

Projet Base de Données

Maillet Laurent Fhal Jonathan Sauvan William König Mélanie
Bregu Erion Maneschi Romain Novak Audrey Boareich Kadedja

12 janvier 2010

Table des matières

Introduction	2
1 Organisation et planification	3
1.1 Organisation	3
1.2 Planification	3
1.3 Analyse	4
1.3.1 Diagramme de classe	4
1.3.2 Diagramme de séquence	7
1.3.3 Diagramme de cas d'utilisation	9
1.3.4 Diagramme d'état	12
2 Développement et utilisation	13
2.1 Choix d'implémentation	13
2.1.1 Outils	13
2.1.2 MVC	13
2.2 Base de données	14
2.2.1 Schéma relationnel	14
2.2.2 Schéma Relationnel-Objet (pour le matériel)	15
2.3 Mode d'emploi	17
2.3.1 Démarrage	17
2.3.2 Fonctionnalités	17
2.3.3 Phase de test	29
Conclusion	31

Introduction

Le but de ce projet est de créer un logiciel pouvant être utilisé par un club de musculation. Comme tout club, MUSQUASH a besoin d'une interface lui permettant de gérer ses adhérents, ses utilisateurs ponctuels, les différentes infrastructures (salle de gym, salle de musculation, terrains de squash...), son personnel, son matériel...

Chapitre 1

Organisation et planification

1.1 Organisation

Pour analyser le sujet, nous avons réalisé plusieurs sortes de diagrammes. Ce rapport contient l'analyse de ces différents diagrammes.

Le rôle de chef du projet a été donné à Maillet Laurent. Les rôles d'acteurs seront définis comme suit :

- Un groupe sera chargé de la partie liée à l'analyse du projet. Il sera constitué de Novak Audrey, König Mélanie, Sauvan William et Bregu Erion.
- Le second groupe sera chargé de la partie liée à l'implémentation du projet. Il sera constitué de Maillet Laurent, Maneschi Romain, Fhal Jonathan et Khedidja Boarech.

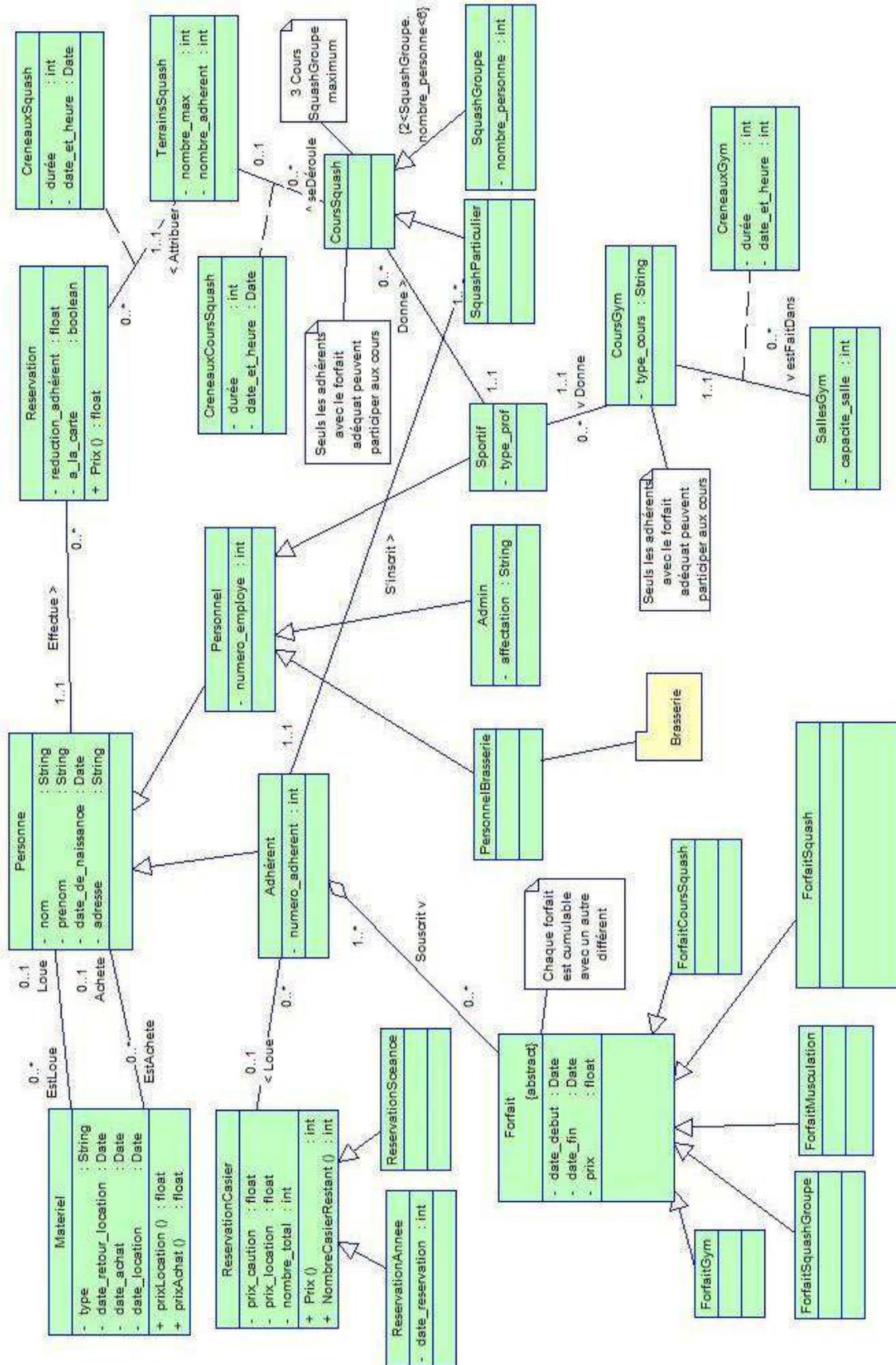
Nous avons réalisé les diagrammes à l'aide du logiciel PowerAMC, qui nous permettra entre autre de gagner du temps pour la partie développement. Afin de mieux organiser le travail au sein du groupe, nous utilisons le gestionnaire de suivi de projet Google Code. Vous pouvez le retrouver à l'adresse suivante : <http://code.google.com/projet-bd-montpellier>

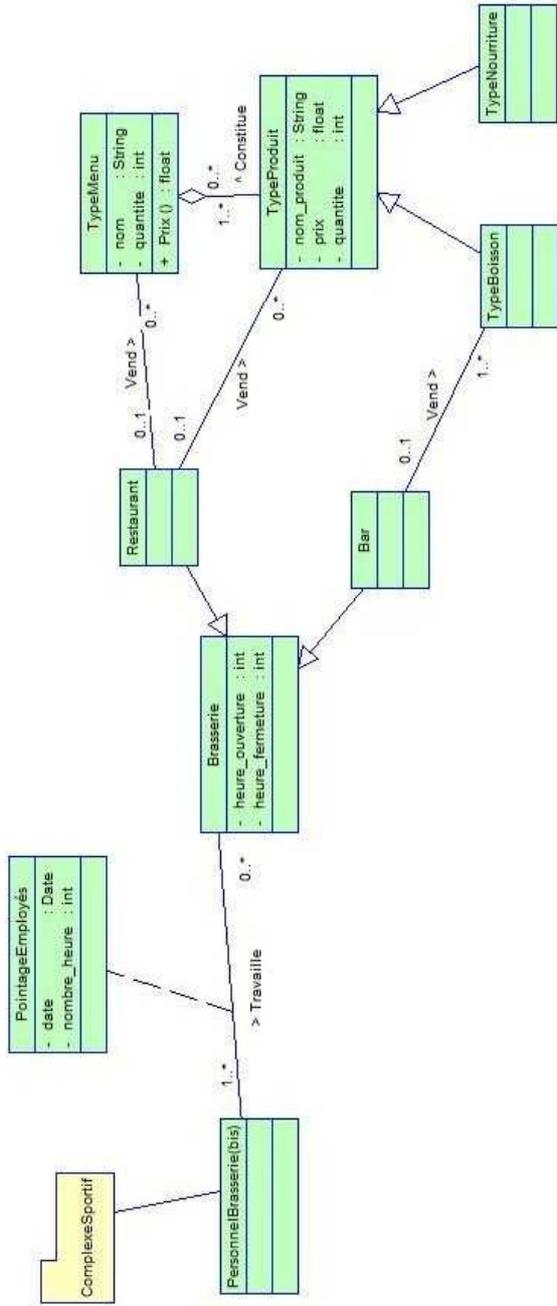
1.2 Planification

ID	Nom	Début	Fin	novembre 2009				décembre 2009	
				09	16	23	30	07	14
1.0	Analyse	13/11/2009	03/12/2009	■					
1.1	Réalisation des diagrammes	13/11/2009	29/11/2009	■					
1.2	Réalisation du premier rapport	28/11/2009	03/12/2009			■			
2.0	Conception	04/12/2009	18/12/2009				■		
2.1	Implementation	04/12/2009	15/12/2009				■		
2.2	Réalisation du rapport	07/12/2009	18/12/2009				■		
2.3	Phase de test	14/12/2009	16/12/2009					■	

1.3 Analyse

1.3.1 Diagramme de classe





Description et contraintes :

Ce diagramme donne un bon aperçu de l'architecture du complexe. Il est tout de même important de rajouter plusieurs choses sur chaque partie :

Adhérents : plusieurs forfaits leurs sont proposés, ils peuvent les cumuler, mais en payer un seul suffit à être adhérent et à profiter de leurs réductions. Pour participer aux cours de gymnastique, cours de squash individuel, cours de squash collectif, accéder à la salle de musculation il faut avoir souscrit respectivement à un abonnement cours de gymnastique, cours de squash individuel, cours de squash collectif, musculation. Par soucis d'esthétisme, certaines actions ne sont pas représentées, ainsi on considère que tous les cours sont planifiés à l'avance par l'administrateur puis les adhérents choisissent d'y participer ou non, sans inscription préalable. En revanche, pour les cours particuliers l'adhérent doit s'inscrire car il faut planifier le cours en fonction de ses disponibilités. Un cours de squash en groupe ne peut être donné que s'il y a entre 2 et 6 participants, sinon il n'a pas lieu. Seuls 3 cours de squash peuvent être donnés simultanément.

