

## REMERCIEMENTS

Au cours de mon apprentissage j'ai eu la chance de rencontrer de nombreuses personnes bienveillantes. Voici l'occasion d'en remercier quelques-unes.

Premièrement je souhaite remercier mes parents ainsi qu'Isabelle Chrispeels, Bertrand Charlier, Georges Fretin et Chantal Roisin qui m'ont soutenu et encouragé dans mon combat contre mes problèmes de dysgraphie, de dyslexie et de dysorthographe et amené sur le chemin de la réussite. Sans eux je n'en serais pas là où j'en suis.

Merci à Barnabé Deliens et Ruben Hillewaere. C'est grâce à vous que, lors de mon passage à l'ECAM, j'ai trouvé ma voie.

Merci à Thierry Morre, Michaël Person et François Koeune pour l'aide précieuse que vous m'avez apportée tout au long de mon apprentissage du développement informatique.

Et enfin un merci tout particulier à Antoine Corbisier. C'est en grande partie, grâce à votre disponibilité, votre bonne volonté et votre gentillesse que ces 2 dernières années si particulières se sont passées au mieux.

# TABLE DES MATIERES

INTRODUCTION :	- 1 -
AVANT-PROPOS.....	- 1 -
MOTIVATION .....	- 1 -
PROJET .....	- 2 -
<i>La partie site web</i> .....	- 2 -
LOGICIELS UTILISES.....	- 3 -
CAHIER DES CHARGES .....	- 4 -
UTILISATEURS DE L'APPLICATION .....	- 4 -
FONCTIONS ATTENDUES .....	- 4 -
Visiteurs :	- 4 -
utilisateurs :	- 4 -
Organisateurs :	- 4 -
PÉRIMÈTRE DU PROJET .....	- 5 -
<i>Dans le scoop</i> .....	- 5 -
Tournoi .....	- 5 -
<i>Hors du scoop</i> .....	- 5 -
Deck :	- 5 -
Tournoi :	- 5 -
ANALYSE FONCTIONNELLE .....	- 6 -
ENTITÉ-ASSOCIATION .....	- 6 -
<i>Schéma</i> .....	- 6 -
<i>Entités</i> .....	- 7 -
Utilisateur.....	- 7 -
Jeu .....	- 7 -
Tournoi.....	- 7 -
Dotation.....	- 7 -
Résultat.....	- 8 -
Deck.....	- 8 -
Round .....	- 8 -
Match .....	- 8 -
Partie.....	- 9 -
<i>Associations</i> .....	- 9 -
Organisateur (tournoi - utilisateur) .....	- 9 -
Joueur (tournoi - utilisateur).....	- 9 -
PseudoIG (Utilisateur – Jeu) .....	- 9 -
a pour (Tournoi – Dotation).....	- 9 -
du (Tournoi – Jeu) .....	- 9 -
est classé (Utilisateur – Résultat).....	- 9 -
au (Résultat – Tournoi) .....	- 10 -
DeckJoueur (utilisateur – Deck – Tournoi) .....	- 10 -
Appartient (Deck – Jeu) .....	- 10 -
à (Tournoi – Round) .....	- 10 -
Est Composé (Round – Match) .....	- 10 -
Participe comme joueur1 (Utilisateur – Match) .....	- 10 -
Participe comme joueur2 (Utilisateur – Match) .....	- 10 -
Manche (Match – partie).....	- 10 -
Est joué comme deck1 (Deck – Partie) .....	- 10 -
Est joué comme deck2 (Deck – Partie) .....	- 10 -
(Utilisateur – Partie).....	- 10 -
(Utilisateur – Partie).....	- 10 -

