Hashcode

**Groupe No :** 6

Rapport de création

Butty Joé

Fuchs Nicolas

Rial Jonathan

**Filière :** Informatique

**Technologies :** Java, MySQL

**Date du rendu :** 23.03.2018

**Superviseurs :** Prof. Houda Chabbi Drissi

Prof. Pierre Kuonen

Prof. Omar Abou Khaled

**Client:** Prof. Pierre Kuonen

**Table des matières**

1 Diagramme de cas d’utilisation 3

2 Description des acteurs 4

3 Fiches Descriptives 7

3.1 NomDuCas 1 7

3.1.1 Description des enchaînements : 7

3.1.2 Besoin d’IHM: 7

3.2 NomDuCas 2 9

4 Modèle Entité-Association (DBMain) 9

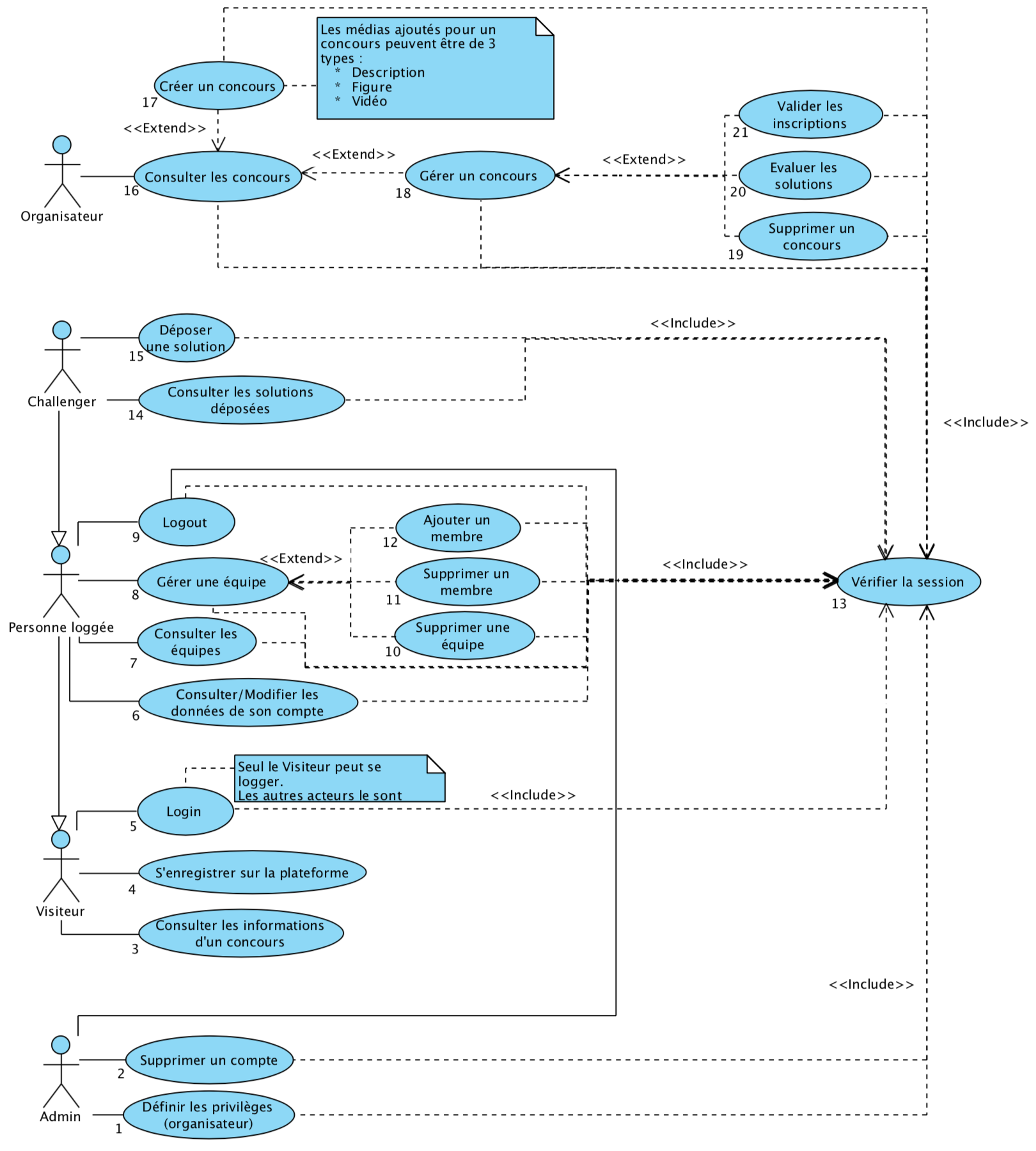
5 Modèle Relationnel normalisé 9

6 XML schéma 9

7 Signatures 9

# Diagramme de cas d’utilisation

La plateforme a été créée dans le but de pouvoir organiser des concours de programmation. Ce genre de concours consiste à écrire un programme qui permet de résoudre un problème donné. Les différentes solutions sont évaluées et classées.



Le visiteur de la plateforme peut consulter les concours actuels et futurs. Il peut également s’enregistrer sur la plateforme en tant qu’utilisateur intéressé à participer à un concours ou en tant qu’organisateur. Dans le cas où le visiteur souhaite s’enregistrer en tant qu’organisateur, l’administrateur doit approuver la création de son compte. Si l’utilisateur s’est déjà enregistré sur la plateforme, il peut se loguer afin d’accéder à son espace personnel.

L’administrateur peut approuver la création de comptes des organisateurs. Il peut également supprimer n’importe quel compte qui a été créé sur la plateforme. Les comptes administrateur ne sont pas créés avec la plateforme.

La personne loguée peut consulter et modifier les données de son compte. Il peut également consulter son équipe pour un concours donné. Il peut également créer une équipe dans laquelle il peut ajouter et retirer des membres. Il a également la possibilité de supprimer une équipe qu’il aurait créé. Il peut également se déconnecter de la plateforme.