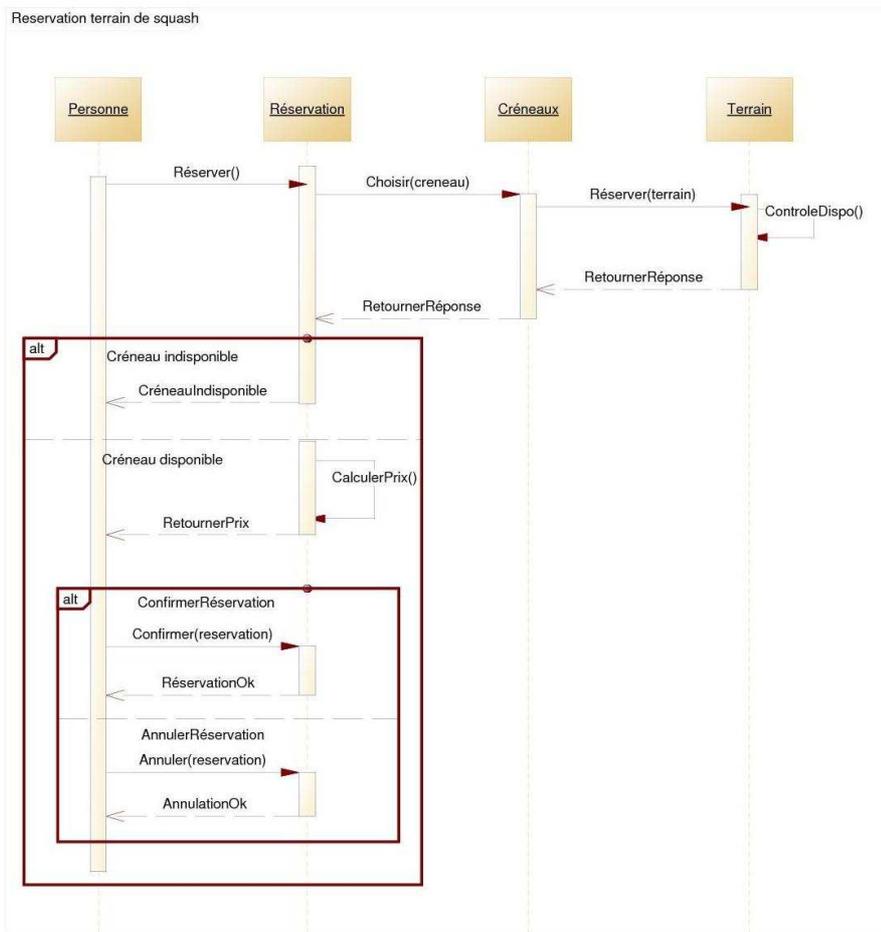
Réservations : le prix des réservations des terrains de squash est variable et sera donc calculé selon plusieurs critères, par abonnement de 10 séances le prix est réduit, les adhérents bénéficient d'une réduction supplémentaire. La gestion de l'abonnement de 10 séances n'est pas représentée car on considère qu'il s'agit d'une carte à poinçon, indépendante du système informatique, mais le calcul du prix de la réservation prend en compte ce choix grâce à un booléen.

Matériels : le client peut aussi acheter ou louer du matériel, qu'il soit adhérent ou non, et une réduction est effectuée pour les adhérents. Le matériel achetable n'est jamais loué, par soucis de qualité. On garde un historique de toutes les ventes et location de matériel.

Brasserie : la brasserie est séparée en deux parties : une partie restaurant ne servant que le midi, et une partie bar ne servant que des boissons. On considère que le bar ne ferme qu'une fois la journée terminée, et que le restaurant n'est ouvert que le midi. La comptabilité des stocks des produits est aussi informatisée. Un employé peut être amené à travailler au bar et au restaurant (mais pas le même jour).

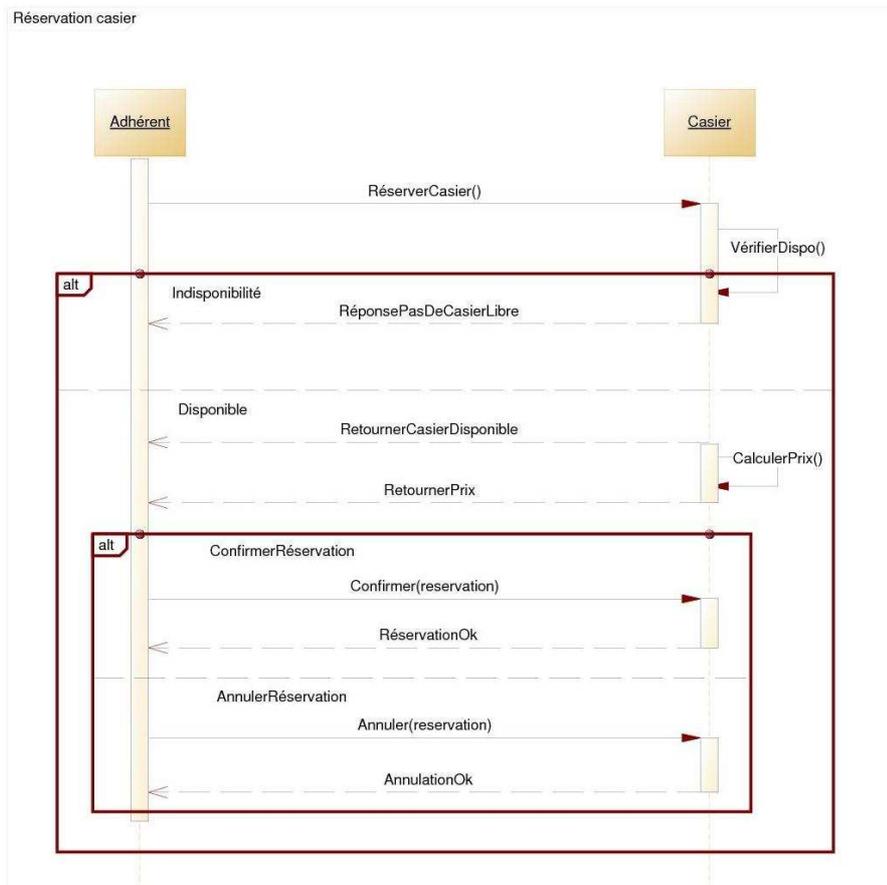
1.3.2 Diagramme de séquence

Réservation de terrain de squash :



Lorsqu'un utilisateur décide de réserver un terrain de squash, cela déclenche le choix d'un créneau. Ainsi le système cherche un terrain disponible. Si aucun terrain n'est libre, l'utilisateur est alors libre de choisir un autre créneau ou de décider d'arrêter sa recherche. Si un terrain est libre, cela déclenche le calcul de son prix, l'utilisateur a alors deux possibilités : régler le prix du terrain ou annuler sa recherche.

