'administration**

 Pistes
  Remontées mécaniques
  Restaurants
  Sauvetage

  
iSnowSports

Bonjour,

Bienvenue sur l'interface d'administration de iSnowSports.

Depuis cet interface, vous pouvez administrer en toute tranquillité les points d'intérêts de l'application.

La fonctionnalité de sauvetage vous permet de contrôler les missions de sauvetage à traiter dans le domaine skiable. Lorsqu'une personne appelle la centrale de sauvetage à travers l'application iSnowSports, vous êtes directement informés de la position à laquelle se trouve actuellement le skieur accidenté. Vous pouvez ensuite contacter vos sauveteurs et les envoyer à l'endroit désiré.

Je vous souhaite d'ores et déjà beaucoup de plaisir lors de l'administration de l'application.

#### Ajout, modification et suppression des points d'intérêt

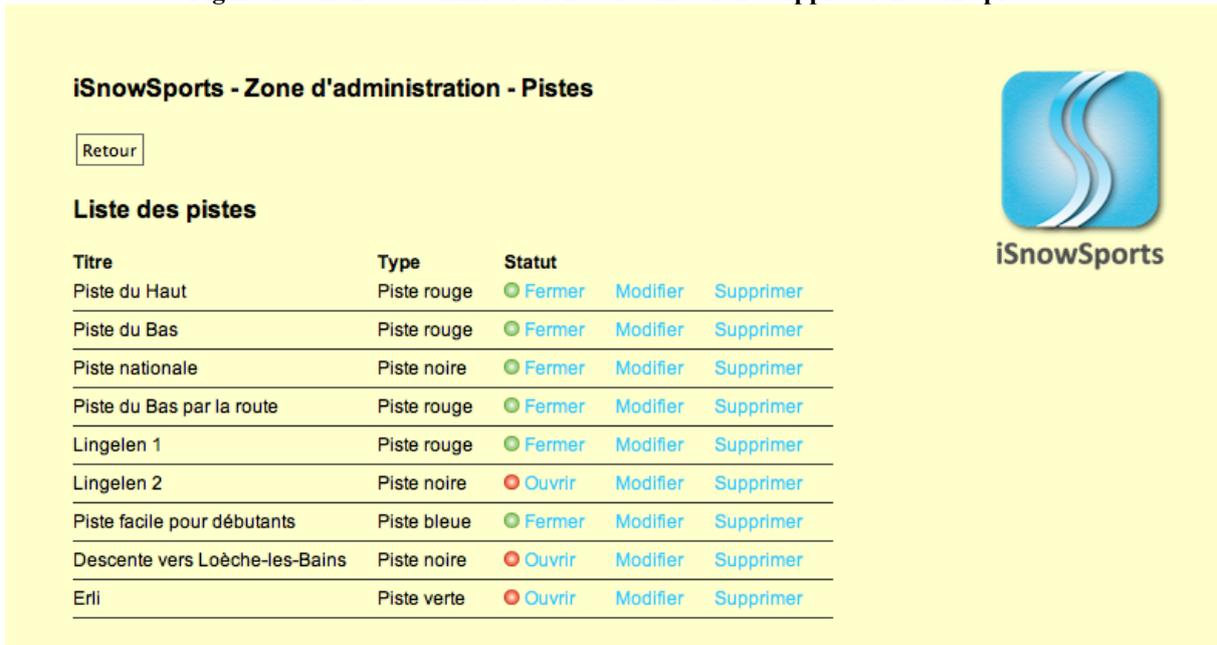
Pour chaque point d'intérêt, un écran d'administration différent a été conçu. Comme vous pouvez le distinguer sur la Figure 14 ci-dessus, l'utilisateur peut atteindre les écrans d'administration de chaque type de points d'intérêt en cliquant sur l'icône respective.

### Administration des pistes

Voici ci-dessous une copie d'écran de la page d'administration des pistes de ski. La première partie de l'écran affiche la liste des pistes de ski déjà enregistrées dans la base de données et permet donc de les modifier ou de les supprimer.

L'administrateur a également la possibilité de modifier le statut d'une piste (ouvrir / fermer).

**Figure 15 : Zone d'administration - Modification / suppression d'une piste**



**iSnowSports - Zone d'administration - Pistes**

[Retour](#)

**Liste des pistes**

Titre	Type	Statut		
Piste du Haut	Piste rouge	 Fermer	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Piste du Bas	Piste rouge	 Fermer	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Piste nationale	Piste noire	 Fermer	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Piste du Bas par la route	Piste rouge	 Fermer	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Lingelen 1	Piste rouge	 Fermer	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Lingelen 2	Piste noire	 Ouvrir	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Piste facile pour débutants	Piste bleue	 Fermer	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Descente vers Loèche-les-Bains	Piste noire	 Ouvrir	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Erli	Piste verte	 Ouvrir	<a href="#">Modifier</a>	<a href="#">Supprimer</a>

  
iSnowSports

La deuxième partie de l'écran d'administration des pistes de ski est réservée à l'insertion. L'utilisateur doit y saisir les données principales de la nouvelle piste de ski, ainsi que tracer le parcours de cette dernière.

Figure 16 : Zone d'administration - Insertion d'une piste

Titre : \*

Description : \*

Type de piste : \*

Enregistrer

Chaque piste est représentée par une multitude de coordonnées géographiques. A travers l'API de Google et celle de Topocoding, il est possible de récupérer la latitude, la longitude ainsi que l'altitude d'un point lors de l'interaction de l'utilisateur avec la carte. Il est possible également de calculer la distance entre un point et un autre grâce à des fonctions spécifiques.

En résumé, lors de l'enregistrement d'une nouvelle piste de ski, nous connaissons son titre, sa description et son type, ainsi que toutes les coordonnées de ses points sur la carte du monde. Nous connaissons également l'altitude de chaque point de la piste et la distance qui les sépare les uns des autres.

A partir du moment où nous sommes en possession de toutes ces informations, il est possible d'effectuer une multitude de calculs tels que :

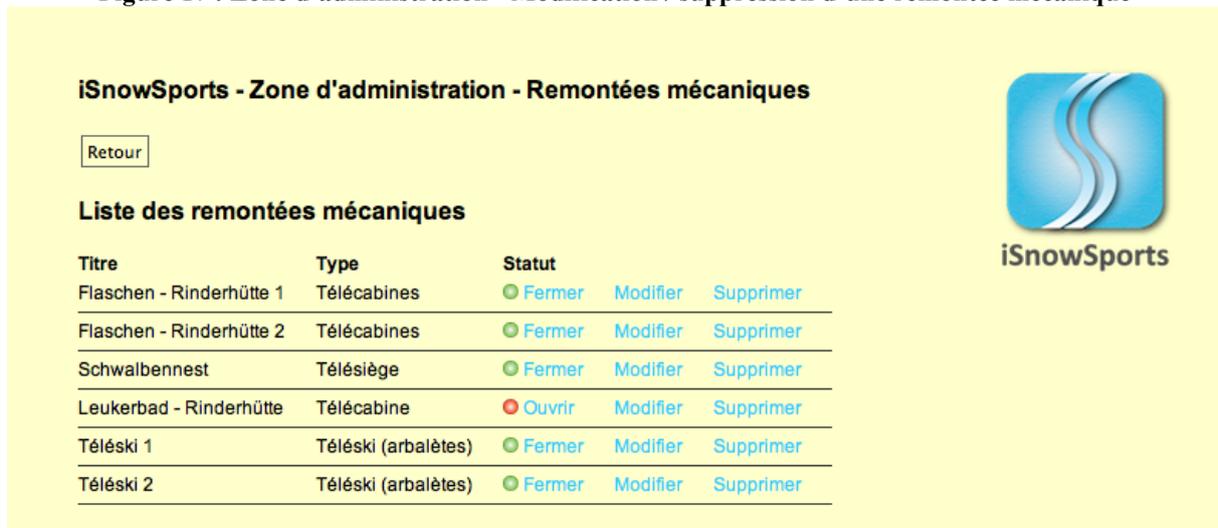
- Dénivelé
- Distance totale
- Pente

Administration des remontées mécaniques

Le principe de l'administration des remontées mécanique est le même que celui des pistes de ski, avec cette différence qu'une remontée mécanique est composée de deux points seulement sur la carte (départ et arrivée). Au sommet de l'écran, nous apercevons la liste des remontées mécaniques déjà enregistrées, comme le démontre la Figure 17 ci-dessous.

L'administrateur peut, comme nous l'avons vu précédemment pour les pistes de skis, modifier le statut des remontées mécaniques (ouvrir / fermer).

**Figure 17 : Zone d'administration - Modification / suppression d'une remontée mécanique**



**iSnowSports - Zone d'administration - Remontées mécaniques**

[Retour](#)

**Liste des remontées mécaniques**

Titre	Type	Statut		
Flaschen - Rinderhütte 1	Télécabines	<span style="color: green;">●</span> Fermer	Modifier	Supprimer
Flaschen - Rinderhütte 2	Télécabines	<span style="color: green;">●</span> Fermer	Modifier	Supprimer
Schwalbennest	Télésiège	<span style="color: green;">●</span> Fermer	Modifier	Supprimer
Leukerbad - Rinderhütte	Télécabine	<span style="color: red;">●</span> Ouvrir	Modifier	Supprimer
Téléski 1	Téléski (arbalètes)	<span style="color: green;">●</span> Fermer	Modifier	Supprimer
Téléski 2	Téléski (arbalètes)	<span style="color: green;">●</span> Fermer	Modifier	Supprimer

  
iSnowSports

La deuxième partie de la page est réservée à l'insertion d'une nouvelle remontée mécanique et se présente de la manière suivante :

Figure 18 : Zone d'administration - Insertion d'une remontée mécanique

Titre : *	<input type="text" value="Schwalbennest"/>
Description : *	<input type="text" value="Télesiège 6 places permettant d'accéder à la piste des Lingeles 1."/>
Type de remontée : *	<input type="text" value="Télesiège"/>
Nombre de remontées : *	<input type="text" value="50"/>
Capacité (par remontée) : *	<input type="text" value="6"/>
Distance du tour (m) : *	<input type="text" value="2300"/>
Vitesse maximale (km/h) : *	<input type="text" value="18"/>

Départ	Arrivée
 <p>           Latitude 46.36778            Longitude 7.65142            Altitude 2242         </p>	 <p>           Latitude 46.37459            Longitude 7.66348            Altitude 2587         </p>

Les champs « **Nombre de remontées** », « **Capacité (par remontée)** », « **Distance du tour (m)** » et « **Vitesse maximale (km/h)** » sont intéressants pour connaître la capacité d'une remontée mécanique par rapport à une durée déterminée.

Soit le nombre de remontées défini par « **r** », la capacité par « **c** », la distance du tour par « **d** » et la vitesse maximale par « **v** ». On obtient la formule suivante :

$$capacity = (r \cdot c \div 2) \cdot (v \cdot 1000 \div d)$$

Si nous attribuons maintenant les chiffres saisis en Figure 17 à cette formule, nous obtenons le résultat suivant :

$$capacity = (50 \cdot 6 \div 2) \cdot (18 \cdot 1000 \div 2300)$$

$$capacity = 1173.9$$

Ce résultat signifie que la remontée mécanique est capable de transporter **1173 personnes** par heures depuis son départ jusqu'à son arrivée.