MODÈLE RELATIONNEL .....	- 11 -
<i>Diagramme</i> .....	- 11 -
<i>Tables</i> .....	- 12 -
Utilisateur.....	- 12 -
Jeu.....	- 12 -
Tournoi.....	- 12 -
PseudoIG.....	- 13 -
Dotation.....	- 13 -
Deck.....	- 13 -
Résultat.....	- 13 -
Organisateur .....	- 14 -
Joueur.....	- 14 -
DeckJoueur.....	- 14 -
Round.....	- 14 -
Match .....	- 15 -
Partie.....	- 15 -
<i>Règles de validation</i> .....	- 15 -
Utilisateur.....	- 16 -
Jeu.....	- 16 -
Tournoi.....	- 16 -
PseudoIG.....	- 17 -
Dotation.....	- 17 -
Deck.....	- 17 -
Résultat.....	- 17 -
Organisateur .....	- 17 -
Joueur.....	- 17 -
DeckJoueur.....	- 18 -
Round.....	- 18 -
Match .....	- 18 -
Partie.....	- 18 -
CAS D'UTILISATION .....	- 19 -
<i>Diagramme de cas d'utilisation globale</i> .....	- 19 -
<i>Scénarios de cas d'utilisation</i> .....	- 20 -
Cas d'utilisation : Connexion.....	- 20 -
Cas d'utilisation : Administration d'un tournoi .....	- 22 -
Gérer une ronde : Commencer une ronde.....	- 23 -
Gérer une ronde : Entrer / Modifier les résultats d'un match.....	- 25 -
Gérer une ronde : Finir une ronde .....	- 27 -
ALGORITHME : <i>PAIRINGS</i> .....	- 29 -
<i>Formalisation du problème</i> .....	- 29 -
Objectif .....	- 29 -
Données nécessaires.....	- 29 -
Données à dispositions.....	- 29 -
Résultat voulu .....	- 30 -
Conditions initiales particulières ou problématiques .....	- 30 -
Erreur possible. ....	- 30 -
Input.....	- 30 -
Output .....	- 30 -
Output en cas d'erreur .....	- 30 -
<i>Explication et schématisation de l'algorithme préliminaire</i> .....	- 31 -
ARCHITECTURE DE L'APPLICATION .....	- 33 -
PROTOTYPES DE PAGES WEB.....	- 34 -
Home .....	- 34 -
Tournoi .....	- 34 -
Profil.....	- 35 -
Admin Tournoi.....	- 35 -

CONCLUSION .....	- 37 -
DÉFINITIONS ET GLOSSAIRE : .....	- 38 -
SOURCES ET BIBLIOGRAPHIE.....	- 41 -
ANNEXES .....	A
<i>Figure 1 - Entité-Association .....</i>	<i>B</i>
<i>Figure 2 - Relationnel.....</i>	<i>C</i>
<i>Figure 3 - Diagramme cas d'utilisation Globale .....</i>	<i>D</i>
<i>Figure 4 - Diagramme de séquence Connexion.....</i>	<i>E</i>
<i>Figure 5 - Diagramme d'activité Connexion.....</i>	<i>F</i>
<i>Figure 6 - Diagramme cas d'utilisation Administration d'un tournoi .....</i>	<i>G</i>
<i>Figure 7 - Diagramme d'activité Administration d'un tournoi .....</i>	<i>H</i>
<i>Figure 8 - Diagramme de séquence Commencer une ronde.....</i>	<i>I</i>
<i>Figure 9 - Diagramme d'activité Commence une ronde .....</i>	<i>J</i>
<i>Figure 10 - Diagramme de séquence Résultat match.....</i>	<i>K</i>
<i>Figure 11 - Diagramme d'activité Résultat match.....</i>	<i>L</i>
<i>Figure 12 - Diagramme de séquence Finir une ronde.....</i>	<i>M</i>
<i>Figure 13 - Diagramme d'activité Finir une ronde.....</i>	<i>N</i>
<i>Figure 14 - Algorithme de pairings .....</i>	<i>O</i>
<i>Figure 15 - Architecture du site web.....</i>	<i>P</i>
<i>Figure 16 - Mockup Home.....</i>	<i>Q</i>
<i>Figure 17 - Mockup Tournoi.....</i>	<i>R</i>
<i>Figure 18 - Mockup Profil.....</i>	<i>S</i>
<i>Figure 19 - Mockup Admin Tournoi.....</i>	<i>T</i>

# INTRODUCTION :

## AVANT-PROPOS

Ce projet est réalisé dans le cadre de l'épreuve intégrée de bachelier en informatique de gestion à l'EPHEC. Il constitue mon travail de fin d'études requis pour l'obtention de mon diplôme.

Il n'y a pas de client final pour ce projet.

Les informations sur les noms et les mots techniques propres au projet seront détaillés dans le glossaire.

L'intégralité du travail réalisé est disponible sur GitHub : <https://github.com/Aedjis/TFE>

## MOTIVATION

Pour mon TFE j'ai choisi comme sujet la gestion en ligne de tournois de jeux de cartes, et plus précisément la gestion des tournois en rondes suisses<sup>1</sup>.

Ce choix est motivé d'abord par le fait que je joue depuis l'enfance à des jeux de cartes, appelés TCG pour « *Trading Card Game* » ce qui signifie que ce sont des jeux de cartes à jouer et à collectionner.