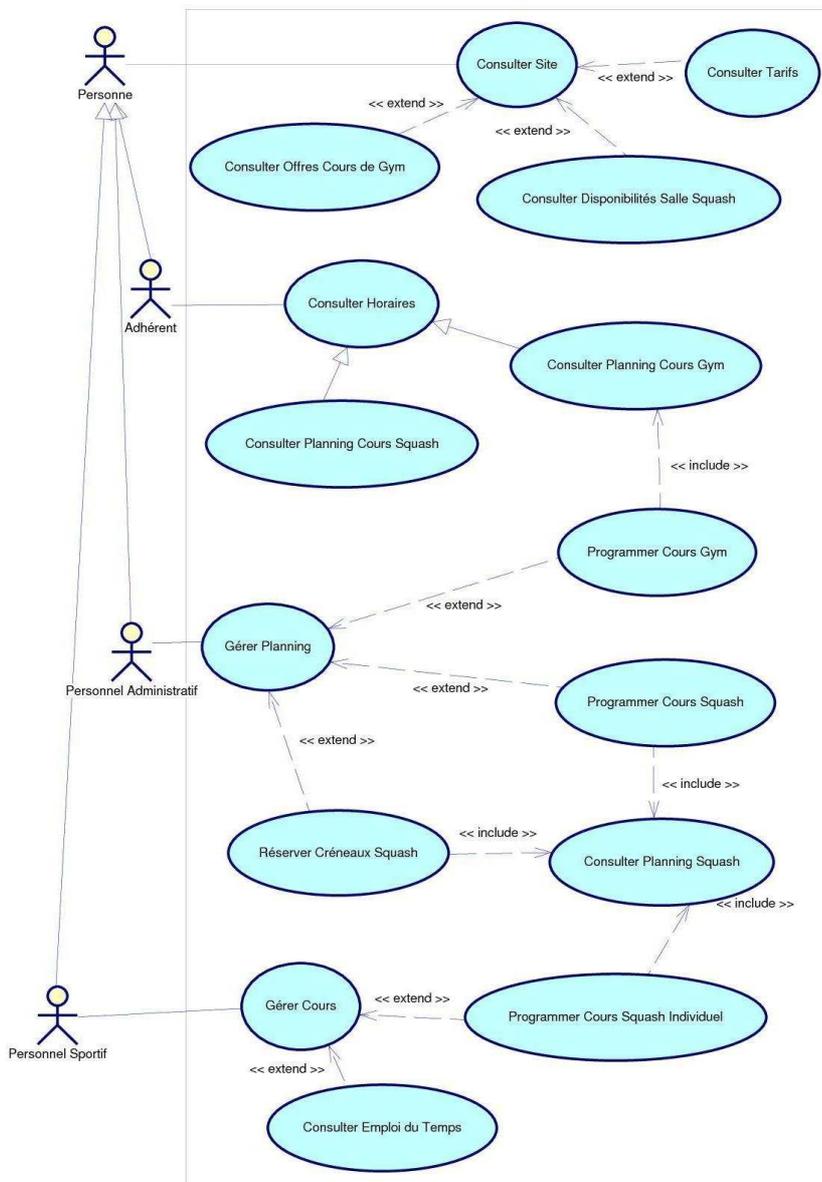
Réservation de casier :



Lorsqu'un adhérent décide de louer un casier, il effectue une recherche, qui déclenche la vérification de la disponibilité d'un casier. Si aucun casier n'est libre, le système envoie un message à l'utilisateur. Sinon, le système retourne le casier libre, puis calcule le prix et le retourne. L'utilisateur a alors le choix de confirmer sa réservation ou non.

1.3.3 Diagramme de cas d'utilisation

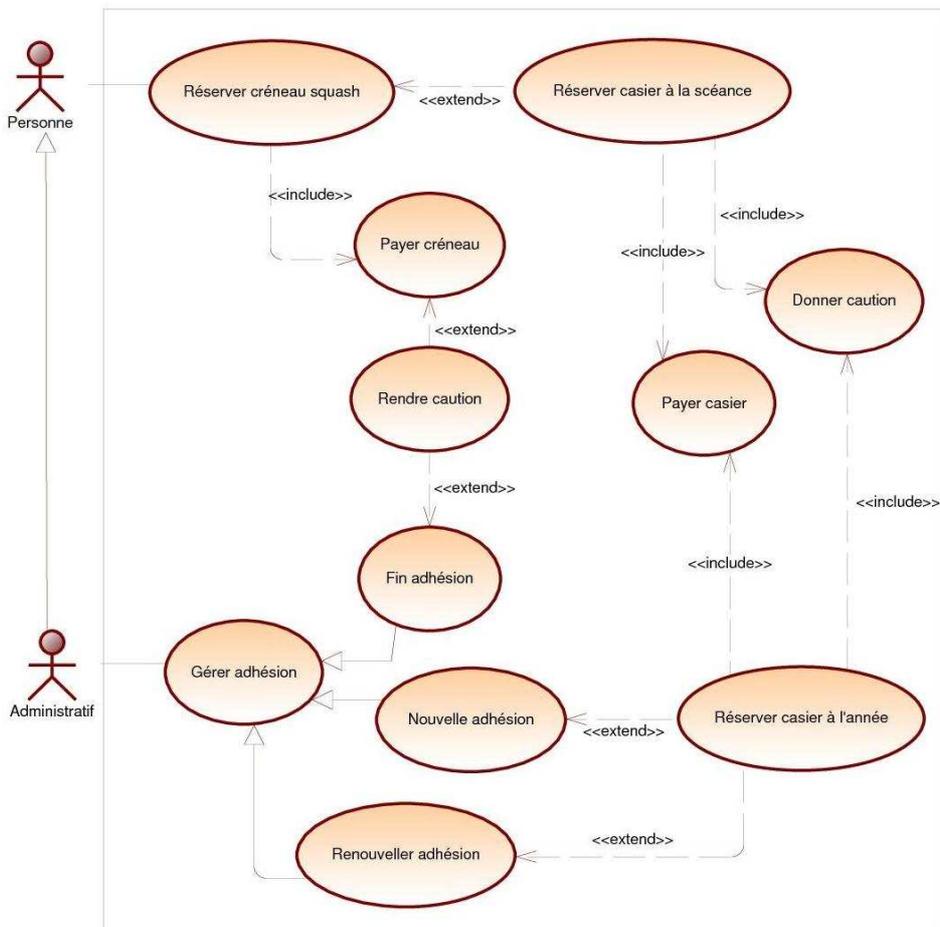
Gestion des cours :



Comme pour le diagramme de classe on suppose qu'un adhérent n'a pas besoin de s'inscrire à un cours car ils sont dans tous les cas programmés par trimestre par le personnel administratif. Cependant les cours individuels sont programmés par le professeur selon sa disponibilité et celle de son élève. Le cas de la musculation n'apparaît pas car il n'y a aucune gestion nécessaire pour les salles et le matériel de musculation.

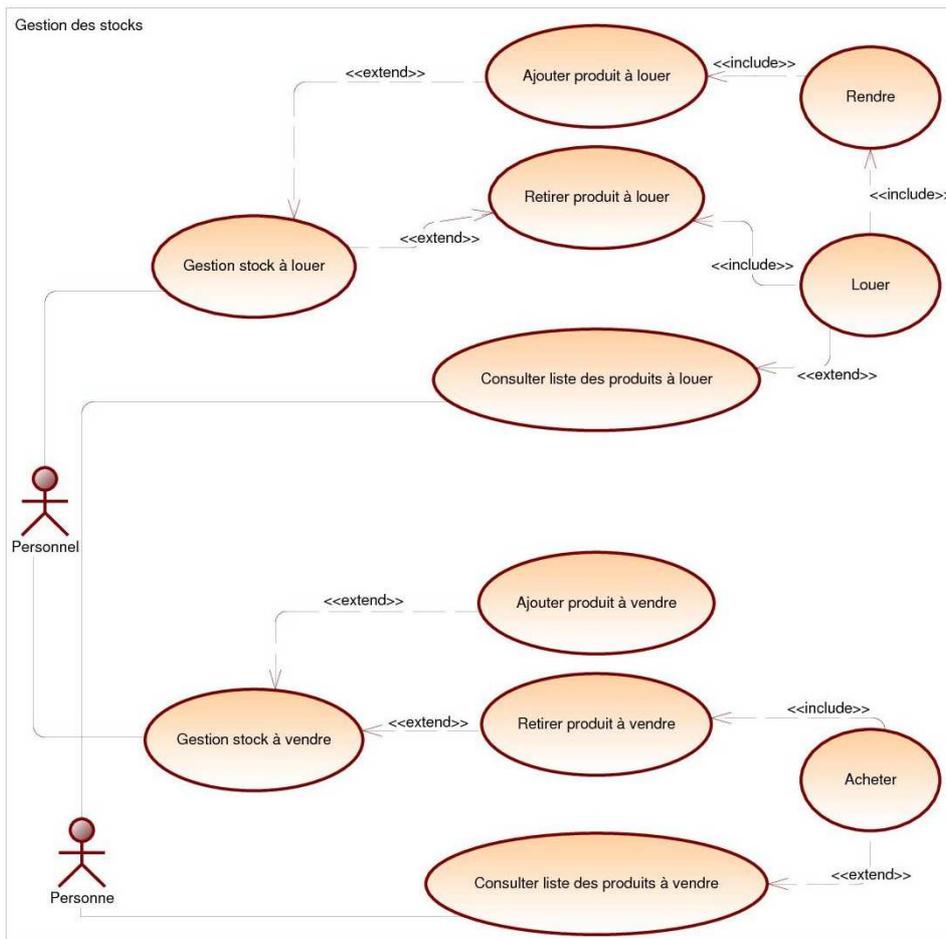
Remarque : on considère comme optionnel le choix de chaque gestion ou consultation. Ainsi, l'utilisateur pourra se trouver sur la page d'accueil et cliquer sur le bouton correspondant à la gestion de son choix. Libre à lui de fermer le programme sans rien faire.

Réservation des terrains de squash et des casiers :



Le paiement des casiers à la séance se fait au moment de la réservation. Pour les casiers à l'année, la réservation et le paiement se font au moment de l'adhésion. La caution des casiers est rendue à la fin de cette adhésion.