Le challenger peut déposer une solution au nom de son équipe pour un concours donné. Il peut également consulter toutes les anciennes versions que son équipe a soumises.

Un organisateur peut créer un concours. Lors de la création de celui-ci, il a la possibilité d’ajouter une description, des figures ou des vidéos. Il peut également consulter tous les concours qu’il a créé. Il peut valider les inscriptions à ses concours, évaluer les solutions déposées par les groupes et supprimer les concours qu’il a créé.

Durant la navigation et les manipulations des utilisateurs, la plateforme vérifiera à de nombreuses reprises la session de ceux-ci. Elle déconnectera automatiquement les utilisateurs qui n’ont plus interagit avec le système dans un délais spécifié. Ceci a pour but de renforcer la sécurité de la plateforme.

# Description des acteurs

**Visiteur :**

Le visiteur, comme son nom l’indique, est une personne qui n’est pas connecté à la plateforme. Il peut consulter les informations d’un concours.

**Vue externe**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tables* | *Select* | *Insert* | *Update* | *Delete* |
| account |  | x |  |  |
| role |  |  |  |  |
| team | x |  |  |  |
| challenge | x |  |  |  |
| solution |  |  |  |  |
| data | x |  |  |  |
| team | x |  |  |  |
| challenge-organizer | x |  |  |  |
| challenger-team | x |  |  |  |

**Personne loguée :**

Une personne loguée est un utilisateur ayant un compte sur la plateforme permettant de s’inscrire à des concours. Pour chaque concours, il peut consulter les informations, créer et modifier une équipe afin d’y participer.

**Vue externe**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tables* | *Select* | *Insert* | *Update* | *Delete* |
| account | x |  | x\* |  |
| role |  |  |  |  |
| team | x | x\* | x\* | x\* |
| challenge | x |  |  |  |
| solution | x\* |  |  |  |
| data | x |  |  |  |
| challenge-organizer | x |  |  |  |
| challenger-team | x | x | x | x |

**Challenger :**

Un challenger est un utilisateur qui participe au concours actuel. Il peut déposer des solutions et les consulter pour le challenge en cours.