Pour les usagés des pistes de ski, il serait intéressant de connaître le temps d'attente auprès des installations du domaine skiable. Dans les grandes stations, chaque remontée mécanique est équipée d'un tourniquet où les touristes doivent valider leur abonnement de ski. Le nombre de personnes utilisant la remontée mécanique est alors incrémenté dans une base de données à chaque fois qu'un touriste franchit le tourniquet. Ceci permet à la station de tirer un bilan en fin de journée sur le nombre de personnes ayant utilisé l'installation en question.

En comparant ce chiffre avec la formule que nous venons de traiter, il serait possible de calculer le temps d'attente actuel auprès des remontées mécaniques (voir aussi chap. 3 - Etat de l'art). Or, toutes les stations ne possèdent pas les moyens financiers pour placer un lecteur d'abonnement à chaque remontée du domaine skiable. Pour cette raison, cette fonctionnalité est pour le moment mise en attente.

### Administration des restaurants

Un restaurant est défini par un titre, une description et un type. Le touriste pourrait également désirer savoir si le restaurant en question est pourvu d'une terrasse extérieure. Il peut être pratique également d'être en possession du numéro de téléphone du restaurant et de pouvoir accéder à quelques photos. Voici ci-dessous l'écran permettant de consulter la liste des restaurants déjà enregistrés dans la base de données :

Figure 19 : Zone d'administration - Modification / suppression d'un restaurant



**iSnowSports - Zone d'administration - Restaurants**

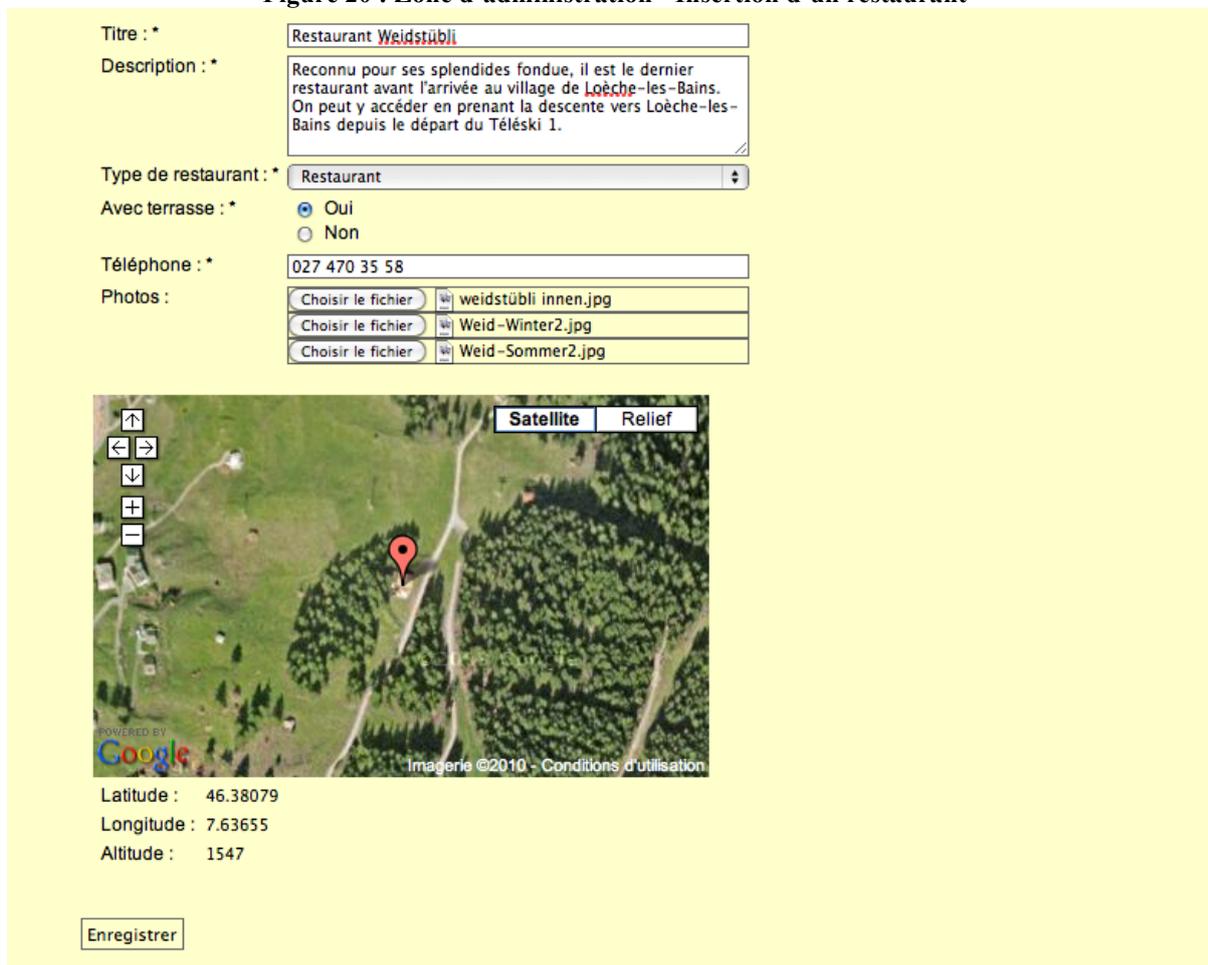
[Retour](#)

**Liste des restaurants**

Titre	Type		
Restaurant Briand	Self-Service	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Restaurant Flaschen	Restaurant	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Hotel Torrenthorn	Hotel	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
Restaurant Alpengruss	Restaurant	<a href="#">Modifier</a>	<a href="#">Supprimer</a>

La figure suivante représente le formulaire de saisie d'un nouveau restaurant :

Figure 20 : Zone d'administration - Insertion d'un restaurant



Titre : \* Restaurant **Weidstübli**  
 Description : \* Reconnu pour ses splendides fondue, il est le dernier restaurant avant l'arrivée au village de Loèche-les-Bains. On peut y accéder en prenant la descente vers Loèche-les-Bains depuis le départ du Télési 1.  
 Type de restaurant : \* Restaurant  
 Avec terrasse : \*  Oui  Non  
 Téléphone : \* 027 470 35 58  
 Photos : Choisir le fichier weidstübli innen.jpg  
 Choisir le fichier Weid-Winter2.jpg  
 Choisir le fichier Weid-Sommer2.jpg  
 Latitude : 46.38079  
 Longitude : 7.63655  
 Altitude : 1547  
 Enregistrer

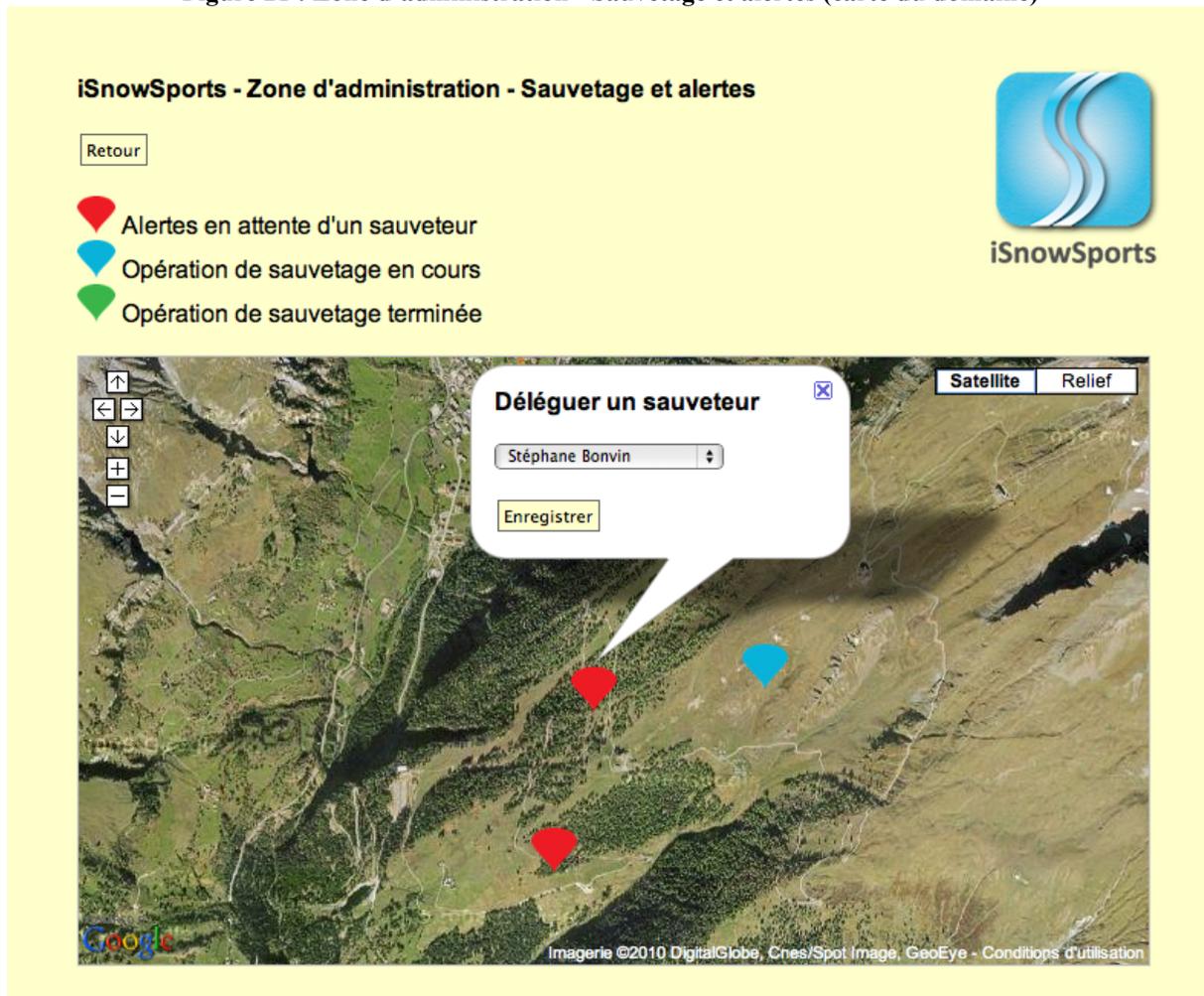
## Gestion des alertes

La gestion des alertes permet aux sauveteurs d'avoir une vue d'ensemble sur les opérations de sauvetage du domaine skiable. Nous verrons plus bas qu'il est possible d'appeler les secours à travers la fonctionnalité d'« **Alerte rapide** » de l'interface mobile sur iPhone. L'appel d'une personne est directement géolocalisé et le point de l'incident est affiché sur une carte. Le centre de sauvetage peut ensuite interagir avec les différentes annotations se trouvant sur la carte et attribuer ainsi une opération de sauvetage à un sauveteur.