Gestion des stocks de matériels :

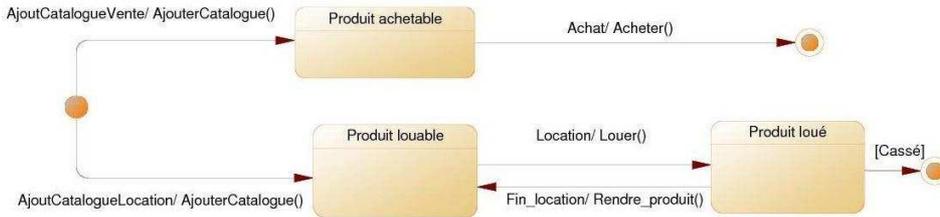


Le personnel est chargé de la gestion du stock, quelque soit le stock : matériel à vendre ou à louer. Il a alors la possibilité d'ajouter ou de retirer du matériel de ce stock. Un utilisateur quant à lui, a la possibilité de louer ou d'acheter ce matériel après avoir consulter la liste de vente ou de location des produits. Dans le cas où il voudrait louer un produit, le produit est alors retiré de la liste des produits à louer jusqu'à ce qu'il le ramène. Dans le cas où il décide d'acheter un produit, ce produit est alors retirer définitivement de la liste.

Pour la gestion des stocks des provisions du bar et du restaurant, le système est équivalent à la vente dans ce diagramme. Donc nous n'avons pas détaillé cette gestion.

1.3.4 Diagramme d'état

Exemple de gestion des produits :



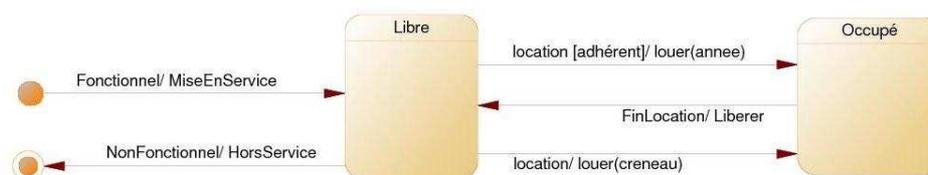
Sur ce diagramme on peut voir que les produits en vente sont supprimés une fois acheté. Et que les produits à louer ont deux états : « état louable », et une fois qu'ils sont loués ils passent en « état loué », une fois qu'ils sont rendus ils repassent en « état louable ». Ils ne sont supprimés que lorsqu'ils ne sont plus en état d'être loués.

Exemple de réservation de terrain :



Un terrain peut avoir deux états : libre ou occupé. Un terrain ne peut être réservé que s'il est libre. Une fois qu'un terrain occupé est libéré, il redevient libre. Les terrains ne sont supprimés du système que lorsqu'ils sont hors service.

Exemple de réservation de casier :



Un casier peut avoir deux états : libre ou occupé. Un casier ne peut être réservé que s'il est libre. Pour réserver un casier à l'année il faut que l'utilisateur soit un adhérent du club, pour un casier réservé à la séance, il n'y a aucune condition préalable. Une fois qu'un casier occupé est libéré, il redevient libre. Les casiers ne sont supprimés du système que lorsqu'ils sont hors service.

Chapitre 2

Développement et utilisation

2.1 Choix d'implémentation

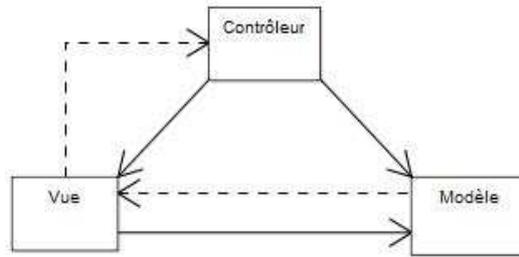
2.1.1 Outils

Nous avons choisi d'utiliser le logiciel de programmation "éclipse", un IDE utilisé au sein même de l'université, afin de bénéficier de toutes ses fonctionnalités. Ce dernier met à notre disposition une interface ergonomique, des outils de mise en place efficace et beaucoup d'assistance au développement pour écrire en JAVA. Ce langage permet aussi une interaction pratique avec notre base de données oracle. Enfin, pour permettre une bonne intégrité du programme lors de son développement, nous utilisons un site internet hébergeant tous les fichiers relatifs au projet. Ainsi, tout le groupe travaille sur les dernières versions et chaque fichier reste très accessible. Plus d'informations sont disponible sur le lien suivant : <http://code.google.com>. Le descriptif du projet et tous les fichiers en rapport peuvent être consultés le lien suivant : <http://code.google.com/p/projet-bd-montpellier/>

2.1.2 MVC

Le Modèle-Vue-Contrôleur (en abrégé MVC, de l'anglais Model-View-Controller) est une architecture et une méthode de conception qui organise l'interface homme-machine (IHM) d'une application logicielle. Ce paradigme divise l'IHM en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface. Cette architecture donne une clarté sans comparaison dans le code et nous permet de développer d'une façon bien plus propre et efficace. Elle facilite aussi les améliorations et les maintenances, permettant même de passer d'une base de données de type SQL à XML en changeant simplement les interactions avec la base.

Notre logiciel s'appuie donc sur ce modèle. Le coeur rassemble les modèles (noms de classe au singulier) et les contrôleurs (noms de classe au pluriel), ainsi que les écouteurs qui vont réagir aux évènements de la partie graphique (la vue). Ainsi, chaque contrôleur possède une interface qui lui correspond (les classes nommées avec "listener"). Donc, quand la partie graphique crée un nouvel objet, on pourra lui associer les évènements d'un contrôleur et implémenter son interface. Par exemple, la création d'un casier (Casier.java) passera par l'ajout et l'implémentation du contrôleur qui lui est associé (CasiersListener.java), donc il aura accès à toutes les méthodes de son contrôleur (Casiers.java) comme l'ajout/modification/suppression d'un horaire pour ce casier. Ces méthodes seront appelées lors d'un ajout, d'une suppression ou d'une modification. Cette méthode respecte la convention JAVA, donc n'importe quel nouvel intervenant au projet connaissant cette convention peut aisément utiliser le programme.



2.2 Base de données

La structure de la base de données ne respecte pas entièrement celle montrée précédemment dans la partie théorique, certaines modifications ont dû être apportées. La base de données possède plus d'une vingtaine de tables, et toutes les contraintes sont respectées en grande partie grâce à de nombreux triggers.

Ainsi, à chaque démarrage du logiciel, si la base n'existe pas, elle est créée, sinon elle est utilisée telle quelle. Nous voulions tout d'abord nous baser sur un schéma relationnel, mais pour éviter la redondance de ligne au niveau des matériels, nous avons choisi une représentation en schéma relationnel-objet.

2.2.1 Schéma relationnel

Personnes

- personne (id_personne, nom, prenom, dateDeNaissance, adresse)
- adherent (num_adherent, #id_personne)
- personnelAdministratif (num_employe, #id_personne, affectation)
- personnelSportif (id_prof, #id_personne, type_sport)

Casiers

- casier (num_casier)
- reservationAnnee (#num_casier, #num_adherent, annee)
- reservationSeance (#num_casier, date_location, heure_debut, heure_fin #id_pers, prix_paye)

Forfaits

- forfait (type_forfait, prix_base)
- souscription (#type_forfait, #num_adherent, date_debut, date_fin)

Terrains Squash

- terrainSquash (num_terrain_squash)
- reservationTerrainSquash (#num_terrain_squash, date_location, heure_debut, heure_fin, minute_debut, minute_fin, #id_personne, prix_paye)

Cours Squash

- coursSquash (id_cours, #id_prof, type_de_cours) //type = indiv ou collectif
- coursSquashIndividuel (#id_cours, #num_adherent)
- coursSquashCollectif (#id_cours, nb_personne)
- inscriptionCoursSqCollectif (#id_cours, #num_adherent)
- creneauCoursSqIndividuel(#id_cours, #num_terrain_squash, date_cours,heure_debut,heure_fin)
- creneauCoursSqColl(#id_cours, #num_terrain_squash, jour, heure_debut, heure_fin, debut_trimestre)

Gym

- salleGym (id_salle, capacite)
- coursGym (type_cours, #id_prof)
- creneauCoursGym (#id_salle, #type_cours, date_cours, heure_debut, heure_fin, debut_trimestre)

Matériel

- materielVente(type_materiel,description, quantite, prix_base)
- materielLocation (num_materiel, type_materiel, date_achat, etat, description, prix_base)

2.2.2 Schéma Relationnel-Objet (pour le matériel)

Types :

type_materiel_loue

num_materiel

tab_materiel_loue

{type_materiel_loue}
num_materiel

type_materiel_achete

{type_materiel}
quantite
prix_base

tab_materiel_achete

{type_materiel_achete}		
{type_materiel}	quantite	prix_base

Tables Relationnel-Objet :

location

num_location	date_location	heure_debut	heure_fin	id_personne	prix_paye	materiel tab_materiel_loue
						num_materiel

vente

num_vente	date_vente	id_personne	prix_paye	materiel tab_materiel_achete		
				type_materiel	qte	prix_base

Voici la liste des contraintes, classées par table, qui sont ainsi vérifiées lors des insertions :

- Table location :
On vérifie que l'objet existe bien dans materielLocation ce qui équivaut à vérifier que ce matériel est bien dans le stock du matériel à louer mais aussi que ce matériel ne soit pas déjà loué.

