

WIKI

Compte Rendu Projet ACSI

TocToac

Élèves ayant participé au projet :

Rouland Quentin
Sentost Geoffrey
Albert Jérémy
Aviles Adrien

Sommaire

Introduction

1) Organisation du travail

- a) Diagramme des Use Case
- b) Architecture MVC
- c) Diagramme des Classes métier
- d) SNI
- e) Schéma Relationnel du Package Métier
- f) Diagramme des Classes contrôleur
- g) Diagramme de Séquence Système
- h) Diagramme de Séquence et Diagramme des Classes Participantes

2) Conception, diagrammes...

3) Affichage de la fenêtre

- a) Maquettes d'écrans

4) Code

- a) Script de création de la base de données

Introduction

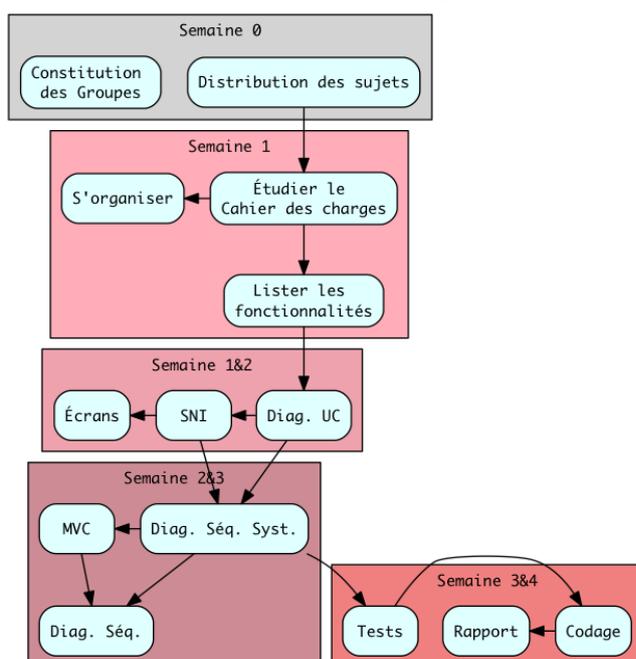
Le but du Projet ACSI 2013 est une étude de cas prévue sur 4 semaines avec comme but de réaliser un Système TocToac pour gérer des inscriptions à des entraînements pour le club Toac triathlon. L'application est donc exécutable en java et permet à 3 types d'utilisateur, un utilisateur lambda, un entraîneur et un membre du bureau de se connecter à l'application et de voir grâce à une base de données le planning des entraînements. Ils pourront aussi gérer leur compte. Nous avons donc suivis une structure MVC pour la réalisation des schémas UML puis pour le codage de l'application dont les fichiers sont disponibles sur notre espace readmine.

1) Organisation du travail

Pour ce projet nous étions un groupe de quatre personnes, composé de : Albert Jeremy, Aviles Adrien, Rouland Quentin, et Sentost Geoffrey.