Les alertes en attente d'un sauveteur sont affichées en rouge, les opérations de sauvetage en cours en bleu et celles qui sont terminées en vert.

Voici donc la première partie de l'écran de sauvetage avec la carte du domaine skiable contenant les différentes alertes :

Figure 21 : Zone d'administration - Sauvetage et alertes (carte du domaine)



La deuxième partie de l'écran représente la liste des sauveteurs avec leur statut actuel. Il est également possible de modifier ou de supprimer un sauveteur.

Figure 22 : Zone d'administration - Sauvetage et alertes (liste des sauveteurs)

Liste des sauveteurs						
	Nom	Prénom	Email	Téléphone		
	Bonvin	Stéphane	<a href="mailto:stephane.bonvin@isnowsports.com">stephane.bonvin@isnowsports.com</a>	027 345 34 34 079 345 34 34	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
	Devantéry	Raphaël	<a href="mailto:raph.dev@isnowsports.com">raph.dev@isnowsports.com</a>	027 456 56 56 079 456 56 56	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
	Pitteloud	Gérald	<a href="mailto:jetpit@isnowsports.com">jetpit@isnowsports.com</a>	027 123 23 23 079 123 23 23	<a href="#">Modifier</a>	<a href="#">Supprimer</a>
	Salamin	Hervé	<a href="mailto:hervé.salamin@isnowsports.com">hervé.salamin@isnowsports.com</a>	027 234 23 23 079 234 23 23	<a href="#">Modifier</a>	<a href="#">Supprimer</a>

La troisième partie de l'écran permet l'insertion d'un nouveau sauveteur :

Figure 23 : Zone d'administration - Sauvetage et alertes (insertion d'un sauveteur)



## 6.2 Application mobile

L'application mobile a pour but d'aider les usagés des pistes de skis à se repérer dans le domaine skiable. Pour ce faire, elle utilise un service web afin de récupérer les données nécessaires à son bon fonctionnement. A chaque lancement de l'application, cette dernière vérifie s'il existe une version plus récente de l'application ou, plus exactement, si des données relatives aux points d'intérêt ont été modifiées dans la base de données depuis le dernier lancement de l'application. Si tel est le cas, l'application va automatiquement mettre à jour les données sur le Smartphone.

### Accueil



Voici l'écran d'accueil de l'application mobile sur iPhone. Lors du lancement de l'application, l'utilisateur est immédiatement géolocalisé. Son emplacement est alors affiché sur une carte qui apparaît derrière le logo iSnowSports.

L'application va ensuite vérifier quel est le point d'intérêt le plus proche de l'emplacement courant de l'utilisateur et affichera la distance ainsi que les informations adéquates dans une bulle d'informations.

Depuis cet écran, l'utilisateur peut accéder à la « Carte » du domaine, à un menu des « Points d'intérêt » ainsi qu'à l'écran d'« Alerte rapide ». Il peut également consulter le « Statut » des remontées mécaniques et des pistes de ski, la « Météo » ainsi que l'« Aide » qui lui permettra d'obtenir de plus amples informations sur l'utilisation de l'application en général.

## La carte du domaine skiable

Figure 25 : Application mobile - Carte du domaine skiable



Cet écran s'affiche lorsque l'utilisateur appuie sur le bouton « **Carte** » de l'écran d'accueil. On y aperçoit tous les points d'intérêt du domaine skiable.

Les pistes sont tracées sur la carte par rapport à leur degré de difficulté (vertes, bleues, rouges ou noires).

Les remontées mécaniques sont tracées en jaune. Une icône correspondant au type de remontée s'affiche au centre de chacune d'entre elles.

Les restaurants quant à eux sont marqués d'une icône verte.

Il est possible d'interagir avec tous les points d'intérêt. Un simple clic sur une icône de la carte ouvre une bulle d'informations contenant les données pertinentes du point d'intérêt sélectionné.

L'utilisateur peut obtenir davantage de détails sur le point d'intérêt en question en cliquant sur la flèche bleue se trouvant à droite de la bulle d'informations.

Au sommet à gauche de l'écran se trouve le bouton « **Retour** », qui permet de revenir à l'écran précédent, c'est-à-dire l'écran d'accueil. Ce dernier est présent sur tous les écrans que nous allons découvrir par la suite, et permet une navigation rapide et efficace entre les différents modes d'affichage de l'application. Le bouton au fond à gauche permet en tout temps de centrer la carte du domaine par rapport à l'emplacement courant de l'utilisateur.

## Liste des points d'intérêt

Figure 26 : Application mobile - Liste des points d'intérêt



Figure 27 : Application mobile - Liste des points d'intérêt - Pistes



Figure 28 : Application mobile - Liste des points d'intérêt - Remontées mécaniques



Figure 29 : Application mobile - Liste des points d'intérêt - Restaurants



L'écran représenté en Figure 26 s'affiche lorsque l'utilisateur appuie sur le bouton « **Points d'intérêt** » de l'écran d'accueil. Ce dernier est représenté sous forme d'une liste contenant les trois types principaux de points d'intérêt : les pistes, les remontées mécaniques et les restaurants. Les Figures 27, 28 et 29 montrent l'affichage des types de points d'intérêt adéquats.

La liste des pistes de ski contient une icône à gauche avec la couleur correspondante à son degré de difficulté. L'utilisateur est également informé si les pistes sont ouvertes ou fermées.

La liste des remontées mécaniques est également enrichie grâce à une icône permettant de reconnaître le type de chaque remontée (télécabine, télésiège, télési, etc.). Là aussi, le statut de la remontée mécanique (ouvert / fermé) est indiqué.

Quant à la liste des restaurants, il est possible de trier ces derniers par distance, toujours par rapport à l'emplacement courant de l'utilisateur.

Détail des pistes de ski

**Figure 30 : Application mobile - Détail des pistes**



**Figure 31 : Application mobile - Détail des pistes sur la carte**



Depuis la capture d'écran représentée sur la Figure 27, l'utilisateur a la possibilité de consulter le détail d'une piste de ski (Figure 30).

Il retrouvera sur cet écran un graphique qui affiche le profil de la piste sélectionnée, l'altitude au départ, celle à l'arrivée, le dénivelé, le pourcentage moyen de la pente et, finalement, la distance totale de la piste.

Le bouton « **Afficher sur la carte** » permet à l'utilisateur de visualiser l'emplacement précis de la piste sur la carte du domaine, comme le démontre la Figure 31.

Détail des remontées mécaniques

Figure 32 : Application mobile - Détail des remontées mécaniques



Figure 33 : Application mobile - Détail des remontées mécaniques sur la carte



L'écran que l'on peut apercevoir sur la Figure 28 permet à l'utilisateur d'accéder aux informations détaillées d'une remontée mécanique (Figure 32).

Il est informé sur l'altitude au départ et à l'arrivée de la remontée. Cet écran affiche également le dénivelé, la capacité individuelle ainsi que la capacité par heure de l'installation.

De même que pour les pistes de ski, le bouton « **Afficher sur la carte** » sert à la visualisation de l'emplacement précis de la remontée mécanique sur la carte du domaine, comme l'indique la Figure 33 ci-dessus.

### Détail des restaurants

Figure 34 : Application mobile - Détail des restaurants



Figure 35 : Application mobile - Détail des restaurants sur la carte



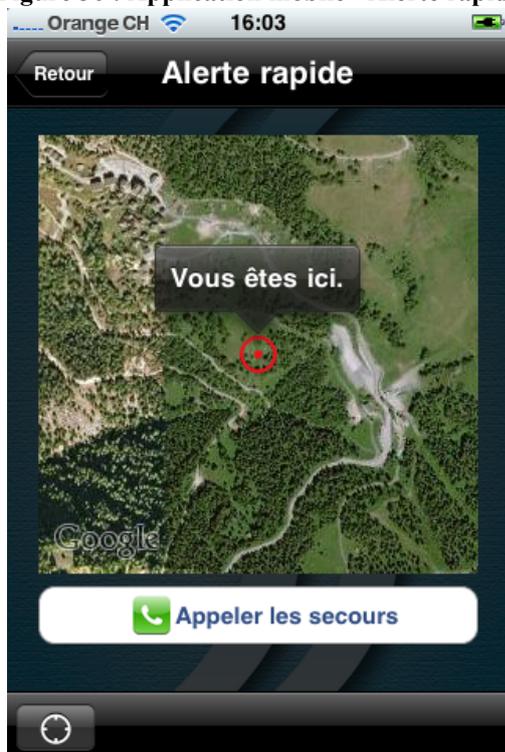
Cet écran apparaît lorsque l'utilisateur sélectionne un restaurant parmi la liste des restaurants représentés sur la Figure 29.

L'utilisateur est informé sur l'altitude du restaurant ainsi que sur la présence éventuelle d'une terrasse ou non. Une photo s'affiche également à droite de l'écran ; celle-ci est accompagnée d'un texte offrant un bref descriptif du restaurant en question.

L'utilisateur a la possibilité d'appeler directement le restaurant afin d'effectuer une éventuelle réservation. Comme l'indique la capture d'écran représentée par la figure 35, le bouton « **Afficher sur la carte** » permet l'affichage de l'emplacement précis du restaurant sur la carte du domaine.

## Alerte rapide

Figure 36 : Application mobile - Alerte rapide



Cet écran s'affiche lorsque l'utilisateur appuie sur le bouton « **Alerte rapide** » de l'écran d'accueil.

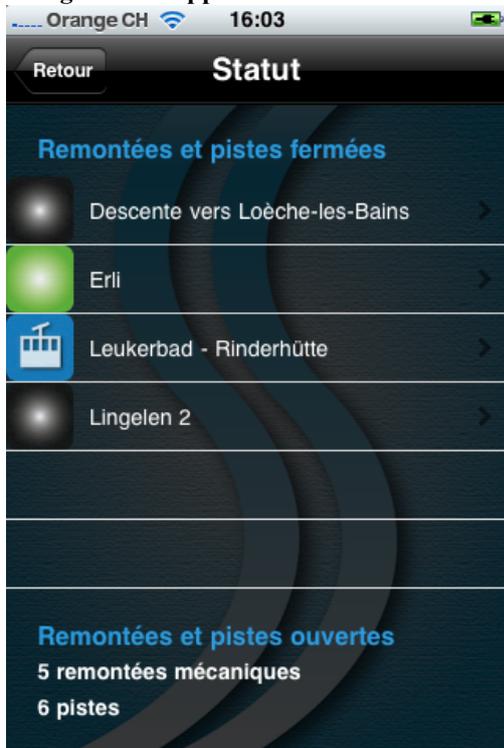
Cette fonctionnalité permet à l'utilisateur d'appeler les secours du domaine skiable afin d'obtenir rapidement de l'aide lors d'un quelconque accident sur les pistes. Le but de cette fonctionnalité est de minimiser la durée de déplacement d'un sauveteur à l'endroit de l'incident et d'améliorer ainsi le déroulement des opérations de sauvetage.