Le premier auquel j'ai joué est Pokémon™ (TCG) dans les années 2000. J'ai de suite accroché même si, à cause de mon jeune âge, j'avais du mal à comprendre toutes les règles. J'ai ensuite découvert Magic The Gathering™ et Yu-Gi-Oh!™ (TCG), pour lesquels je me suis également passionné. Parce que, à l'époque, je comprenais très mal l'anglais, j'ai fini par abandonner Magic. Et comme à cette période il était aussi extrêmement difficile de trouver des joueurs pour jouer à ces jeux, j'ai arrêté d'y jouer pendant quelques années.

En 2012, sous l'impulsion de quelques amis, je me suis remis à jouer à Magic. J'y joue encore aujourd'hui ainsi qu'à quelques autres TCG qui sont eux, principalement, en ligne.

Être un joueur assidu m'a permis de participer à certains événements qui se déroulaient sous la forme de mini tournois, mais également de suivre quelques compétitions qui se déroulaient sous le format des rondes suisses. En suivant et participant à ces événements j'ai pu me rendre compte que la gestion de ces tournois n'était pas toujours évidente. Dès lors quand j'ai dû choisir un sujet pour mon travail de fin d'études je me suis dit qu'il serait intéressant de voir comment créer un programme pour gérer ce genre de problématique.

Qui plus est, la création d'un algorithme pour la gestion des rondes suisses était un défi que je trouvais attrayant.

---

<sup>1</sup> Type de tournoi, voir glossaire pour plus de détails.

# PROJET

Un organisateur de tournoi souhaite avoir un programme organisant automatiquement les rondes suisses de ses tournois. Il désire que l'appareillement et le calcul du classement avec le premier et le deuxième départage soient également inclus.

Ces tournois se jouent avec de 3 à 5 *decks*<sup>2</sup> de 30 cartes (ou plus) en fonction des règles du tournoi. Chaque *deck* peut contenir jusqu'à 2 fois la même carte. L'organisateur voudrait donc que le programme gère aussi les *decks* soumis par les participants (pas de vérification automatique).

L'organisateur souhaite aussi que le programme enregistre les gains et les points engrangés par les joueurs durant ces tournois.

Le cœur de la problématique est la gestion des rondes suisses ainsi que des points de départage.

---

## LA PARTIE SITE WEB

Elle devra contenir :

- La liste des tournois en cours, passés et à venir
- Pour chaque tournoi :
  - Les informations basiques du tournoi (nom, date, description, jeu, nombre de *decks*, maximum de joueurs ...)
  - La dotation du tournoi (s'il y en a une)
  - Le résultat du tournoi s'il est fini ou le classement temporaire
  - La liste des organisateurs
  - La liste des joueurs
  - La liste des rondes du tournoi
  - Pour chaque ronde :
    - La liste des matchs
- Pour chaque utilisateur :
  - Les informations basiques de l'utilisateur
  - La liste des pseudos<sup>3</sup> en jeu qui ont été renseignés
  - La liste des résultats pour les tournois finis

Les différents contenus du site seront encodés soit par les utilisateurs (pour les informations de leur profil et les résultats de leurs matchs), soit par les organisateurs (pour les informations et la gestion du tournoi) soit exceptionnellement par l'administrateur, lors de l'ajout d'un nouveau jeu, afin d'éviter les doublons.

---

<sup>2</sup> Anglicisme qui désigne un paquet de cartes à jouer, voir glossaire pour plus de détails.

<sup>3</sup> Diminutif de pseudonyme, qui est un nom d'emprunt, voir glossaire pour plus de détails.

## LOGICIELS UTILISES

- Rapport du projet : Microsoft® Word pour Microsoft 365 16.0
- Schéma entité-association : Looping 3.0
- Diagrammes des cas d'utilisation, de séquences et d'activités : Visual Paradigm 16.2
- Schéma algorithme : Lucidchart (mars 2021)
- Maquettes des pages web : MockFlow (avril 2021)
- Serveur de bases de données : MS SQL Server 2017 17.9.1
- Environnement de développement : Visual Studio 2019 Community 4.8
- Langages utilisés : C#, ASP.NET, RAZOR, HTML 5, Javascript, SQL, CSS 3
- Frameworks<sup>4</sup> : ASP.NET CORE 5 MVC, RAZOR 3.2, jQuery 3.5, Bootstrap 3.4.1
- Logiciel de gestion de versions : GitHub 2.8

---

<sup>4</sup> Ensemble d'outils et de composants logiciels, voir glossaire pour plus de détails.

# CAHIER DES CHARGES

Nom du projet : Tour0Suisse

## UTILISATEURS DE L'APPLICATION

Potentiellement tout le monde peut utiliser l'application pour obtenir des informations sur les tournois et leurs résultats. Il ne faut pas être authentifié sur le site web pour voir son contenu public.