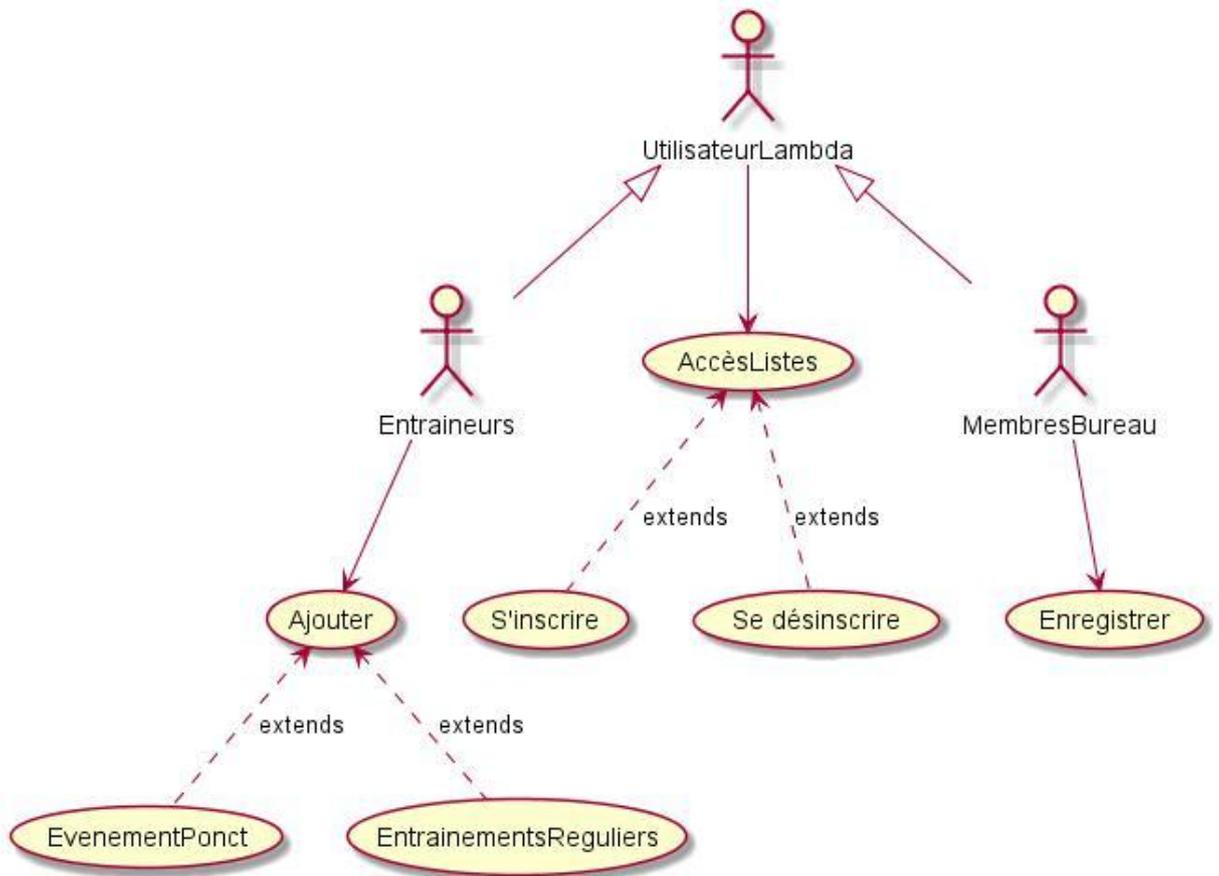
Pour plus de rentabilités de travail, nous avons créé deux sous-groupes de deux personnes. Le premier sous-groupe, composé d'Adrien et de Jeremy, s'est occupé du SNI, du DSS, du DS, et du DCP. Le second sous-groupe, composé de Geoffrey et Quentin, s'est occupé du Diagramme Use Case, des Maquettes écran, du Diagramme de Classes, et du Modèle Vue Contrôleur. Pour la réalisation du code, Geoffrey, Jeremy et Adrien se sont découpés les différentes classes des packages vue et contrôleur. Alors que Quentin a réalisé le package modèle avec la création de la base de données et différentes méthodes pour accéder à celle-ci.

Le temps était réparti de cette façon :