Lors du chargement de cet écran, la position actuelle de l'utilisateur est immédiatement mise à jour. Cette géolocalisation de l'utilisateur permet d'informer les sauveteurs de la position de l'alerte. Le centre de sauvetage a la possibilité d'afficher une carte du domaine skiable contenant les différentes alertes en cours, comme indiqué en fin du chapitre 6.1 - Interface administrateur - Gestion des alertes.

Le bouton au fond à gauche permet en tout temps de recalculer l'emplacement courant de l'utilisateur.

## Statut des remontées mécaniques et des pistes de ski

Figure 37 : Application mobile - Statut



Cet écran s'affiche lorsque l'utilisateur appuie sur le bouton « **Statut** » de l'écran d'accueil.

En cas de vent ou d'intempéries, il est possible que certaines remontées mécaniques ne soient pas en service ou qu'une piste soit fermée.

A chaque chargement de cet écran, l'application vérifie dans la base de données si l'état des remontées mécaniques ainsi que celui des pistes de ski ont été modifiés.

Cet écran permet donc à l'utilisateur d'être continuellement informé sur l'état des remontées mécaniques et des pistes de ski. Sur la capture d'écran affichée à gauche, on aperçoit que deux pistes noires, une piste verte et une remontée mécanique sont fermées.

L'utilisateur est également informé sur le fait que cinq remontées mécaniques et six pistes sont ouvertes.

## Météo

Figure 38 : Application mobile - Météo



Cet écran apparaît lorsque l'utilisateur appuie sur le bouton « **Météo** » de l'écran d'accueil.

Grâce à l'API Google Weather, il est possible de récupérer un fichier XML contenant les informations météorologiques d'un endroit bien précis. Cet écran questionne donc le service web en question et affiche la température, les conditions actuelles de la station ainsi que les prévisions pour les quatre prochains jours.

Les données sont mises à jour assez fréquemment (1 fois par heure), ce qui permet à l'utilisateur d'avoir des informations pertinentes sur la météo actuelle.

## Aide

Figure 39 : Application mobile - Aide



Finalement, voici l'écran d'aide. Ce dernier s'affiche lorsque l'utilisateur appuie sur le bouton « **Aide** » de l'écran d'accueil.

Il indique à l'utilisateur les cinq fonctionnalités principales de l'application et donne quelques informations sur leur fonctionnement.

## 7 Adaptation du cahier des charges initial

Le cahier des charges initial décrit au chapitre 4 de ce document a été modifié tout au long de la phase de développement. Toutes ces modifications ont été effectuées suite aux nombreuses discussions lors des séances entre le réalisateur de projet et le professeur. Les procès verbaux de ces séances sont disponibles en annexes de ce document au chapitre 12.5 - Procès-verbaux.

### 7.1 Suppression de la fonctionnalité de réalité augmentée

Nous avons décidé de supprimer la fonctionnalité de « **Réalité augmentée** ». Une première raison pour cette décision est qu'il existe déjà des applications fournissant ce genre de service. Une deuxième est que le but de cette application mobile est aussi de se démarquer des logiciels existants. Cependant, nous n'estimions pas que la réalité augmentée serait le point fort de l'application et qu'il fallait plutôt se concentrer sur une autre fonctionnalité. La troisième et dernière raison est la durée du développement. Après avoir effectué quelques recherches sur Google afin de trouver d'éventuels tutoriaux sur le développement de réalité augmentée sur iPhone, j'ai dû constater qu'il n'existait pas vraiment de documentation intéressante à ce sujet. Nous avons donc décidé avec mon professeur d'éliminer cette fonctionnalité.

### 7.2 Amélioration de la fonctionnalité d'alerte

Figure 40 : Icône - Alerte rapide



En revanche, nous avons décidé de mettre davantage l'accent sur la fonctionnalité d'« **Alerte rapide** ». Cette dernière a énormément évolué depuis le début du projet. A la base, il était prévu d'afficher des annotations représentant les sauveteurs sur la « **Carte** » du domaine de l'interface mobile. Or, les résultats que nous aurions obtenus n'auraient jamais été concluants. Un sauveteur est toujours en mouvement : sa position devrait être continuellement mise à jour afin que la fonctionnalité soit efficace. Hélas, il faudrait que les sauveteurs mettent à jour leur position à travers un interface utilisateur à peu près toutes les cinq minutes, ce qui est ingérable. Nous avons donc essayé de voir la fonctionnalité d'alerte sous un angle différent et de l'implémenter comme indiqué au chapitre 6.2 - Application mobile - Alerte rapide.

## 7.3 Remplacement de la planification d'un itinéraire par le statut des points d'intérêt

Figure 41 : Icône - Statut



Il était également prévu d'implémenter une fonctionnalité de « **Planification d'un itinéraire** » à travers la « **Carte** » du domaine skiable. Or, deux semaines avant la fin du travail de bachelor, nous avons remarqué que l'application ne gère pas le « **Statut** » des remontées mécaniques et des pistes. Cette fonctionnalité est primordiale car il arrive souvent que les conditions météorologiques ne permettent pas la mise en fonction de certaines remontées mécaniques ou que le danger d'avalanche oblige la station de fermer certaines pistes. Nous avons donc estimé plus important que les usagés des pistes de ski soient informés sur ces événements que de pouvoir planifier des itinéraires à travers la carte du domaine skiable.

## 7.4 Météo

Figure 42 : Icône - Météo



Une dernière fonctionnalité qui a été ajoutée est la possibilité de consulter la « **Météo** » de la station. La météo reste un des critères les plus importants dans les sports de neige. Un touriste aime être informé sur les conditions météorologiques afin de planifier au mieux son séjour dans la station. C'est pour cette raison que cette fonctionnalité a été implémentée.

## 8 Potentiel de l'application

### 8.1 iSummerSports

Comme indiqué lors de la préface de ce document, notre application s'appelle iSnowSports. Cette dernière permet pour le moment de gérer tous les points d'intérêt relatifs aux sports d'hiver.

Il faut savoir que la manière dont l'application a été conçue permet d'ajouter aisément d'autres types de points d'intérêt. Nous pourrions donc tout à fait imaginer d'appliquer ce genre de concept aux sports d'été et de créer ainsi une application « **iSummerSports** », regroupant des activités sportives telles que la randonnée, les via ferrata, les parcours en VTT, en vélo de descente, etc. Dans ce cas, l'interface mobile de l'application permettrait d'afficher sur la carte les différents parcours permettant de pratiquer les sports cités précédemment.

### 8.2 Utilisation possible dans le monde entier

Les points d'intérêt qu'un administrateur enregistre dans la base de données à travers l'interface administrateur décrite au chapitre 6.1 pourraient se trouver n'importe où dans le monde. Il suffit de préciser l'emplacement du point d'intérêt sur la carte du monde, ce qui signifie que n'importe quelle station de ski pourrait tirer profit de cette application.

### 8.3 Apport de l'application dans le milieu du tourisme

Voici le slogan qui pourrait se trouver au départ des remontées mécaniques d'une station de ski : « **iSnowSports - La carte du domaine skiable sur votre iPhone** ». Le touriste sort son iPhone de sa poche, télécharge l'application et découvre les différents points d'intérêt du domaine skiable.

L'avantage de la géolocalisation lui permet de savoir à tout moment à quel endroit il se trouve au sein du domaine. Il pourra donc facilement en déduire quelle piste ou quelle remontée mécanique il doit emprunter pour rejoindre la destination désirée.

Cette application peut être très intéressante en cas de brouillard. Il est clair que toutes les pistes d'un domaine skiable sont balisées. Or, les balises des pistes sont parfois trop dispersées et il se peut que le brouillard empêche le touriste de distinguer s'il se trouve encore sur la piste ou s'il a quitté cette dernière involontairement. Là aussi, le touriste peut facilement s'orienter grâce à la fonctionnalité de géolocalisation ainsi qu'à la carte du domaine.

Il est aussi intéressant pour le touriste de savoir où il peut se désaltérer et savourer un bon dîner. L'affichage des restaurants sur la carte du domaine lui permet de savoir où ces derniers se trouvent dans le domaine skiable et comment il peut les atteindre. Les numéros de téléphones des restaurants sont également référencés et permettent au touriste d'appeler directement le restaurant en question pour effectuer une réservation.

## 9 Extensions possibles

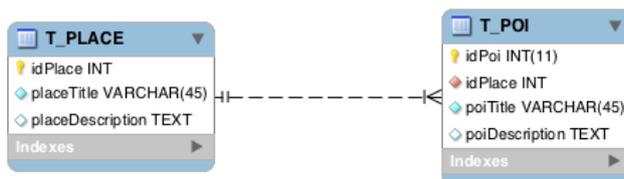
### 9.1 Globalisation de l'application

Rappelez-vous que l'interface administrateur permet pour le moment d'enregistrer les points d'intérêt d'une station de ski dans une base de données. Mais que se passe-t-il lorsque plusieurs stations désirent mettre en place ce système ? A ce moment-là, il faudrait créer une base de données différente pour chaque station afin d'éviter que la carte de l'interface mobile affiche des points d'intérêt n'ayant rien en commun avec une station en particulier. Cette solution ne serait en fait pas très idéale.

Il serait préférable de faire en sorte que l'utilisateur puisse, à travers un écran « **Préférences de l'application** », sélectionner la station pour laquelle il souhaite obtenir des informations. Ainsi, la carte du domaine skiable afficherait uniquement les points d'intérêt en relation avec la station sélectionnée.

Les conséquences de cette modification seraient l'ajout d'une table T\_PLACE (représentant les différentes stations) dans la base de données. Cette dernière devrait encore être reliée à la table T\_POI par une relation 1 : n (une station peut contenir n points d'intérêt et un point d'intérêt ne peut être lié qu'à une et une seule station). Le modèle physique de données peut être consulté en annexe au chapitre 12.2 - Diagrammes. La figure ci-dessous représente le changement qui devrait être apporté à ce modèle afin de mettre en place cette nouvelle fonctionnalité.

**Figure 43 : Modèle physique de données - Changements pour globalisation**



### 9.2 Planning et agenda des sauveteurs

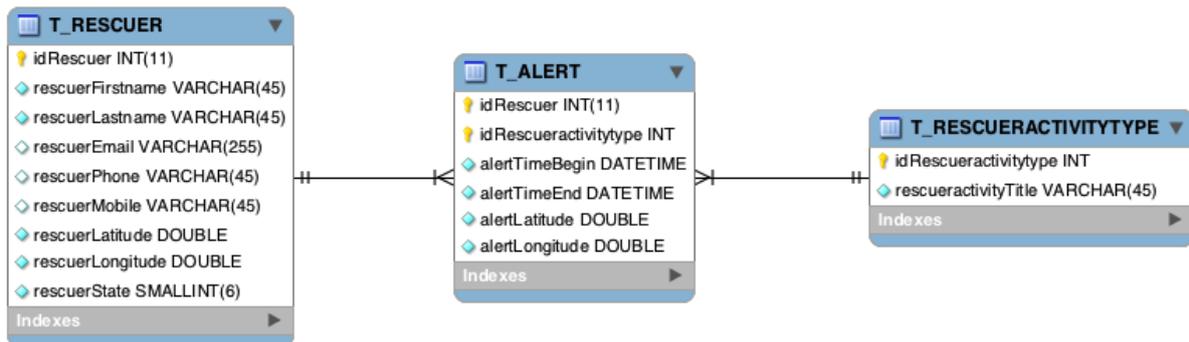
Une fonctionnalité qui devrait encore être exploitée davantage est celle de l'« **Alerte rapide** ». Pour l'instant, les alertes données par les usagés des pistes de ski s'affichent sur une carte qui peut être consultée par le centre de sauvetage à travers l'interface administrateur décrite au chapitre 6.1 de ce document. Le centre de sauvetage peut également donner la responsabilité d'une opération de sauvetage en sélectionnant un sauveteur parmi une liste donnée.