**Vue externe**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tables* | *Select* | *Insert* | *Update* | *Delete* |
| account | x |  | x\* |  |
| role |  |  |  |  |
| team | x | x\* | x\* | x\* |
| challenge | x |  |  |  |
| solution | x | x |  |  |
| data | x |  |  |  |
| challenge-organizer | x |  |  |  |
| challenger-team | x | x | x | x |

**Organisateur :**

Un organisateur est une personne qui peut gérer les différents concours sur la plateforme.

**Vue externe**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tables* | *Select* | *Insert* | *Update* | *Delete* |
| account | x |  | x\* |  |
| role |  |  |  |  |
| team | x |  |  | x\* |
| challenge | x | x | x | x |
| solution | x |  |  |  |
| data | x | x | x | x |
| challenge-organizer | x | x\* | x\* | x\* |
| challenger-team | x |  |  |  |

**Admin :**

Un administrateur est une personne qui définit les privilèges des comptes. En plus de pouvoir consulter les informations d’un concours, il gère les organisateurs.

**Vue externe**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Tables* | *Select* | *Insert* | *Update* | *Delete* |
| account | x | x | x | x |
| role | x |  |  |  |
| team |  |  |  |  |
| challenge |  |  |  |  |
| solution |  |  |  |  |
| data |  |  |  |  |
| challenge-organizer |  |  |  |  |
| challenger-team |  |  |  |  |

# Fiches Descriptives

**Remarque** : Chaque fiche descriptive commence au haut d’une nouvelle page

## NomDuCas 1

**Description résumée** : ….

**Acteurs** : ..

### Description des enchaînements :

**Pré conditions** : ….

**Scénario nominal**

1. …
2. …
3. Fin du cas

**Enchaînements alternatifs**

A1 : …

Démarre au(x) point (s) x du scénario nominal

1. ….
2. ….
3. Reprend au point xx du scénario nominal

Enchaînements d’exception

E1 : ….

Démarre au(x) point(s) xx du scénario nominal

1. ….
2. ….
3. Le cas est terminé

**Post conditions : …**

### Besoin d’IHM:

Cette section doit contenir un aperçu commenté en détail des interfaces nécessaires à l’exécution de cas d’utilisation. Indiquez les technologies qui seront utilisées.

|  |  |
| --- | --- |
| IHM | Relation use case – interface |
|  | * Nom du cas d’utilisation  : * Acteurs concernées : * Identifier les parties de l’interface concernée par ce cas d’utilisation : |

L'aperçu de l’interface peut être réalisé soit à la main (scanné) soit avec un outil de dessin soit avec une copie d’écran de la vraie interface.