Les modifications de contenu ne pourront se faire que par des utilisateurs authentifiés ayant les droits nécessaires. Cependant, n'importe qui peut s'inscrire comme utilisateur.

## FONCTIONS ATTENDUES

### VISITEURS :

- Voir les utilisateurs et leurs détails
- Voir les tournois et leurs détails
- S'inscrire comme utilisateur
- Voir la liste des matchs

### UTILISATEURS :

- Voir les utilisateurs et leurs détails
- Voir les tournois et leurs détails
- Voir la liste des matchs (pour connaître ses adversaires)
- Rentrer les résultats des parties pour ses matchs en cours
- S'inscrire à un tournoi (avant qu'il ne commence)
- Se désinscrire d'un tournoi (avant qu'il ne commence)
- Abandonner un tournoi en cours
- Modifier ou supprimer son compte

### ORGANISATEURS :

- Créer des tournois
- Pour chacun de leurs tournois :
  - o Modifier ou supprimer le tournoi
  - o Ajouter de nouveaux organisateurs
  - o Générer les *pairings*<sup>5</sup> d'une ronde
  - o Modifier les scores d'un match
  - o Clôturer une ronde
  - o Enregistrer l'abandon d'un joueur
  - o Clôturer le tournoi

---

<sup>5</sup> Anglicisme pour appariement, voir glossaire pour plus de détails.



# PÉRIMÈTRE DU PROJET

---

## DANS LE SCOOP

---

### TOURNOI

- Créer des rondes
- Appareillage automatique
- Reporter les résultats des matchs avec les *decks* utilisés pour chacune des parties
- Le système de départage SPA<sup>6</sup>
- Un départage manuel

---

## HORS DU SCOOP

---

### DECK :

- Bien que les listes des *decks* soient enregistrées celles-ci ne feront l'objet d'aucune vérification automatisée. Ce travail sera géré soit par les organisateurs du tournoi, soit via un logiciel tiers.
- Les *decks* sont enregistrés comme des entités uniques. Il n'est donc pas prévu de faire des statistiques dessus et la base de données ne sera pas forcément prévue pour.

---

### TOURNOI :

- Les litiges entre joueurs seront gérés par les organisateurs.
- La communication entre les différentes parties se fera avec un logiciel tiers.
- Les tournois privés ne seront pas considérés.
- L'appareillage manuel ne sera pas considéré.
- Les tournois n'utilisant pas exclusivement un système de ronde suisse ne seront pas considérés.
- La vérification que l'entièreté des résultats des matchs ait été reportée avant la clôture d'une ronde ne sera pas automatique.
- L'implémentation<sup>7</sup> de bannissement des *decks* pour un match ne sera pas supporté nativement.
- Un match ne pourra pas avoir plus de parties que le nombre de *decks* à soumettre par les 2 joueurs moins un.

---

<sup>6</sup> Système de départage additionnant tous les points des adversaires, voir glossaire pour plus de détails.

<sup>7</sup> Anglicisme passé dans l'usage, qui signifie mettre en œuvre, voir glossaire pour plus de détails.



---

## ENTITÉS

---

### UTILISATEUR

---

L'entité Utilisateur représente une personne s'étant inscrite sur le site.

- ID\_User : identifiant unique de l'utilisateur
  - Pseudo : nom de l'utilisateur sur le site
  - Email : adresse email de l'utilisateur qui sert d'identifiant pour se connecter
  - Password : mot de passe de l'utilisateur enregistré de manière encryptée
  - Organizer : spécifie si l'utilisateur souhaite être considéré comme un organisateur de tournoi (pour l'affichage du site)
  - Deleted : date à laquelle la personne a fait savoir sa volonté de supprimer son compte
- 

### JEU

---

L'entité Jeu représente un jeu pour lequel on peut créer un tournoi.

- ID\_Game : identifiant unique du jeu
  - Name : le nom du jeu
  - Deleted : indique que plus aucun tournoi ne peut être créé pour ce jeu
- 

### TOURNOI

---

L'entité Tournoi représente une compétition qui a été créée sur le site.