Les sauveteurs d'un domaine skiable sont souvent impliqués dans de nombreuses tâches et ne sont donc pas toujours disponibles pour une mission de sauvetage. Il serait donc intéressant de pouvoir gérer leur disponibilité à travers un agenda. A partir du moment où cet agenda existe, les missions de sauvetage ne pourraient plus être attribuées qu'aux sauveteurs disponibles.

La mise en place d'un agenda pour les sauveteurs impliquerait l'ajout et la modification des tables suivantes dans la base de données :

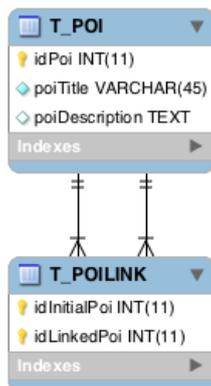
- Ajout de la table T\_RESCUERACTIVITYTYPE (types d'activités des secouristes)
- Modification de la table T\_ALERT
  - Ajout du champ « alertTimeBegin » et « alertTimeEnd »
  - Ajout des clés étrangères « idRescuer » et « idRescueractivitytype », également clés primaires de cette table

Figure 44 : Modèle physique de données - Changements pour agenda des sauveteurs



### 9.3 Planification d'un itinéraire

Figure 45 : Modèle physique de données - Changements pour planification d'un itinéraire



La fonctionnalité de « **Planification d'un itinéraire** » avait été supprimée et remplacée par celle de « **Statut** » des remontées mécaniques et des pistes de ski (voir chap. 7.3 - Remplacement de la planification d'un itinéraire par le statut des points d'intérêt). Or, cette fonctionnalité est tout de même intéressante pour le touriste. Il faudrait que ce dernier puisse indiquer un point d'arrivée sur la carte du domaine et de pouvoir calculer l'itinéraire à partir du lieu où il se trouve actuellement. Ceci lui permettrait de savoir quelle remontée mécanique ou quelle piste il doit emprunter pour rejoindre le lieu désiré.

Pour que la planification d'un itinéraire puisse fonctionner, il faut ajouter une table T\_POILINK dans la base de données, permettant de savoir comment les différents points d'intérêt sont liés entre eux. Ce changement est représenté sur la Figure 45. Grâce à l'ajout de cette nouvelle table, nous saurons qu'à partir du point d'intérêt avec l'identifiant « 1 », il est possible d'atteindre par exemple les points d'intérêt avec les identifiants « 2 », « 5 » ou « 9 ». A partir de ce moment-là, il faudrait utiliser un graph ainsi qu'un algorithme récursif permettant de trouver quel est le chemin à emprunter par le touriste afin d'atteindre le point d'intérêt désiré.

## 9.4 Diversification des points d'intérêt

Pour l'instant, l'application permet la gestion des points d'intérêt principaux d'un domaine skiable tels que les pistes, les remontées mécaniques ainsi que les restaurants. Mais il existe une multitude de types de points d'intérêt qui pourraient encore être ajoutés à cette application :

- Halfpipe
- Piste de luge
- Piste de ski de fond
- Randonnée en peau de phoque
- Randonnée en raquettes
- Snowpark
- etc.

Il serait également intéressant de développer le même type d'application, mais pour les sports d'été comme par exemple la randonnée (voir aussi chap. 8.1 - iSummerSports).

## 10 Conclusion

Les trois mois passés sur ce travail de bachelor m'ont permis de découvrir de nombreux concepts informatiques. La création d'applications mobiles en général ainsi que le développement d'applications iPhone en particulier, m'étaient pour l'instant inconnus. Il fallait donc tout d'abord apprendre un nouveau langage de programmation qui est l'Objective-C.

J'ai remarqué que lorsqu'on développe une application mobile, la philosophie de développement est très différente que lors de la conception d'un logiciel déployé sur un ordinateur portable ou fixe. Ces derniers sont connectés à Internet en permanence, tandis que les Smartphones, eux, peuvent parfois se trouver à des endroits dépourvus d'une telle connexion. Ceci change énormément la manière de concevoir un logiciel informatique.

L'application qui a été développée à travers ce travail de bachelor propose déjà de nombreux services aux usagers des pistes de ski. Il faudrait maintenant se focaliser sur ce qui a été fait afin de proposer davantage de fonctionnalités. Il existe encore passablement de services utiles qui pourraient faire partie intégrante de cette application.

Je pense que cette application pourrait énormément intéresser les stations de ski. L'administration des points d'intérêt est très simple et la gestion des alertes un atout considérable permettant d'accélérer et d'améliorer le déroulement d'une opération de sauvetage. La station augmente ainsi sa crédibilité et paraît plus attractive aux yeux des touristes.

Aujourd'hui, les amateurs de sports de neige sont devenus de plus en plus nombreux. En parallèle, la quantité de personnes en possession d'un Smartphone augmente également sans cesse. Je pense qu'il est impératif de saisir ces opportunités et ainsi de développer davantage d'applications mobiles de ce genre. Ces dernières permettront à l'utilisateur d'exercer ses activités sportives de la manière la plus conviviale possible.

# 11 Références

## L'étudiant

Frédéric Aebi  
ch. des Roseaux 23  
CH - 1963 Vétroz

fred.aebi@gmail.com  
+41 27 346 36 79  
+41 79 408 06 74

## Le Professeur

Yann Bocchi  
Institut Informatique de gestion  
HES-SO Valais  
TechnoArk 3  
CH - 3960 Sierre

yann.bocchi@hevs.ch  
+41 27 606 90 13

## 12 Annexes

### 12.1 Accès aux ressources

#### Interface administrateur

URL : <http://www.dreeng.com/tb/admin>  
Nom d'utilisateur : expert  
Mot de passe : isnowsports

#### Base de données

URL : <http://imu209.infomaniak.ch/MySQLAdmin/index.php?db=dreengcom2>  
Serveur : mysql.dreeng.com  
Nom d'utilisateur : dreengcom  
Mot de passe : Smp70M23

#### Service web des points d'intérêts

URL : <http://dreeng.com/tb/webservice.php>  
Paramètre(s) :

- poitype (slope, lift, restaurant, allcoordinates) => retour de tous les points d'intérêt par rapport à un type donné

ex. : <http://dreeng.com/tb/webservice.php?poitype=slope>

- distance (on), userlat (par ex. 46.382212), userlng (par ex. 7.624583) => retour du point d'intérêt le plus proche d'une coordonnée géographique

ex. : <http://dreeng.com/tb/webservice.php?distance=on&userlat=46.382212&userlng=7.624583>

- distance (on), poitype (slope, lift, restaurant), userlat (par ex. 46.382212), userlng (par ex. 7.624583) => retour des distances de tous les points d'intérêt par rapport à un type donné ainsi qu'une coordonnée géographique

ex. : <http://dreeng.com/tb/webservice.php?distance=on&poitype=slope&userlat=46.382212&userlng=7.624583>

#### Service web de la météo

URL : <http://dreeng.com/tb/weather.php>  
Paramètre(s) :