- Table vente :
De la même manière que pour les locations, on vérifie que l'objet existe bien dans `matérielVente`
- Table souscription :
Avant toute inscription à un forfait, on regarde que les dates de validité du forfait correspondent bien à un forfait de 1, 3, 6 ou 12 mois, mais aussi que l'adhérent ne possède pas déjà ce type de forfait pour une date valide.
Dans le cas où l'adhérent possède ce type de forfait mais qu'il est périmé l'ancien forfait est transféré dans `historiqueSouscription` avant d'insérer le nouveau.
- Table reservationTerrainSquash :
On teste que la réservation d'un terrain dure bien 40 minutes, que le terrain soit libre pour cet horaire (ni cours ni réservation)
Si c'est un adhérent qui réserve, on compte le nombre de terrain réservé par les adhérents pour le même créneau horaire et on interdit l'insertion de tuple si ce nombre est supérieur à 4
Un terrain ne peut pas être réservé par la même personne au même moment
- Table coursSquash :
L'`id_prof` doit être de `type_prof` égal à `squash`
- Table coursSquashIndividuel :
L'`id_cours` doit bien correspondre à un cours individuel
L'adhérent doit avoir un forfait valide pour les cours individuels
- Table creneauCoursSqIndividuel :
On vérifie que le terrain soit libre pour ce créneau horaire (c'est à dire qu'il n'y a pas de réservation pour ce terrain ou pas de cours de squash groupé) mais aussi que le prof soit libre et que l'adhérent soit libre (c'est à dire que l'adhérent n'ait pas d'autre cours ou une réservation pour cet horaire)
- Table coursSquashCollectif :
L'`id_cours` correspond bien à un cours collectif
- Table inscriptionCoursSqCollectif :
L'adhérent doit avoir un forfait valide pour participer aux cours collectifs. Il doit rester des places dans le cours et l'adhérent doit être libre
- Table creneauCoursSqColl :
Il ne doit pas y avoir plus de 3 cours de squash collectif en même temps
Les cours ont lieu une fois par semaine à un horaire précis, et ils ne peuvent être révisés que par trimestre
Il faut également vérifier que le prof désigné soit libre et que le terrain soit libre pour ce créneau
- Table coursGym :
Il faut vérifier que l'`id_prof` correspond à un prof de gym
- Table creneauCoursGym :

Les cours de gym ont lieu plusieurs fois par semaine, avec un prof en particulier (le prof de step donne des cours de step), les horaires sont révisés par trimestre

2.3 Mode d'emploi

2.3.1 Démarrage

Le programme devra se trouver sur l'ordinateur attribué à la gestion du complexe qui servira de serveur. Ce dernier devra aussi posséder une base de données Oracle installée et avoir l'autorisation nécessaire pour la modifier. N'importe quel autre ordinateur connecté en réseau pourra installer et utiliser le logiciel mais en utilisant la même base de données. Dans le cas d'un raccourci placé sur le bureau, l'utilisateur ouvrira le logiciel par son intermédiaire et sera invité à entrer son login, son mot de passe et l'adresse de la base. Il est bien sûr nécessaire de créer au préalable un compte correspondant dans la base de données. Une fois les informations saisies, il suffit de valider.

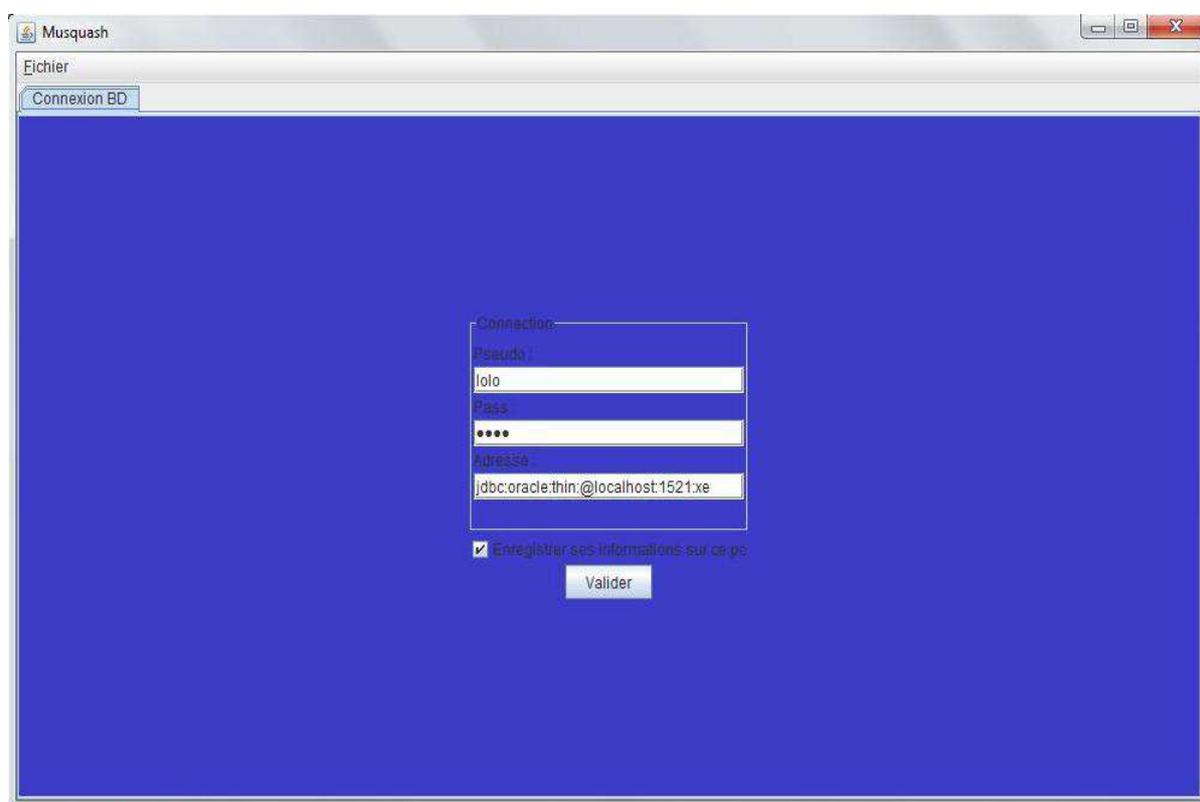


FIG. 2.1 – Page de connexion

2.3.2 Fonctionnalités

Musquash est un logiciel informatique destiné à la gestion d'un complexe sportif abritant des cours de squash, de gym et de musculation. Son interface simple et intuitive donne la possibilité aux administrateurs d'ajouter, supprimer ou modifier des réservations de terrain et des cours, puis de générer un planning.

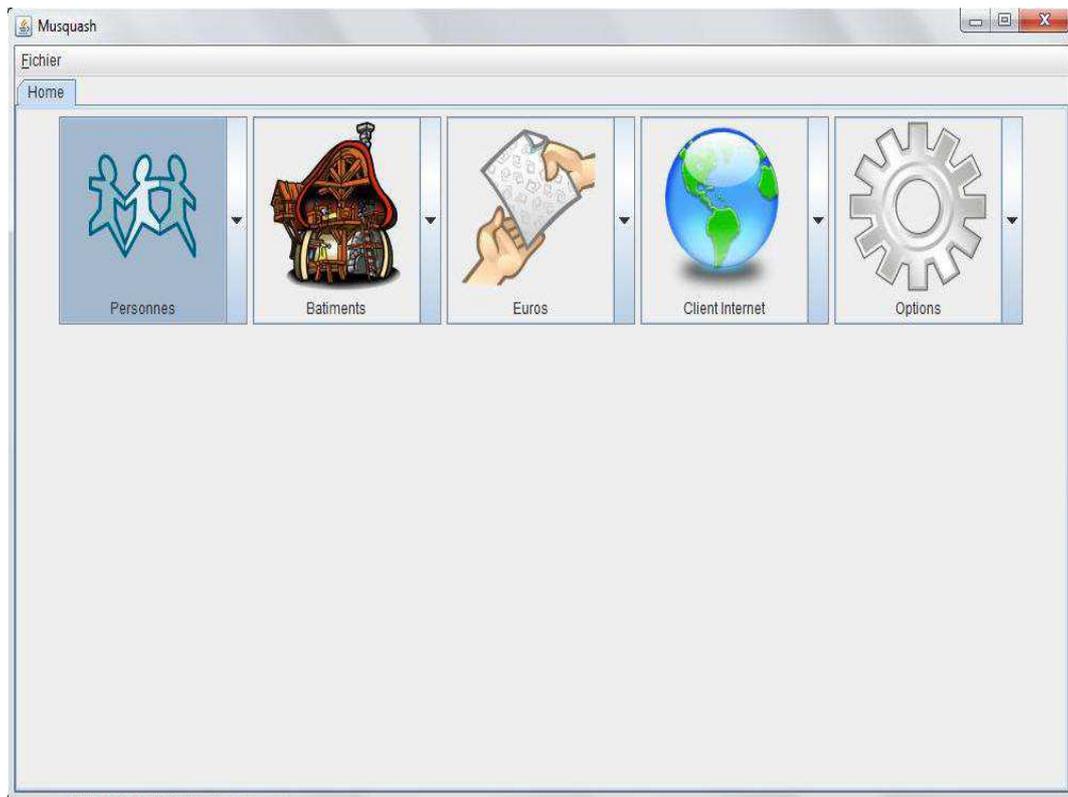


FIG. 2.2 – Page d'accueil

Cinq boutons principaux sont visibles :

Personnes :

Permet ajouter, supprimer ou rechercher des adhérents ou du personnel.