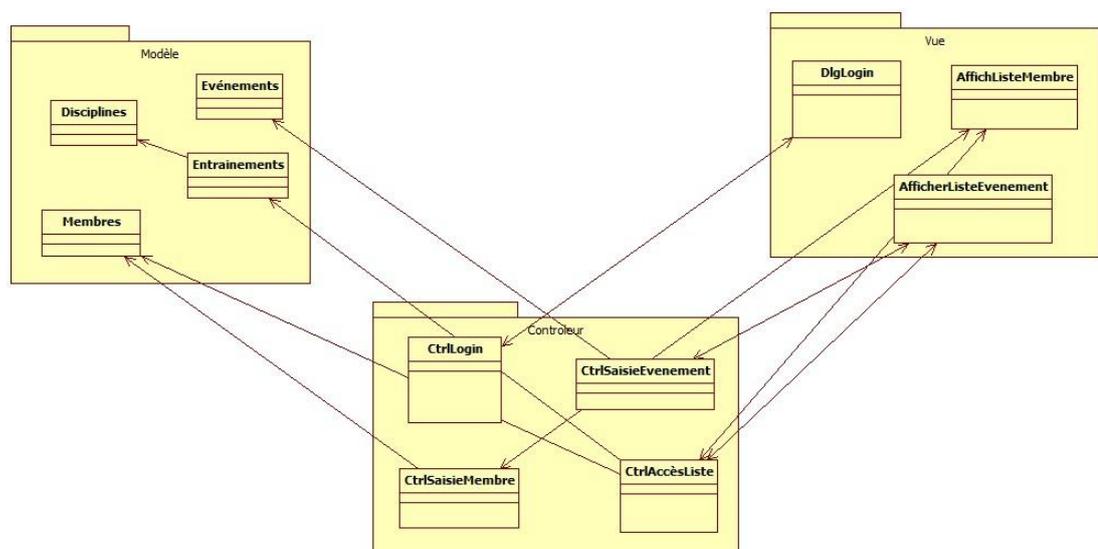


2) Conception, diagrammes...