- location => retour des conditions météorologiques par rapport à un lieu donné

ex. : <http://dreeng.com/tb/weather.php?location=Leukerbad>

## 12.2 Diagrammes

Figure 46 : Diagramme de Use Case

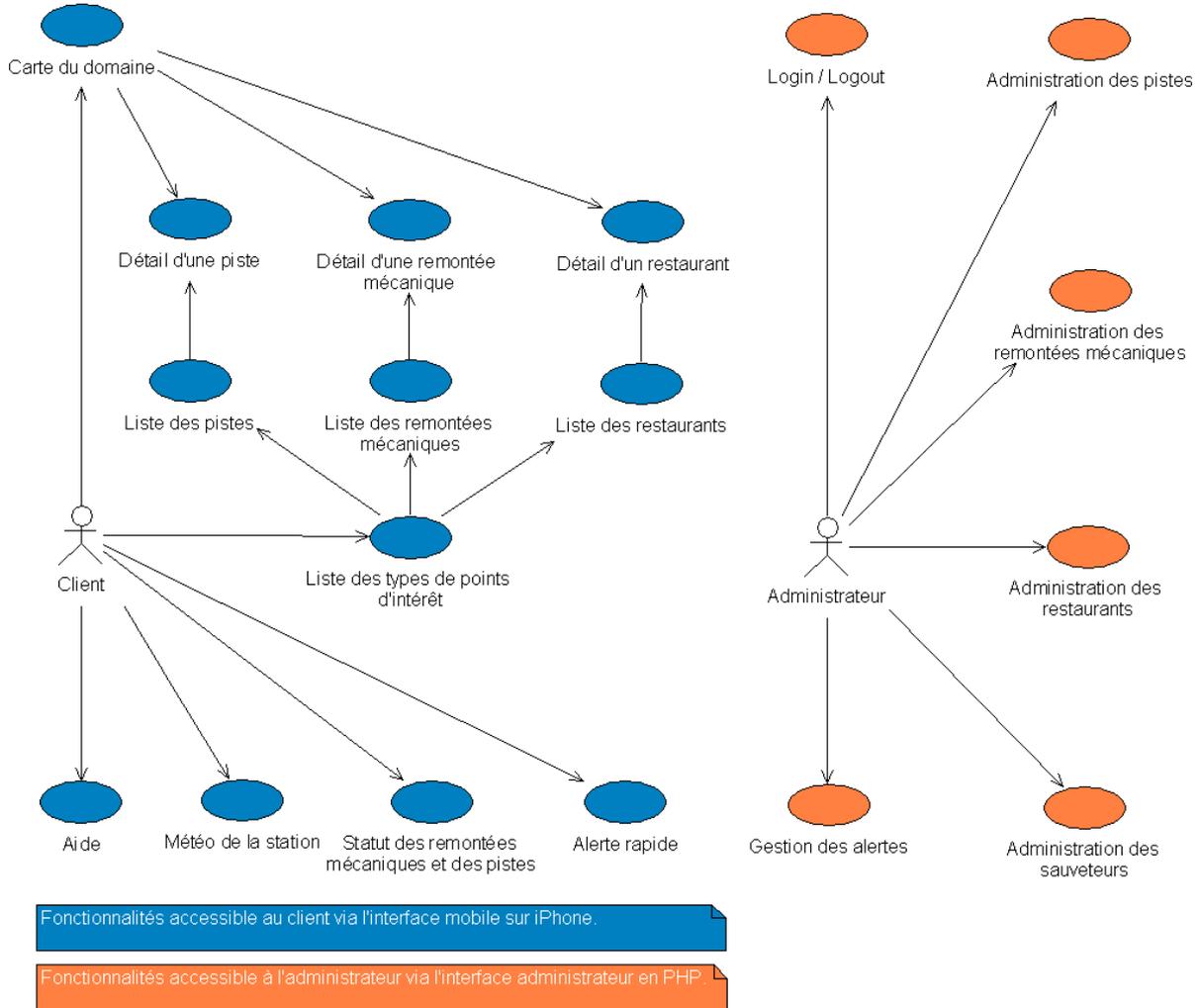
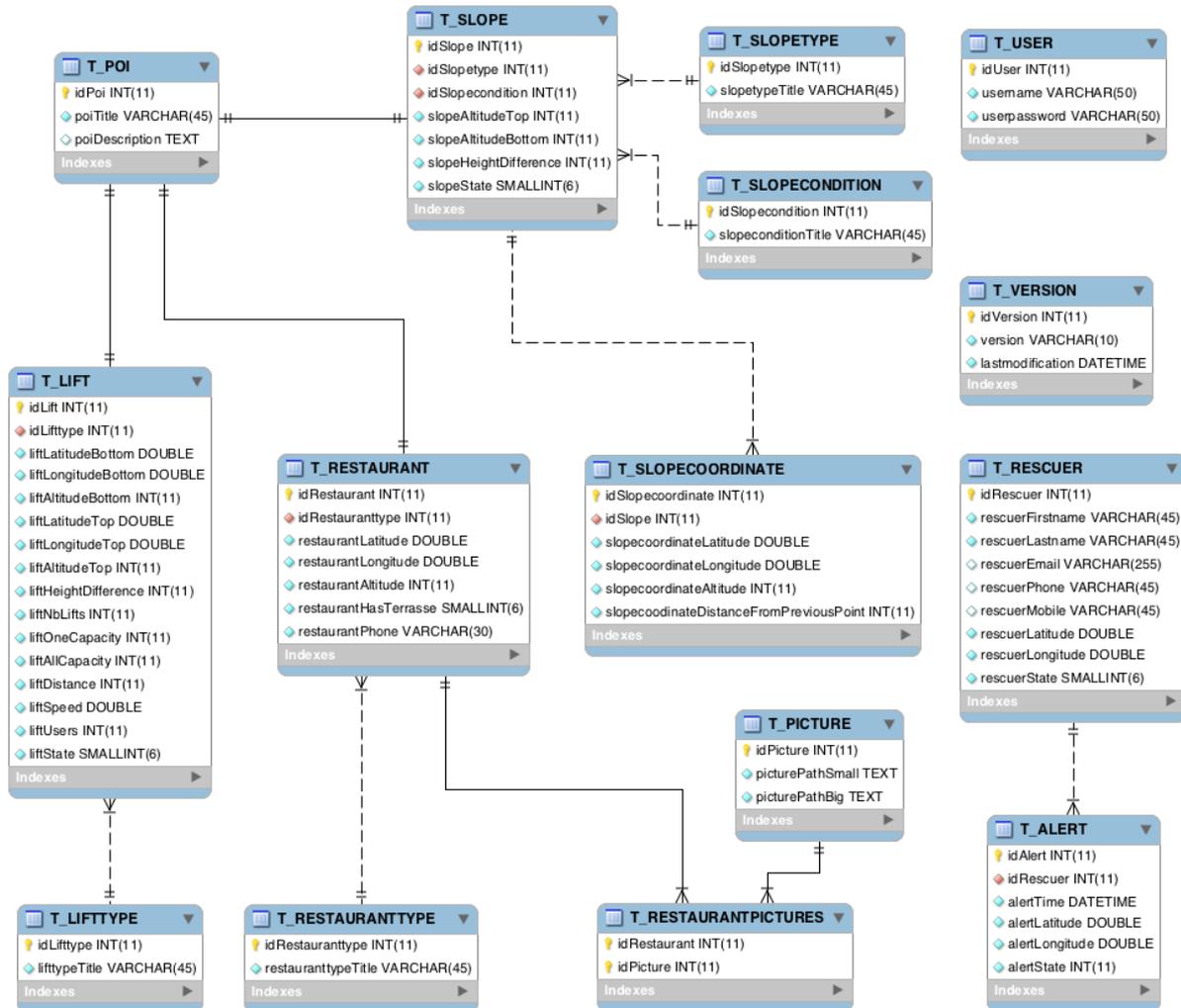


Figure 47 : Modèle physique de données



## 12.3 Tutorial sur le parsing d'un document XML en Objective-C pour iPhone

### Récupération de données depuis un service web

Imaginons maintenant que nous voulons récupérer des données à partir d'un service web. Les DVDs que nous souhaitons voir apparaître dans notre application sont listés dans un fichier XML.

Pour l'exercice, il vous faut copier le fichier « **movies.xml** » depuis le dossier de ressources de ce tutoriel sur votre MAC dans le répertoire `/Users/votre_nom_d_utilisateur_mac/Sites`.

Veillez bien à ce que le partage web soit activé. Vous pouvez vérifier ceci en allant dans les « **Préférences Système - Partage** », et vérifiez que la case à cocher « **Partage web** » soit bien activée. Si cette démarche a été faite correctement, vous devriez pouvoir consulter votre service web à travers l'URL suivante :

`http://localhost/~votre_nom_d_utilisateur_mac/dvds.xml`

Si vous visualisez l'URL sur Firefox, vous devriez obtenir le résultat ci-dessous :

**Figure 48 : Parsing XML en Objective-C - Fichier XML**

```
- <movies>
  - <movie id="1">
    <title>Avatar</title>
    <coverPath>avatar.png</coverPath>
  </movie>
  - <movie id="2">
    <title>Star Wars</title>
    <coverPath>starwars.png</coverPath>
  </movie>
  - <movie id="3">
    <title>Terminator</title>
    <coverPath>terminator.png</coverPath>
  </movie>
</movies>
```

Vous remarquez ici que le nom des balises de chaque DVD (**title** et **coverPath**) correspond exactement aux propriétés de l'interface « **Movie.h** » créée précédemment. Nous verrons pourquoi tout à l'heure.

Nous allons maintenant implémenter la classe « **XMLParser** ». Cette classe permet à l'application de lire le fichier XML, de traiter les balises et ainsi d'affecter les données à un objet « **Movie** ». Rendez-vous sur le répertoire « **Classes** » et faites y un clic droit, puis « **Add / New File...** ». Choisissez « **Cocoa Touch Class** » et sélectionnez « **Objective-C class** ». Entrez « **XMLParser** » dans le champ « **File name** » et cliquez sur « **Finish** ».

Modifions d'abord le fichier « **XMLParser.h** » de manière à lui ajouter trois nouveaux attributs dont deux propriétés. La propriété de type « **Movie** » permet de traiter le film courant détecté par le parseur XML. La propriété de type « **NSMutableArray** » représente le tableau auquel on va ajouter le film courant dès que le parseur XML détecte la fermeture d'une balise movie. Pour terminer, l'attribut de type « **NSMutableString** » est privé à l'interface du

parseur. Ce dernier ne sera donc jamais édité ou consulté ailleurs que dans la classe du parseur lui-même. C'est pourquoi nous ne l'ajoutons pas en tant que propriété. Cet attribut ne sert uniquement à interpréter chacune des valeurs détectées par le parseur. Voici le fichier « **XMLParser.h** ». N'oubliez pas d'importer l'interface « **Movie.h** » :

**Figure 49 : Parsing XML en Objective-C - XMLParser.h**

```

#import <UIKit/UIKit.h>
#import "Movie.h"

@interface XMLParser : NSObject {
    Movie *aMovie;
    NSMutableArray *movies;
    NSMutableString *currentElementValue;
}

@property (nonatomic, retain) Movie *aMovie;
@property (nonatomic, retain) NSMutableArray *movies;

- (XMLParser *) initWithXMLParser;

@end
  
```

Comme vous pouvez l'apercevoir dans le fichier « **XMLParser.h** », nous avons rajouté une méthode « **initWithXMLParser** ». Cette méthode représente le constructeur. Nous allons le définir de la manière suivante dans le fichier « **XMLParser.m** » :

**Figure 50 : Parsing XML en Objective-C - XMLParser (constructeur)**

```

- (XMLParser *) initWithXMLParser {

   NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];
    NSData *data = [defaults objectForKey:@"movies"];
    NSArray *arrayMovies = [NSKeyedUnarchiver unarchiveObjectWithData:data];
    self.movies = [arrayMovies mutableCopy];

    [super init];
    return self;
}
  
```

Rappelez-vous du chapitre « **Persistance des données** ». Dans le constructeur, nous allons essayer de charger les films déjà disponibles dans la mémoire de l'iPhone. Vous verrez plus tard que lorsque le parseur a terminé la lecture du fichier XML, le tableau des films est sauvegardé directement sur l'iPhone.

Voici ci-dessous l'implémentation de la méthode permettant de débiter la lecture du fichier XML. Si le parseur détecte une balise « **movies** », on instancie le tableau de films. Si le parseur détecte une balise « **movie** », on instancie un nouvel objet « **Movie** ». Les lignes de codes « **NSLog** » permettent uniquement l'affichage dans la console de ce qui se passe au cours de la compilation. Elles sont donc optionnelles, mais tout de même intéressantes pour la compréhension du code.

**Figure 51 : Parsing XML en Objective-C - XMLParser (didStartElement)**

```

- (void)parser:(NSXMLParser *)parser didStartElement:(NSString *)elementName
    namespaceURI:(NSString *)namespaceURI qualifiedName:(NSString *)qualifiedName
    attributes:(NSDictionary *)attributeDict {
    if([elementName isEqualToString:@"movies"]) {
        movies = [[NSMutableArray alloc] init];

        NSLog(@"Début de la lecture des films");
    } else if([elementName isEqualToString:@"movie"]) {
        aMovie = [[Movie alloc] init];

        NSLog(@"Début de la lecture du film avec ID = %i", [[attributeDict objectForKey:@"id"] integerValue]);
    }
}
  
```

Ajoutez également la méthode suivante dans le fichier « **XMLParser.h** ». Elle permet de lire individuellement chaque élément du fichier XML. Elle vérifie si le nom d'une balise XML correspond à une propriété de l'interface « **Movie.h** ».