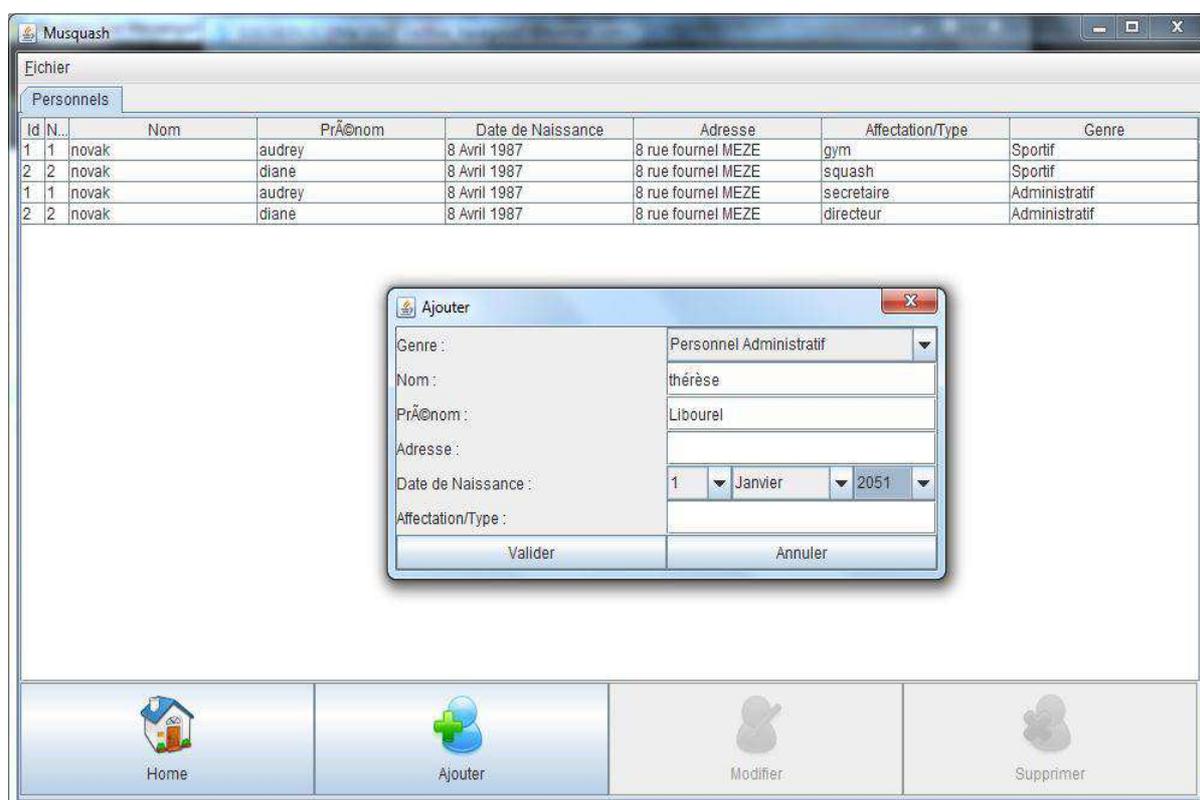


FIG. 2.3 – Gestion des ventes/locations

C'est ici que l'utilisateur gère le personnel du complexe. La liste du personnel existant ou des adhérents est affichée, et il peut ajouter, supprimer ou modifier chacun d'entre eux en le sélectionnant et en cliquant sur le bouton correspondant.

Batiments :

Cette partie gère tout ce qui est en rapport avec la réservation des terrains, des cours et des casiers. C'est ici que l'on attribue les cours aux terrains, que l'on voit leurs disponibilités, que l'on voit le planning des cours et que l'on gère les casiers à la séance/à l'année.

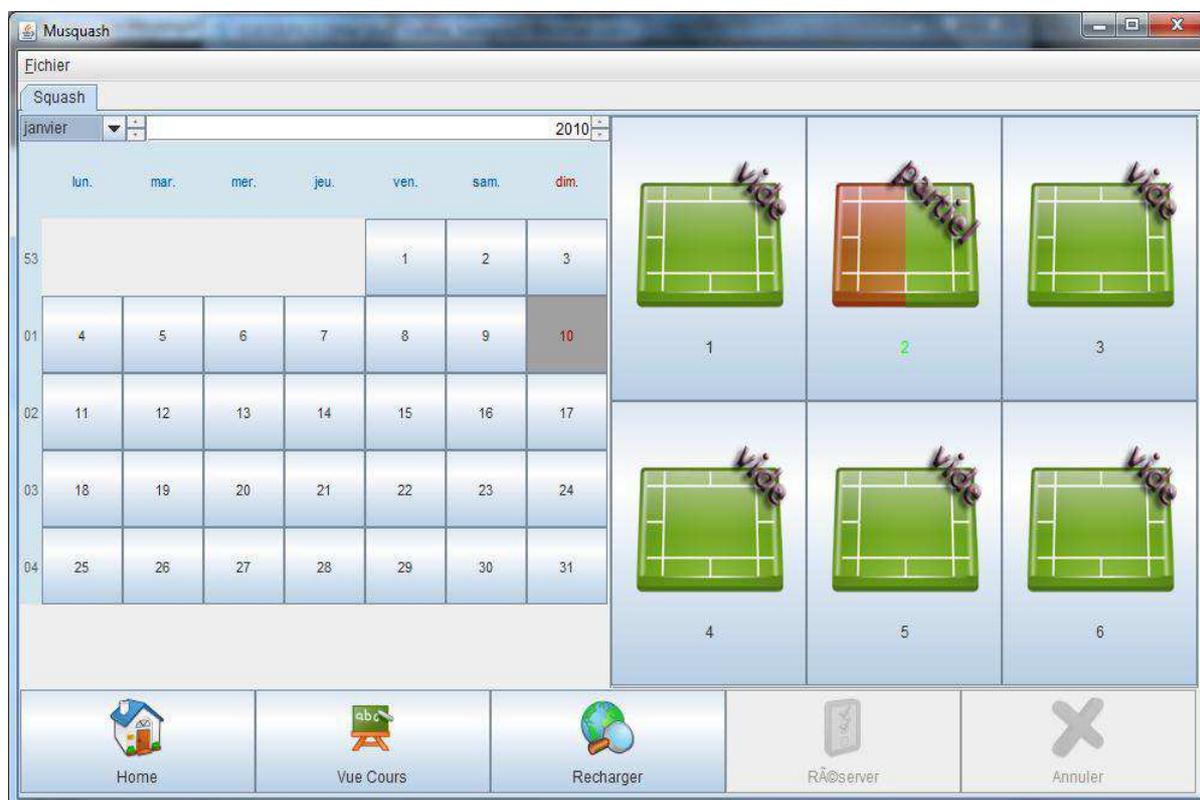


FIG. 2.4 – Gestion des terrains

Ici, l'utilisateur peut voir la disponibilité des 6 terrains selon le jour choisi. Il peut ajouter ou supprimer des réservations grâce aux boutons correspondants.

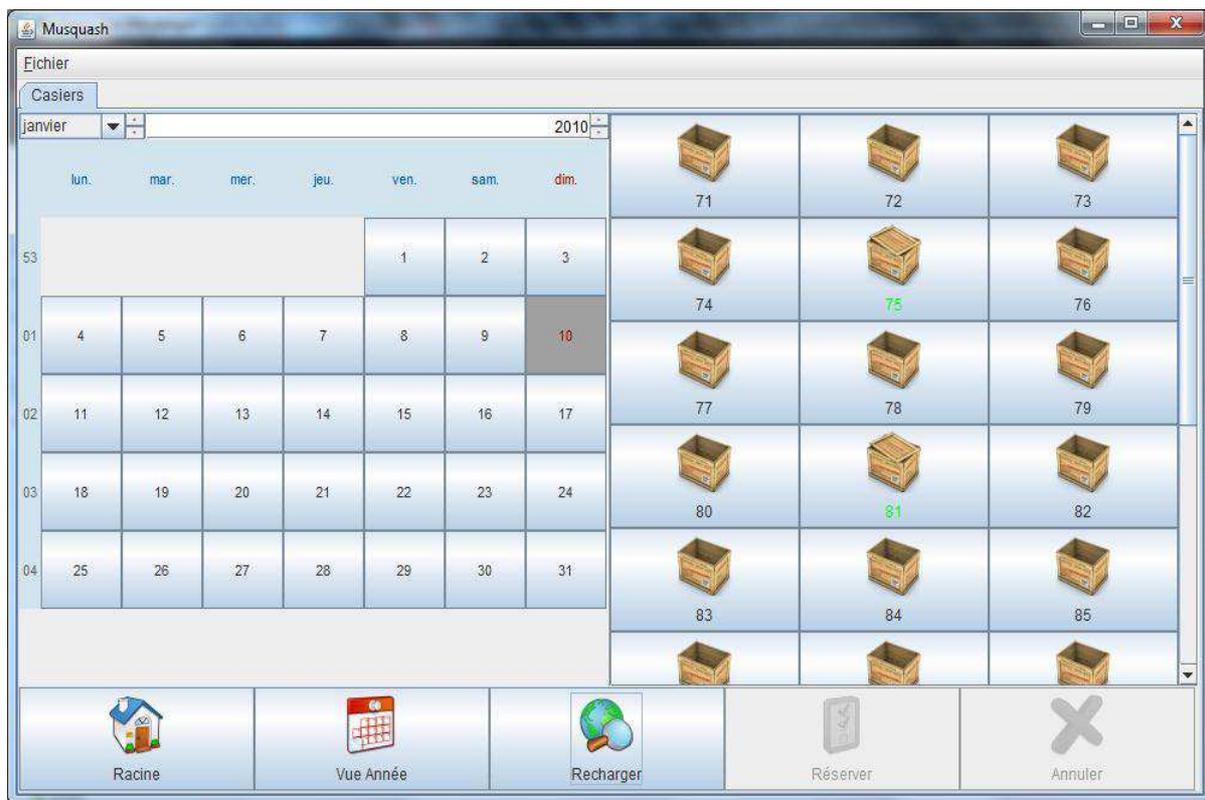


FIG. 2.5 – Gestion des casiers

La gestion des casiers s'effectue de la même façon que les terrains, la liste des casiers s'affichant pour chaque jour du calendrier. Ici encore, l'utilisateur peut ajouter ou supprimer des réservations de casiers.