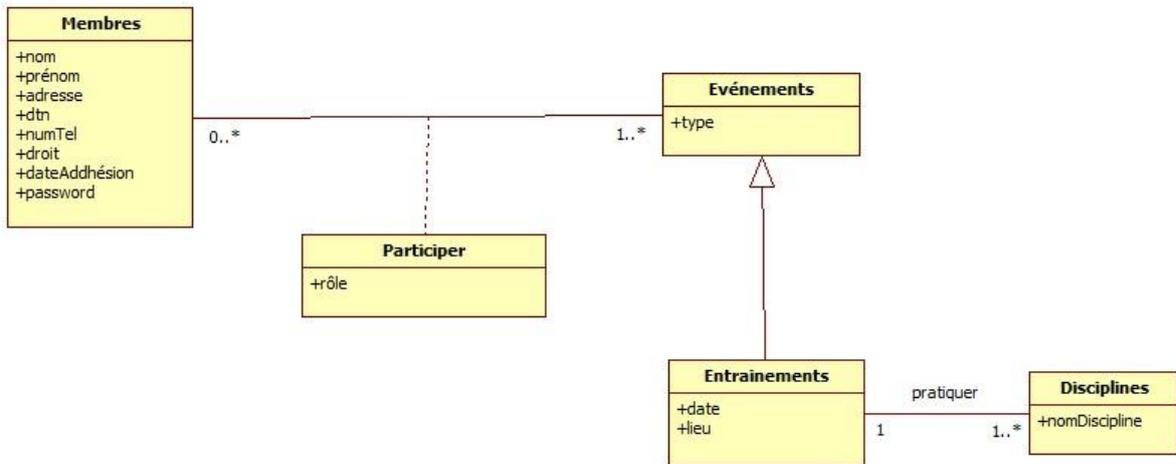
a) Diagramme des Use Case



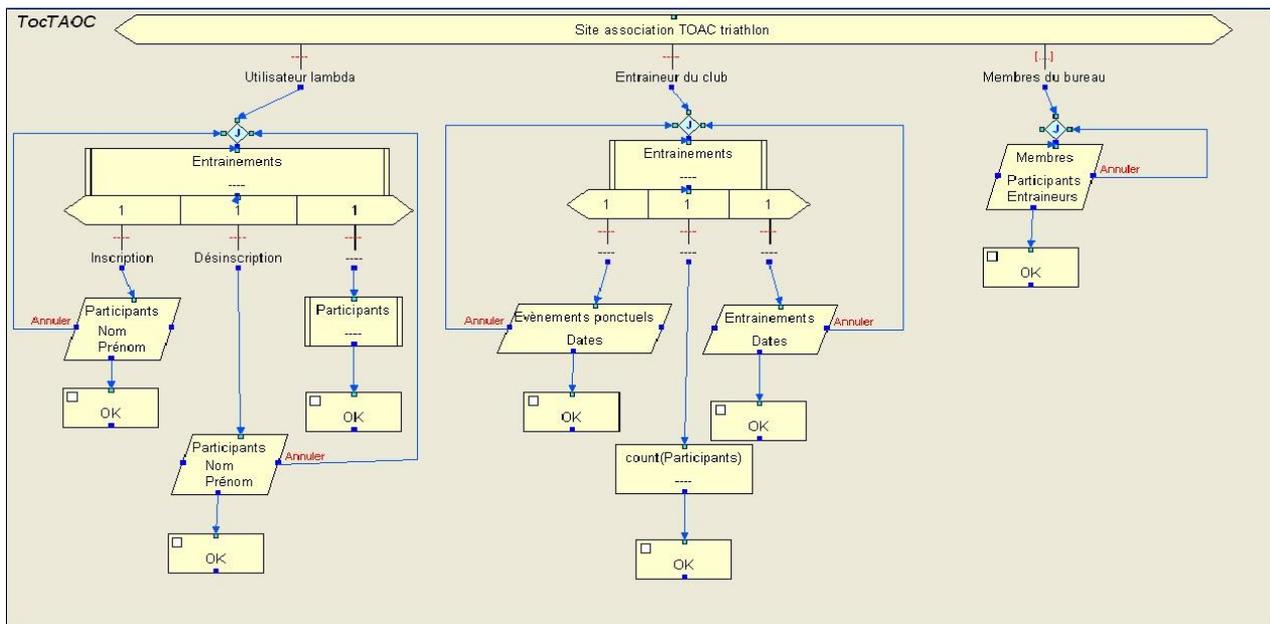
b) Architecture MVC



c) Diagramme des Classes métier



d) SNI



e) Schéma Relationnel du Package Métier