**Figure 52 : Parsing XML en Objective-C - XMLParser (foundCharacters)**

```

- (void)parser:(NSXMLParser *)parser foundCharacters:(NSString *)string {
    if(!currentElementValue)
        currentElementValue = [[NSMutableString alloc] initWithString:string];
    else
        [currentElementValue appendString:string];
    NSLog(@"Valeur lue = %@", currentElementValue);
}
  
```

Il nous faut encore définir la méthode qui se charge de terminer la lecture d'un élément du fichier XML. La Voici :

**Figure 53 : Parsing XML en Objective-C - XMLParser (didEndElement)**

```

- (void)parser:(NSXMLParser *)parser didEndElement:(NSString *)elementName
  namespaceURI:(NSString *)namespaceURI qualifiedName:(NSString *)qName {

    if([elementName isEqualToString:@"movies"]) {
        NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];
        NSData *data = [NSKeyedArchiver archivedDataWithRootObject:movies];
        [defaults setObject:data forKey:@"movies"];

        NSLog(@"Fin de la lecture des filmes");

        return;
    }

    if([elementName isEqualToString:@"movie"]) {
        [movies addObject:aMovie];
        [aMovie release];
        aMovie = nil;

        NSLog(@"Fin de la lecture du film");
    } else {
        [aMovie setValue:currentElementValue forKey:elementName];
    }

    [currentElementValue release];
    currentElementValue = nil;
}
  
```

Si le parseur détecte la balise « **movies** », on sauvegarde le tableau de films dans la mémoire de l'iPhone. Si le parseur détecte la balise « **movie** », on ajoute l'objet « **Movie** » courant au tableau de films. Sinon on ajoute la valeur courante à la propriété correspondante de la classe « **Movie** ».

A la fin du fichier « **XMLParser.m** », n'oubliez pas d'ajouter la méthode « **dealloc** » permettant la libération de la mémoire.

**Figure 54 : Parsing XML en Objective-C - XMLParser (dealloc)**

```

- (void) dealloc {
    [aMovie release];
    [movies release];
    [super dealloc];
}
  
```

Il ne reste plus qu'à modifier le fichier « **ListMovieViewController.m** ». La seule chose à faire est d'informer la méthode « **viewDidLoad** » qu'elle doit récupérer les données des films à partir du service web. N'oubliez pas d'importer l'interface « **XMLParser.h** ». Voici

donc ci-dessous la nouvelle implémentation de la méthode « **viewDidLoad** » du fichier « **ListMovieViewController.m** » :

Figure 55 : Parsing XML en Objective-C - ListMovieViewController (viewDidLoad)

```
- (void)viewDidLoad {  
  
    self.title = @"DVD Listing";  
  
    XMLParser *parser = [[XMLParser alloc] initWithXMLString:@""];  
    self.movies = parser.movies;  
  
    if(self.movies == nil || [[self.movies] count] <= 0) {  
        NSURL *url = [[NSURL alloc] initWithString:@"http://localhost/~votre_nom_d_utilisateur_mac/movies.xml"];  
        NSXMLParser *xmlParser = [[NSXMLParser alloc] initWithContentsOfURL:url];  
  
        [xmlParser setDelegate:parser];  
        [xmlParser parse];  
  
        self.movies = parser.movies;  
    }  
  
    UIBarButtonItem *editMovie = [[UIBarButtonItem alloc] initWithTitle:@"Edit"  
                                style:UIBarButtonItemStyleBordered  
                                target:self  
                                action:@selector(toogleEditMovie:)];  
  
    UIBarButtonItem *addMovie = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAdd  
                                target:self  
                                action:@selector(toogleAddMovie:)];  
  
    self.navigationItem.leftBarButtonItem = editMovie;  
    self.navigationItem.rightBarButtonItem = addMovie;  
  
    [editMovie release];  
    [addMovie release];  
}
```

Tout d'abord, on instancie un nouvel objet de type « **XMLParser** » en faisant appel au constructeur implémenté précédemment. Ensuite, on affecte à la propriété « **movies** » du « **ListMovieViewController** » le tableau de films de l'objet « **XMLParser** ». Ce qui veut dire que nous partons du principe que le tableau de films est déjà disponible dans la mémoire de l'iPhone et qu'il n'est donc pas nécessaire de parser de fichier XML (rappelez-vous de l'implémentation du constructeur du fichier « **XMLParser.m** »).

Si le tableau de films est toujours vide, on instancie un objet de type « **NSURL** » en lui passant en paramètre l'URL de notre service web. N'oubliez pas de préciser votre nom d'utilisateur MAC OS.

On va également instancier un objet de type « **NSXMLParser** » en lui passant en paramètre l'objet de type « **NSURL** » qu'on vient de créer.

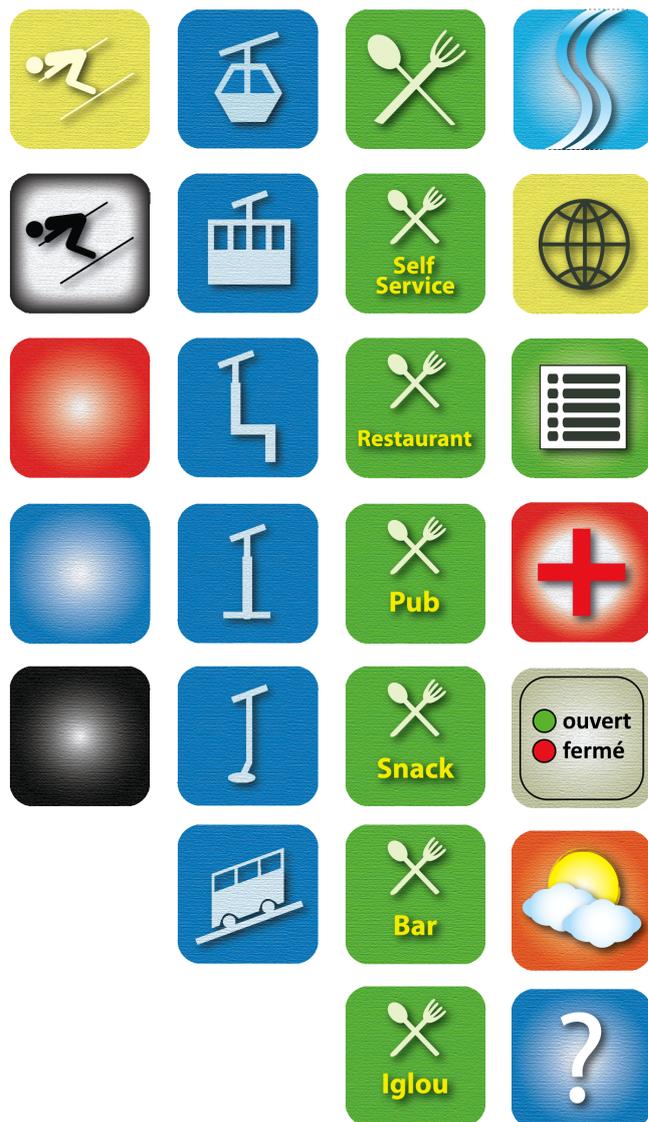
Il ne nous reste plus qu'à informer l'objet de type « **NSXMLParser** » de son delegate qui est l'objet parser de type « **XMLParser** » et d'appeler la méthode « **parse** » qui lui est associée.

Pour finir, on affecte à la propriété « **movies** » du « **ListMovieViewController** » le tableau de films obtenu à travers le parsing.

## 12.4 Icônes de l'application

Toutes les icônes ci-dessous ont été créées à travers le logiciel « Adobe Illustrator ». La première colonne représente toutes les icônes en lien avec les pistes de skis, la deuxième montre celles des remontées mécaniques et la troisième les restaurants. La première icône de la dernière colonne représente le logo de l'application iPhone. Les autres icônes de la quatrième colonne sont celles de l'écran d'accueil.

Figure 56 : Icônes de l'application



## 12.5 Procès-verbaux

### Procès-verbal #1

Date	Lundi, 17 mai 2010
Personnes présentes	Yann Bocchi (Professeur) Frédéric Aebi (Réalisateur du projet)
Personnes absentes	Aucune
Cadre de la séance	<p>Séance de démarrage</p> <p>Définition des points de base pour la réalisation du projet :</p> <ul style="list-style-type: none"> <li>▪ Suivi hebdomadaire (rapport, avancement, etc.)</li> <li>▪ Elaboration d'un planning, cahier des charges, à rendre pour le vendredi 21 mai 2010 à Yann Bocchi</li> <li>▪ Le code doit être réalisé et commenté en anglais</li> <li>▪ Le rapport final peut être réalisé en français ou en anglais et doit être rédigé au fur et à mesure de l'avancement du projet</li> <li>▪ Définition de la prochaine séance le mardi, 18 mai 2010 à 13h30, bureau Yann Bocchi</li> </ul>

## Procès-verbal #2

Date	Mardi, 18 mai 2010
Personnes présentes	<p>Yann Bocchi (Professeur)</p> <p>Olivier Crettol (Contact technique)</p> <p>Frédéric Aebi (Réalisateur du projet)</p>
Personnes absentes	Aucune
Cadre de la séance	<p>Séance de démarrage et de brainstorming</p> <ul style="list-style-type: none"> <li>▪ Validation du procès-verbal #1</li> <li>▪ Définition de la marche à suivre pour débiter le projet             <ol style="list-style-type: none"> <li>1. Analyser l'état de l'art</li> <li>2. Analyser les logiciels existants</li> <li>3. Analyser les points positifs/négatifs d'un développement sur iPhone/Android, car certaines fonctionnalités ne pourront éventuellement pas être développées sur iPhone/Android.</li> <li>4. Chercher et apporter des idées personnelles.</li> <li>5. Définir les fonctionnalités à développer</li> </ol> </li> <li>▪ Décisions prises par rapport aux questions posées par le réalisateur du projet :             <ul style="list-style-type: none"> <li>- La version à développer sera une version générique et non pas une version customisée pour une station de ski spécifique.</li> <li>- Pas de partenariat avec CMA. Eviter aussi toutes influences externes sur le projet afin de minimiser des pertes de temps inutiles.</li> <li>- Pas d'enquêtes ou de questionnaires nécessaires</li> <li>- Mise à jour en temps réel des données importantes</li> <li>- Nouvelles idées : géolocalisation des secouristes ou des pisteurs, géolocalisation et description d'une avalanche à travers les utilisateurs de l'application. Les informations saisies par ce dernier seraient directement envoyées aux secouristes.</li> </ul> </li> <li>▪ Divers :             <ul style="list-style-type: none"> <li>- Le cahier des charges doit être rendu et validé pour le 8 juin 2010. En cas de besoin, prendre rendez-vous avec le professeur et le contact technique d'ici cette date pour parler du cahier des charges.</li> <li>- A la fin du projet, l'application doit être fonctionnelle.</li> </ul> </li> </ul>

### Procès-verbal #3

Date	Jeudi, 10 juin 2010
Personnes présentes	Yann Bocchi (Professeur) Frédéric Aebi (Réalisateur du projet)
Personnes absentes	Aucune
Cadre de la séance	<ul style="list-style-type: none"> <li>▪ Validation du cahier des charges rendu le vendredi 4 juin 2010</li> <li>▪ Définition de la plateforme de développement</li> <li>▪ Démonstration de l'évolution du développement par le réalisateur du projet</li> <li>▪ Définition de la méthodologie de travail SCRUM pour le développement du logiciel. L'évolution de l'application doit être présentée chaque semaine au professeur (intégration du professeur au iPhone Development Provisioning Profiles).</li> <li>▪ Définition des tâches pour la prochaine séance :             <ul style="list-style-type: none"> <li>- Cahier des charges des fonctionnalités</li> <li>- Cas d'utilisation (Use Cases)</li> </ul> </li> </ul>