Euros :

Permet de gérer les stocks de matériel lors d'une location ou d'un achat par un client.

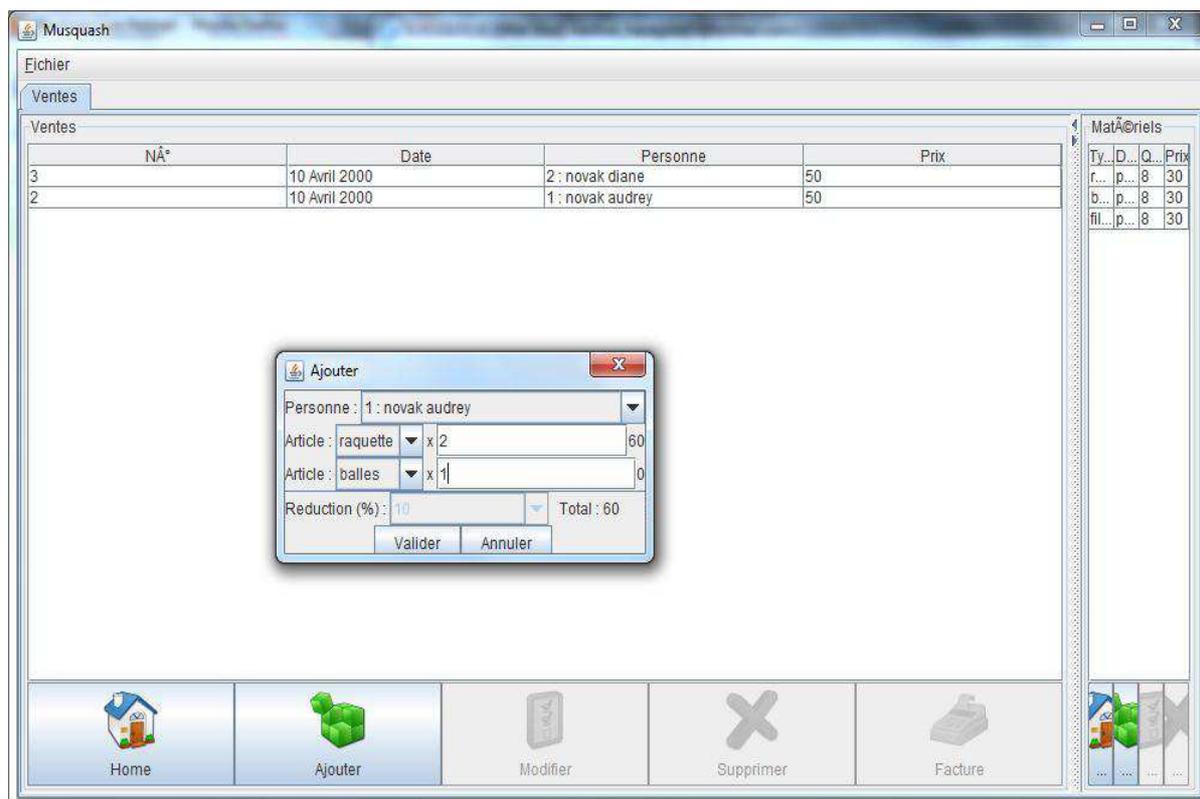


FIG. 2.6 – Gestion des ventes/locations

Ici, l'historique des matériels vendu est affiché ainsi que la liste des matériels en stock (petite fenêtre de droite, s'agrandit en cliquant dessus). L'utilisateur peut ajouter une nouvelle vente (ou location) et il peut gérer les stocks du complexe.

Client Internet :

Le client internet est un portail web permettant à n'importe quel adhérent d'accéder à de nombreuses informations. Il ne s'agit que de consultation.

Bienvenue sur Musquash
Pour entretenir le corps et l'esprit

Accueil Cours/Tarifs Planning Contact Aide

Présentation

Vous êtes sur le portail destiné à obtenir des informations concernant les cours, les horaires et les salles.
Chers membres, nous sommes ouverts à toutes les suggestions concernant ce portail.

[Votre avis](#)

Planning

Vous pouvez consulter le planning des cours, des salles et des entraînements.

Vous êtes adhérent !
Vous pouvez connaître les cours où vous êtes inscrit.



Une personne, quelle qu'elle soit, peut accéder à différentes informations. Il s'agit des forfaits proposés ainsi que les prix de location de terrains et de casiers.

Forfaits / Tarifs	Mois	Trimestre	Semestre	Année
Aucun forfait				

N'oubliez pas! Les prix sont dégressifs en fonction du nombre de forfait choisi.
Demander plus de renseignements directement en salle.

■ Locations

<u>Terrains de Squash</u>		
	Non Adhérent	Adhérent
A la Séances	10 €	5 €
Avec Carte (10 séances)	90 €	40 €

<u>Casiers</u>			
	Prix	Caution	Montant Total
A la Séances	2 €	5 €	7 €
A l'Année	20 €	50 €	70 €

En effet, celui-ci peut visualiser le planning général, c'est-à-dire les terrains de squash réservés, les créneaux des cours de squash individuels et collectifs, ainsi que les créneaux de gymnastique.

Planning

<input type="checkbox"/>	Vous êtes adhérent ! Visualiser vos cours.	N°Adhérent : <input type="text"/>	<input type="button" value="OK"/>
<input type="checkbox"/>	Visualiser le planning général par...	Tout <input type="button" value="OK"/>	

■ Squash

Réservations des Terrains

Date	Horaire	Terrain n°	Loueur	Tarif
03/04/10	14 h à 15 h	2	novak audrey	30 €
03/04/10	14 h à 15 h	5	maneschi romain	30 €
03/04/10	14 h à 15 h	6	smith john	30 €
05/04/10	14 h à 15 h	2	smith john	30 €

Créneaux de Cours Individuels

Date	Horaire	Terrain n°
05/01/10	9 h à 10 h	1
12/01/10	9 h à 10 h	2
12/01/10	10 h à 11 h	3
12/01/10	11 h à 12 h	1

Créneaux de Cours Collectifs

Trimestre	Date	Horaire	Terrain n°	Nb Place Dispo
Du 01/10/10 au 01/04/10	Tous les Mardi	10 h à 12 h	2	0 pl.

■ Gymnastique

Tous les Créneaux

Date	Type	Horaire	Salle	Capacité Max	Professeur
14/01/10	cardio	13 h à 14 h	2	30 pers.	maillet laurent
14/01/10	cardio	14 h à 15 h	3	30 pers.	maillet laurent
04/02/10	cardio	14 h à 15 h	2	30 pers.	maillet laurent
04/02/10	step	14 h à 15 h	1	30 pers.	novak audrey

De même, grâce à la recherche par numéro d'adhérent, un adhérent peut savoir quels sont les terrains de squash et les différents cours de squash individuels et collectifs qu'il a réservé, ainsi que les matériels et casiers qu'il a loué.

■ Squash

Vos Réservations

Terrain n°	Date	Horaire	Tarif
2	03/04/10	14h20 à 15h	30 €

Cours Collectifs

Terrain n°	Jour	Horaire	Trimestre
2	Tous les Mardi	10 h à 12 h	du 01/10/10 au 01/04/10

Cours Individuels

Terrain n°	Date	Horaire
1	05/01/10	9 h à 10 h
1	12/01/10	11 h à 12 h
3	12/01/10	10 h à 11 h

■ Locations

Matériels

Date	Horaire	Somme Payée	Materiel	
			Type	Prix
01/12/10	14 h à 15 h	5 €	raquette	5 €
			raquette	5 €
01/12/10	15 h à 16 h	5 €	raquette	5 €

Casiers à l'Année

Casier n°	Année
1	2009
1	2010

Casiers à la Séance

Casier n°	Date	Horaire	Prix payé
71	03/03/09	14 h à 15 h	30 €

A partir de ce portail de consultation, une personne, adhérent ou autre, peut envoyer des messages pour donner un avis ou alors poser des questions auxquels une personne du personnel répondra ultérieurement.

The image shows a web form titled "Contact" with two main sections: "Par Courrier" and "Par Email".