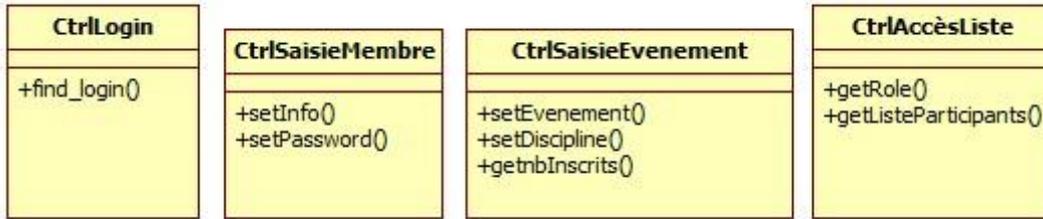
membres(idMembre, nom, prénom, adresse, dtn, numTel, droit, dateAdhesion, password);

evenements(idEvenement, type, dateE, lieu);

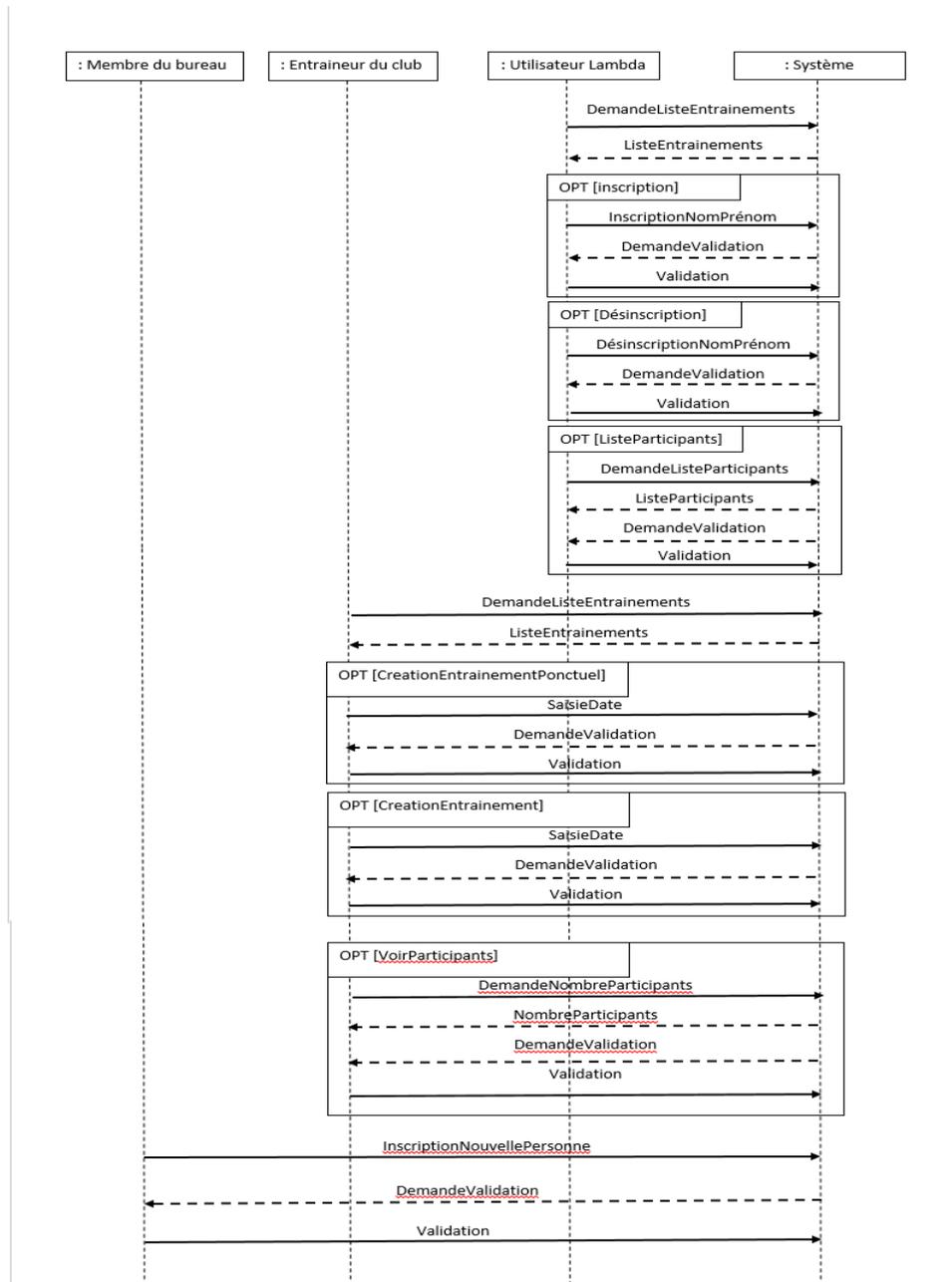
participer(#idEvenement, #idMembre, role);