### Procès-verbal #4

Date	Mardi, 21 juin 2010
Personnes présentes	Yann Bocchi (Professeur) Frédéric Aebi (Réalisateur du projet)
Personnes absentes	Aucune
Cadre de la séance	<ul style="list-style-type: none"> <li>▪ Proposition d'amélioration du cahier des charges des fonctionnalités           <ul style="list-style-type: none"> <li>- Ajout d'un titre à chaque fonctionnalité</li> <li>- Ajout d'un tableau récapitulatif des fonctionnalités avec le temps estimé de ces dernières par rapport à la première tâche « basique » réalisée. (2 fois plus de temps, 3 fois plus de temps, etc.)</li> </ul> </li> <li>▪ Rédaction d'un tutoriel expliquant comment interroger un service web en XML à partir de l'iPhone. Le tutoriel doit être rédigé sur le même principe que celui de Guillaume Fort (Tutoriel iPhone – Une première application iPhone)</li> <li>▪ Démonstration de l'évolution du développement par le réalisateur du projet</li> </ul>

### Procès-verbal #5

Date	Jeudi, 8 juillet 2010
Personnes présentes	Yann Bocchi (Professeur) Frédéric Aebi (Réalisateur du projet)
Personnes absentes	Aucune
Cadre de la séance	<ul style="list-style-type: none"> <li>▪ Discussion à propos du planning de développement           <ul style="list-style-type: none"> <li>- Intervertir les points « planification d'un itinéraire » et « contact par téléphone et fonctionnalité d'alerte »</li> </ul> </li> <li>▪ Itération sur la « fonctionnalité d'alerte »           <ul style="list-style-type: none"> <li>- Intégrer 3 acteurs (client, sauveteur, superviseur) à cette fonctionnalité</li> <li>- Possibilité de définir le nom et l'âge de la personne accidentée, ainsi que la gravité de l'accident (fait par le client)</li> <li>- Notifier le client lors de l'affectation d'un sauveteur et l'informer du temps nécessaire au sauveteur pour arriver au lieu de l'accident (fait par le superviseur)</li> </ul> </li> <li>▪ Démonstration de l'évolution du développement par le réalisateur du projet et propositions d'amélioration par le professeur           <ul style="list-style-type: none"> <li>- Restaurants : donner la possibilité au client de pouvoir téléphoner à un restaurant à partir de l'écran de détail d'un restaurant</li> <li>- Points d'intérêt dans le menu : ajouter la fonctionnalité de tri (alphabétique, distance)</li> <li>- Affichage de la carte : afficher une icône (petit skieur) au point central de chaque piste de ski</li> <li>- Affichage de la carte : utiliser la fonctionnalité du « i » d'information et afficher un écran d'aide lors d'un clic sur ce bouton</li> <li>- Affichage de la carte : utiliser la surbrillance pour le point d'intérêt que l'utilisateur a décidé d'afficher sur la carte</li> <li>- Vue détaillée d'un point d'intérêt : masquer le bouton « Afficher sur la carte » si l'utilisateur arrive sur la vue détaillée via la carte du domaine skiable</li> </ul> </li> <li>▪ Définition de la date de la défense du travail de bachelor le mardi 7 septembre 2010 de 14h00 à 15h00</li> <li>▪ Définition de la date de prochaine séance le lundi 19 juillet 2010 à 15h00</li> </ul>

### Procès-verbal #6

Date	Jeudi, 19 juillet 2010
Personnes présentes	Yann Bocchi (Professeur) Frédéric Aebi (Réalisateur du projet)
Personnes absentes	Aucune
Cadre de la séance	<ul style="list-style-type: none"> <li>▪ Carte du domaine :           <ul style="list-style-type: none"> <li>- Ajouter la possibilité d'interagir avec les points d'intérêt, même si un point d'intérêt spécifique a été sélectionné dans le menu auparavant.</li> <li>- Donner la possibilité à l'utilisateur de centrer la carte du domaine par rapport à son emplacement actuel.</li> </ul> </li> <li>▪ Modifier le titre de certains boutons, labels, etc.</li> <li>▪ Modifier la page d'accueil avec une carte au sommet qui affiche le point d'intérêt le plus proche de la position actuelle de l'utilisateur</li> <li>▪ Modifier la page d'accueil en remplaçant les boutons par des icônes.</li> <li>▪ Itération sur la fonctionnalité d'alerte :           <ul style="list-style-type: none"> <li>- Simplifier l'écran d'« alerte rapide » sur l'application iPhone =&gt; affichage d'une carte contenant une annotation avec l'emplacement actuel de l'utilisateur, ainsi qu'un bouton « Appeler les urgences ». Les coordonnées de l'accident sont envoyées aux urgences lors d'un appel. Ainsi, l'administrateur peut directement gérer le cas via une interface web sur son Ordinateur.</li> </ul> </li> <li>▪ Définition de la date des deux prochaines séances le lundi 26 juillet 2010 à 15h00 et le jeudi 29 juillet 2010 à 15h00</li> </ul>

**Procès-verbal #7**

Date	Mardi, 27 juillet 2010
Personnes présentes	Yann Bocchi (Professeur) Frédéric Aebi (Réalisateur du projet)
Personnes absentes	Aucune
Cadre de la séance	<ul style="list-style-type: none"> <li>▪ Démonstration de la fonctionnalité d’alerte par le réalisateur de projet et propositions d’amélioration du professeur :           <ul style="list-style-type: none"> <li>- Affichage des points d’alerte directement sur une carte</li> <li>- Attribution des missions de sauvetage par rapport à une liste de sauveteurs disponibles</li> </ul> </li> <li>▪ Fixation des objectifs des deux dernières semaines du travail de bachelor :           <ul style="list-style-type: none"> <li>- Ajouter la visualisation de l’état des pistes (ouvertes, fermées)</li> <li>- Débogage de la fonctionnalité de tri des restaurants</li> <li>- Terminer l’interface administrateur</li> <li>- Test général de l’application</li> <li>- Finaliser le rapport</li> </ul> </li> <li>▪ Présentation du rapport final :           <ul style="list-style-type: none"> <li>- Présenter une table des matières complète au professeur pour le jeudi 29 juillet 2010</li> </ul> </li> </ul>

## 13 Table des illustrations

Figure 1 : Mobile Ski .....	5
Figure 2 : iTrailMap .....	5
Figure 3 : RealSki.....	5
Figure 4 : SkiReport .....	6
Figure 5 : Snocator .....	6
Figure 6 : Satski.....	6
Figure 7 : TrialMaps.....	7
Figure 8 : iSki Trail Maps .....	7
Figure 9 : Pénétration des Smartphones dans le marché.....	11
Figure 10 : Parts de marchés des Smartphones .....	12
Figure 11 : Prochain système d'exploitation désiré par rapport au système actuel .....	13
Figure 12 : Couche d'accès aux données - Table restaurant .....	14
Figure 13 : Couche d'accès aux données - Classe restaurant.....	15
Figure 14 : Zone d'administration - Ecrans de login et d'accueil.....	17
Figure 15 : Zone d'administration - Modification / suppression d'une piste.....	18
Figure 16 : Zone d'administration - Insertion d'une piste .....	19
Figure 17 : Zone d'administration - Modification / suppression d'une remontée mécanique .	20
Figure 18 : Zone d'administration - Insertion d'une remontée mécanique.....	21
Figure 19 : Zone d'administration - Modification / suppression d'un restaurant .....	22
Figure 20 : Zone d'administration - Insertion d'un restaurant .....	23
Figure 21 : Zone d'administration - Sauvetage et alertes (carte du domaine) .....	24
Figure 22 : Zone d'administration - Sauvetage et alertes (liste des sauveteurs) .....	24
Figure 23 : Zone d'administration - Sauvetage et alertes (insertion d'un sauveteur) .....	25
Figure 24 : Application mobile - Accueil.....	25
Figure 25 : Application mobile - Carte du domaine skiable .....	26
Figure 26 : Application mobile - Liste des points d'intérêt .....	27
Figure 27 : Application mobile - Liste des points d'intérêt - Pistes.....	27
Figure 28 : Application mobile - Liste des points d'intérêt - Remontées mécaniques .....	27
Figure 29 : Application mobile - Liste des points d'intérêt - Restaurants .....	27
Figure 30 : Application mobile - Détail des pistes.....	28
Figure 31 : Application mobile - Détail des pistes sur la carte .....	28
Figure 32 : Application mobile - Détail des remontées mécaniques.....	29
Figure 33 : Application mobile - Détail des remontées mécaniques sur la carte .....	29
Figure 34 : Application mobile - Détail des restaurants.....	30
Figure 35 : Application mobile - Détail des restaurants sur la carte .....	30
Figure 36 : Application mobile - Alerte rapide .....	31
Figure 37 : Application mobile - Statut.....	32

Figure 38 : Application mobile - Météo .....	32
Figure 39 : Application mobile - Aide .....	33
Figure 40 : Icône - Alerte rapide .....	34
Figure 41 : Icône - Statut .....	35
Figure 42 : Icône - Météo .....	35
Figure 43 : Modèle physique de données - Changements pour globalisation.....	37
Figure 44 : Modèle physique de données - Changements pour agenda des sauveteurs.....	38
Figure 45 : Modèle physique de données - Changements pour planification d'un itinéraire ..	38
Figure 46 : Diagramme de Use Case.....	43
Figure 47 : Modèle physique de données.....	44
Figure 48 : Parsing XML en Objective-C - Fichier XML.....	45
Figure 49 : Parsing XML en Objective-C - XMLParser.h.....	46
Figure 50 : Parsing XML en Objective-C - XMLParser (constructeur) .....	46
Figure 51 : Parsing XML en Objective-C - XMLParser (didStartElement) .....	46
Figure 52 : Parsing XML en Objective-C - XMLParser (foundCharacters).....	47
Figure 53 : Parsing XML en Objective-C - XMLParser (didEndElement) .....	47
Figure 54 : Parsing XML en Objective-C - XMLParser (dealloc).....	47
Figure 55 : Parsing XML en Objective-C - ListMovieViewController (viewDidLoad) .....	48
Figure 56 : Icônes de l'application.....	49

# 14 Bibliographie

## 14.1 Bibliographie

CRETTEL, Olivier : « Utilisation des Smartphones pour informer les usagés des remontées mécaniques et améliorer les flux de skieurs »

## 14.2 Sites Internet

[www.iphonedevlopertips.com](http://www.iphonedevlopertips.com)

[www.iphonedevsdk.com](http://www.iphonedevsdk.com)

[www.iphonesdkarticles.com](http://www.iphonesdkarticles.com)

[www.ipup.fr](http://www.ipup.fr)

[www.layar.com](http://www.layar.com)

[www.nielsen.com](http://www.nielsen.com)

[www.topocoding.com](http://www.topocoding.com)

[www.wikipedia.org](http://www.wikipedia.org)

## 15 Accord de confidentialité

Je déclare, par ce document, que j'ai effectué le travail de bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

-

Lieu et date: \_\_\_\_\_

Signature de l'étudiant : \_\_\_\_\_