Par Courrier

Adresse	Place Eugène Bataillon
Code Postal	34095
Ville	Montpellier
Téléphone	04 67 14 30 28

Par Email

Email	<input type="text"/>
Objet	<input type="text"/>
Message	<input type="text"/>
<input type="button" value="Envoyer"/>	

Nous gérons également la documentation du logiciel pour le personnel du club.

Nous avons utilisé les langages de programmations CSS pour le style, HTML pour la mise en forme et PHP pour la connexion à la base oracle et les différentes requêtes.

Options :

Permet de choisir les options générales, comme choisir un thème d'apparence, changer l'adresse de la base de données ou encore modifier la langue du logiciel.

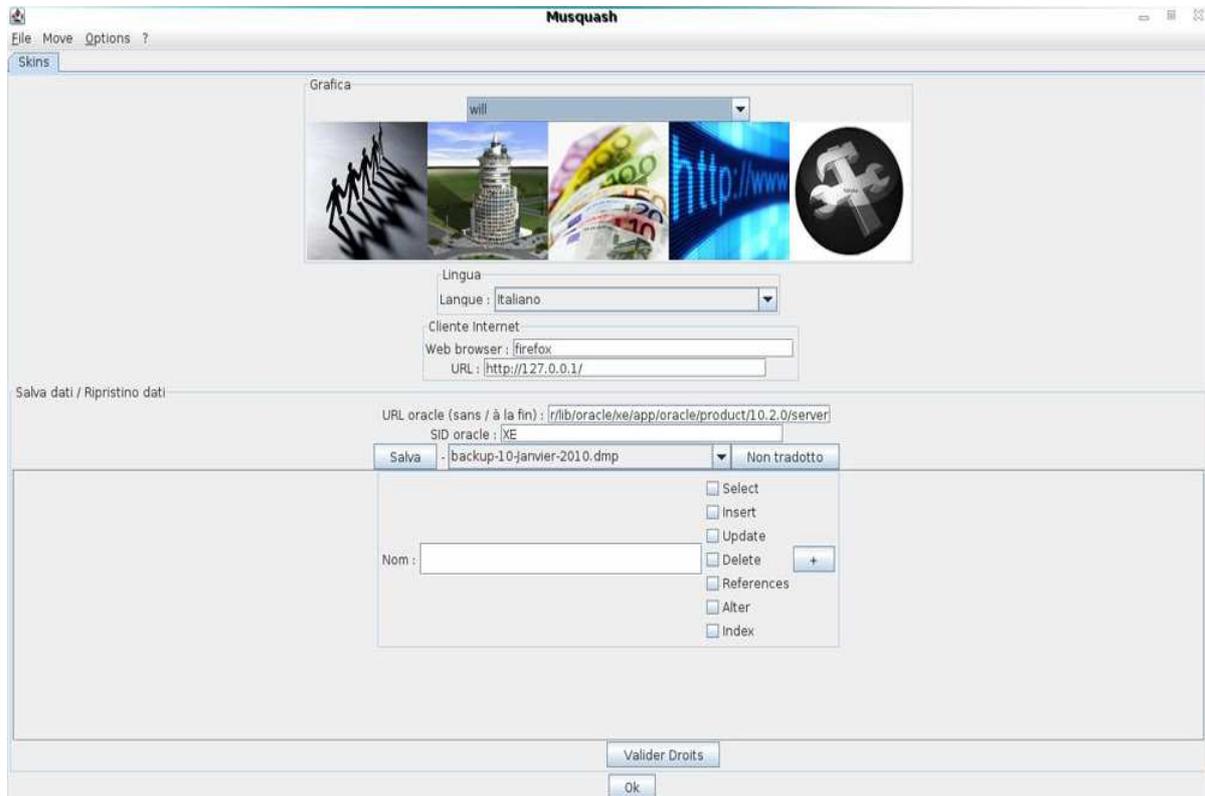


FIG. 2.7 – Gestion des casiers

Enfin, le logiciel est traduit en 4 langues différentes, au choix de l'utilisateur : Français, Anglais, Italien et Albanais.

2.3.3 Phase de test

Le projet fut testé selon des grilles de test afin de vérifier toutes les contraintes importantes. Les tables ont été recoupées deux par deux, une possédant des données toujours fixée et l'autre variable. "ok" signifie que le trigger a fonctionné (donc que l'opération est annulée)

Voici une de ces grilles :

reservationTerrainSquash (fixé a 14h20-15h00) × reservationTerrainSquash :

Test	Exemple	Adhérent variable	Terrain variable
heure_debut identique et minute_debut identique	14h20-15h00	ok	ok
heure_debut identique, heure_fin identique et minute_debut2 minute_debut1	14h40-15h20	ok	ok
heure_fin2 = heure_debut1 et minute_debut2 plus petit que minute_debut1	14h00-14h40	ok	ok
heure_fin2 = heure_debut1 et minute_debut2 minute_debut1	13h50-14h30	ok	ok
heure_fin2 = heure_debut1 et minute_fin2 = minute_debut1	13h40-14h20	ok	ok
heure_debut2 = heure_fin1 et minute_debut2 = minute_fin1	15h00-15h40	ok	ok

coursSquashIndividuel (csi) fixé a 14h00-15h00 × reservationTerrainSquash (rts) :

Test	Exemple	Adhérent variable	Terrain variable
heure_debut identique	14h00-14h40	ok	ok
heure_fin identique	14h30-15h10	ok	ok
heure_debut rts plus petit que heure_debut csi et heure_debut csi = heure_fin rts1	13h40-14h20	ok	ok

coursSquashCollectif (csc) fixé a 14h00-16h00 × reservationTerrainSquash (rts) :

Test	Exemple	Adhérent variable	Terrain variable
heure_fin rts = heure_debut csc et minute_debut rts ≠ 0	13h40-14h20	ok	ok
heure_debut identique	14h00-14h40	ok	ok
heure_debut rts = heure_debut csc + 1	15h00-15h40	ok	ok
heure_debut csc = heure_fin rts	13h20-14h00	ok	ok

reservationTerrainSquash (rts) fixé a 14h30-15h10 × coursSquashIndividuel (csi) :

Test	Exemple	Adhérent variable	Terrain variable
heure_debut identique et heure_fin identique	14h00-15h00	ok	ok
heure_debut csi = heure_fin rts	15h00-16h00	ok	ok

reservationTerrainSquash (rts) fixé a 14h30-15h10 × coursSquashCollectif (csc) :

Test	Exemple	Adhérent variable	Terrain variable
heure_debut identique	14h00-16h00	ok	ok
heure_debut rts = heure_debut csi + 1	15h00-16h00	ok	ok
heure_fin csi = heure_fin rts	13h00-15h00	ok	ok

coursSquashIndividuel (csi1) fixé a 15h00-16h00 × coursSquashIndividuel (csi2) :

Test	Exemple	Terrain fixé	Adhérent fixé	Professeur fixé
heure et jour identique, cours ≠		ok		
heure identique	15h00-16h00	ok	ok	ok

coursSquashCollectif (csc) fixé a 13h00-15h00 × coursSquashIndividuel (csi) :

Test	Exemple	Terrain fixé	Adhérent fixé	Professeur fixé
heure_debut identique	13h00-14h00	ok	ok	ok
heure_debut csi = heure_debut csc + 1	14h00-15h00	ok	ok	ok
heure_debut csi = heure_debut csc - 1	12h00-13h00	ok	ok	ok

coursSquashCollectif (csc1) fixé a 15h00-17h00 × coursSquashCollectif (csc2) :

Test	Exemple	Terrain fixé	Adhérent fixé	Professeur fixé
heure identique	15h00-17h00	ok	ok	ok
heure_debut csc2 = heure_debut csc1 + 1	14h00-15h00	ok	ok	ok
heure_debut csc1 plus petit que heure_fin csc2 plus petit que heure_fin csc1	12h00-13h00	ok	ok	ok

coursSquashCollectif (csi) fixé a 15h00-16h00 × coursSquashCollectif (csc) :

Test	Exemple	Terrain fixé	Adhérent fixé	Professeur fixé
heure_debut identique	15h00-17h00	ok	ok	ok
heure_debut csc = heure_fin csi	16h00-18h00	ok	ok	ok
heure_fin csc = heure_debut csi	13h00-15h00	ok	ok	ok
heure_fin identique	14h00-16h00	ok	ok	ok

Conclusion

Bilan

L'analyse du sujet nous a permis de bien cerner le travail et les difficultés présentes. Des diagrammes ont été fait afin de représenter la structure générale du logiciel et les contraintes associées. La création du logiciel est une réussite, les contraintes ont toutes été respectées et le programme fonctionne normalement. Son interface simple et personnalisable reste claire et permet aisément de procéder aux ajouts, suppressions, etc... L'exécutable peut facilement être déployé sur n'importe quelle machine possédant Oracle et le responsable du complexe peut immédiatement commencer à gérer ses plannings, terrains, stocks et cours.

Perspectives d'évolutions

Ce logiciel est déjà utilisable tel quel, mais plusieurs évolutions sont envisageables. Une version réservée aux adhérents pourrait être déployée sur des terminaux pour que les clients s'inscrivent d'eux même puis payent à la caisse. Une version plus générale pourrait être développée pour permettre à d'autres complexes sportifs de ce type d'utiliser le logiciel. Il faudrait par la suite créer les nouvelles contraintes.