disciplines(nomDiscipline,#idEvenement)

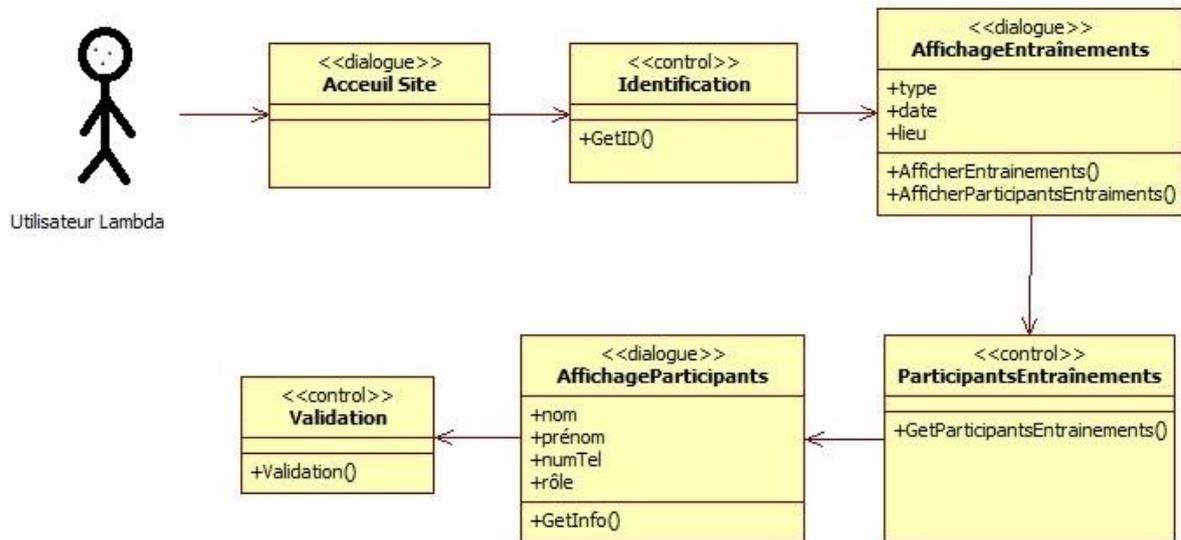
f) Diagramme des Classes contrôleur



g) Diagramme de Séquence Système

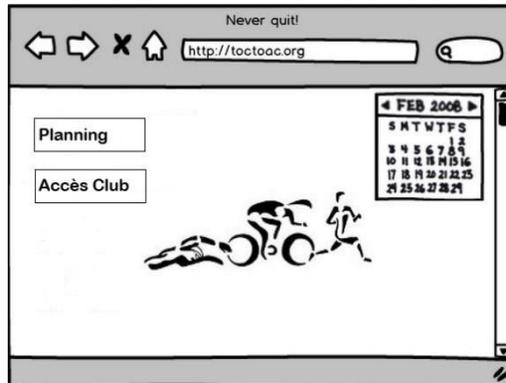


h) Diagramme de Séquence et Diagramme des Classes Participantes



4 - Affichage de la fenêtre

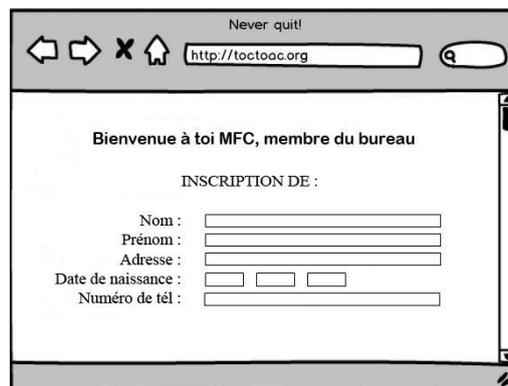
a) Maquettes d'écrans



Page d'accueil du site



Page de connexion



Page d'inscription de membres pour les membres du bureau

4 – Code

a) Script de création de la base de données

```
create table membres
```

```
(  
    idMembre IDENTITY,  
    nomvarchar(20),  
    prenomvarchar(20),  
    adressevarchar(50),  
    dtn date,  
    numTelvarchar(20),  
    droitint,  
    dateAdhesion date,  
    passwordvarchar(20),  
);
```

```
create table evenements
```

```
(  
    idEvenement IDENTITY,  
    typeint,  
    datedateE,  
    lieuvarchar(20),  
);
```

```
create table participer
```

```
(  
    idEvenementint,  
    idMembreint,  
    roleint,  
  
    FOREIGN KEY (idMembre) REFERENCES membres(idMembre),  
    FOREIGN KEY (idEvenement) REFERENCES evenements(idEvenement),  
    PRIMARY KEY (idMembre,idEvenement)  
);
```

```
create table disciplines
```

```
(  
    nomDisciplinevarchar(20),  
    idEvenementint,  
    FOREIGN KEY (idEvenement) REFERENCES evenements(idEvenement),  
    PRIMARY KEY (nomDiscipline)  
);
```

La voie que nous avons retenue, mais que nous n'avons pas pu terminer dans la partie codage était la suivante :

Au lancement de l'application, une fenêtre de login apparaît, elle permet de nous identifier peu importe notre "statut" dans le club, il faut juste être inscrit au préalable. Si nous ne sommes pas inscrits, ce sont les membres du bureau qui gèrent notre inscription à travers un formulaire basique.

Une fois identifié grâce à notre login et notre mot de passe, nous avons accès à notre planning.

1 - Etant un utilisateur lambda, je peux voir mes entraînements dans un calendrier.

Je peux m'inscrire à un nouveau créneau s'il est établi par mon entraîneur, sinon je peux me désinscrire d'un créneau où je me suis déjà inscrit. De plus, en cliquant sur une journée, j'ai le détail horaire de mes entraînements, ainsi que la discipline pratiquée. Tous les entraînements passés, actuels et futurs sont enregistrés dans notre base de données et accessibles via le planning grâce à un code couleur précisant si le créneau est disponible pour une inscription, ou si nous sommes déjà inscrits sur le créneau.

2 - Etant un entraîneur, je peux faire tout ce qu'à la possibilité de faire un utilisateur lambda, avec comme plus, de rajouter un créneau d'entraînement, ou un événement ponctuel au planning.

3 - Etant un membre du bureau, j'ai accès à toutes les informations relatives au club, et je peux inscrire et désinscrire les membres du club. Je peux faire de même avec les entraîneurs.