**Remarques (optionnel) :**

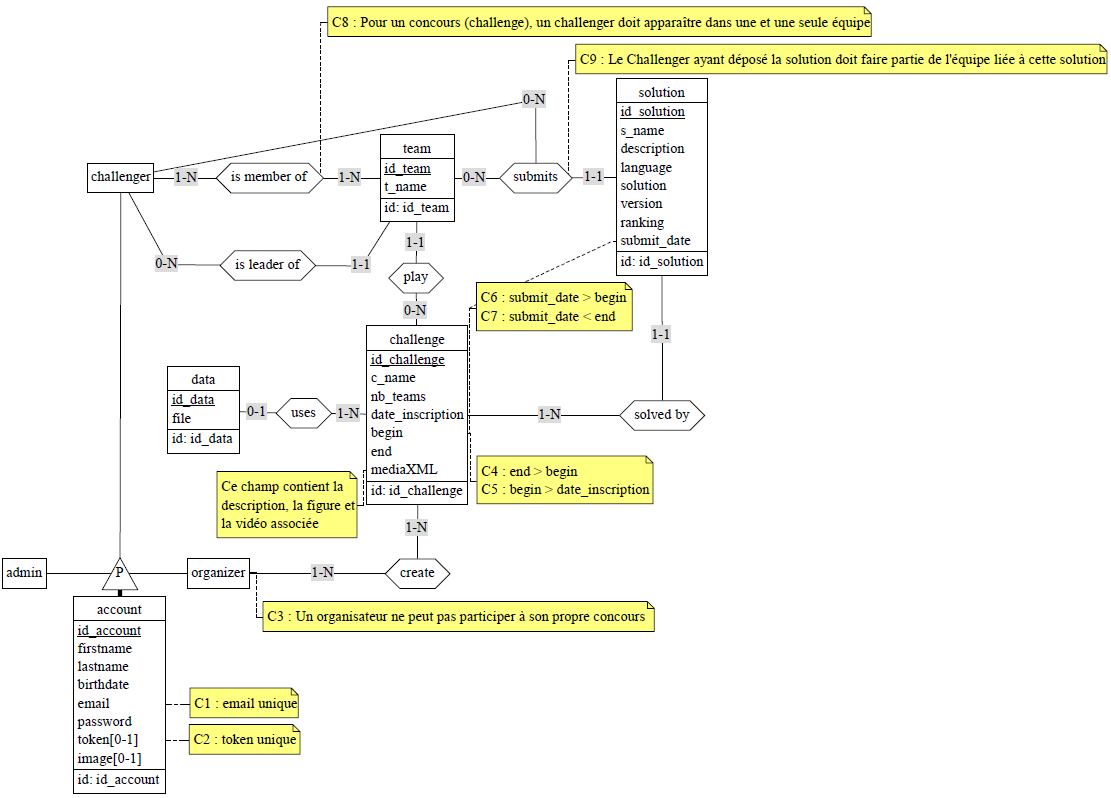
…..

## NomDuCas 2

…..

# Modèle Entité-Association

Voici ci-dessous le modèle entité-association de la base de données du projet :



**C1** : Afin de pouvoir identifier le compte de chaque utilisateur, les adresses email doit être unique dans la base de données.

**C2** : Afin de pouvoir identifier chaque utilisateur grâce à leur token, ceux-ci seront unique dans la base de données.

**C3** : Afin de pouvoir garder les concours équitables, les organisateurs ne pourront pas participer à leurs propres concours.

**C4** : Afin de pouvoir préserver une chronologie logique, la date de fin du concours devra être plus ancienne que la date de début.

**C5** : Afin de pouvoir préserver une chronologie logique, la date du début du concours devra être plus ancienne que la date de la fin des inscriptions.

**C6** : Afin de pouvoir garantir qu’une solution soit déposée dans les délais, la date de la soumission d’une solution devra être plus ancienne que la date de début du concours.

**C7** : Afin de pouvoir garantir qu’une solution soit déposée dans les délais, la date de la soumission d’une solution devra être plus récente que la date de fin du concours.

**C8** : Afin de pouvoir prévenir que les challengers puissent uniquement participer à un concours donné avec une et une seule équipe, une vérification sera effectuée.

**C9** : Afin de pouvoir garantir qu’une solution soit déposée par une personne qui fasse partie de l’équipe à laquelle cette solution est liée, une vérification sera effectuée.

# Modèle Relationnel normalisé

4 Modèle Entité-Association

* Modèle relationnel relatif à votre modèle E/A. Pas de snapshot de ce que donne DBMain ou autre!
* Listes des Contraintes d’Intégrités (Les Ci du EA + les nouvelles de nouveau nommées : Cr1, Cr2 etc.)
* Étude de la normalisation de chaque table :
  + Liste des DFs relative à tout votre schéma relationnel
  + Justificatifs de la forme normale par relation

# XML schéma

Il s’agit de fournir :

1. Un modèle XML schéma documenté (avec commentaires) des documents XML qui seront stockés dans la base de données
2. Une instance XML valide par rapport au modèle proposé
3. Les justifications de l’utilisation de XML pour votre projet dans la base de données

# Signatures

Fribourg, le ……………..............

Etudiant1 Etudiant2 Etudiant3