- ID\_Tournament : identifiant unique du tournoi
  - ID\_Game : identifiant du jeu pour lequel le tournoi a été créé
  - Name : nom du tournoi
  - Date : date et heure prévues pour le commencement du tournoi
  - Description : description du tournoi qui contient toutes les informations supplémentaires
  - MaxNumberPlayer : nombre maximum de joueurs pouvant participer au tournoi
  - DeckListNumber : nombre de *decks* à soumettre pour le tournoi
  - PPWin : points gagnés par victoire
  - PPDraw : points gagnés par égalité
  - PPLose : points gagnés par défaite
  - Over : précise si le tournoi est fini
  - Deleted : date à laquelle le tournoi a été annulé
- 

### DOTATION

---

L'entité Dotation représente les sommes qui seront gagnées par les joueurs en fonction de leur résultat final au tournoi.

- ID\_Tournament : identifiant du tournoi auquel la dotation se rapporte
- Place : place pour laquelle la dotation est définie
- Gain : somme gagnée

---

## RÉSULTAT

L'entité Résultat représente le classement de chaque utilisateur pour les tournois fini auxquels il a participé

- ID\_Tournament : identifiant du tournoi auquel le résultat se rapporte
- ID\_User : identifiant de l'utilisateur pour lequel le résultat se rapporte
- Rank : définit la place finale occupée par l'utilisateur quand le tournoi est fini
- Score : score final obtenu par l'utilisateur
- TieBreaker : score servant à départager les utilisateurs en cas d'égalité au score
- AdditinalTieBreaker : *tiebreaker*<sup>8</sup> additionnel servant à départager les utilisateurs si le *tiebreaker* n'a pas suffi
- AdditionalTieBrekerRules : explication du score obtenu pour le *tiebreaker* additionnel

---

## DECK

L'entité Deck représente la liste des cartes jouées par un joueur dans un de ses *decks*.

- ID\_Deck : identifiant unique du *deck*
- ID\_Game : identifiant du jeu auquel le *deck* se rapporte
- DeckList : la liste des cartes jouées dans le *deck*

---

## ROUND

L'entité Round représente une ronde (un tour) d'un tournoi.

- ID\_Tournament : identifiant du tournoi auquel la ronde se rapporte
- RoundNumber : numéro de la ronde
- Start : date et heure du début de la ronde
- End : date et heure de la fin de la ronde

---

## MATCH

L'entité Match représente un match entre 2 joueurs.

- ID\_Tournament : identifiant du tournoi auquel le match se rapporte
- RoundNumber : numéro de la ronde auquel le match se rapporte
- ID\_PlayerOne : identifiant du joueur 1 du match
- ID\_PlayerTwo : identifiant du joueur 2 du match (null si le joueur 1 a un *bye*<sup>9</sup>)

---

<sup>8</sup> Anglicisme pour dire départage, voir glossaire pour plus de détails.

<sup>9</sup> Anglicisme pour dire qu'un joueur ne participe pas à une ronde de tournoi, voir glossaire pour plus de détails.

---

## PARTIE

L'entité Partie représente un affrontement entre 2 joueurs.

- ID\_Tournament : identifiant du tournoi auquel la partie se rapporte
- RoundNumber : numéro de la ronde auquel la partie se rapporte
- ID\_PlayerOne : identifiant du joueur 1 de la partie
- ID\_PlayerTwo : identifiant du joueur 2 de la partie
- ID\_Deck\_PlayerOne : identifiant du *deck* joué par le joueur 1 pour la partie
- ID\_Deck\_PlayerTwo : identifiant du *deck* joué par le joueur 2 pour la partie
- ResultPart : donne le résultat de la partie

---

## ASSOCIATIONS

---

### ORGANISATEUR (TOURNOI - UTILISATEUR)

Associations entre Utilisateur et Tournoi. Un tournoi possède un ou plusieurs organisateurs.

- Level : niveau de référence pour les prises de décision concernant le tournoi (0 correspond au créateur)

---

### JOUEUR (TOURNOI - UTILISATEUR)

Association entre Utilisateur et Tournoi. Un tournoi possède de zéro à plusieurs joueurs. Ceux-ci s'inscrivant uniquement après la création du tournoi.

- RegisterDate : date et heure de l'inscription au tournoi
- CheckIn<sup>10</sup> : date et heure de la confirmation de présence
- Drop<sup>11</sup> : spécifie si le joueur a abandonné le tournoi

---

### PSEUDOIG (UTILISATEUR – JEU)

Association entre Utilisateur et Jeu. Pour chaque jeu un utilisateur peut avoir un pseudo différent.

- IG\_Pseudo : pseudo permettant de contacter l'utilisateur dans le jeu

---

### A POUR (TOURNOI – DOTATION)

Association entre Dotation et Tournoi. Un tournoi possède de zéro à plusieurs dotations, qui dépendent des places dites « payées ».

---

### DU (TOURNOI – JEU)

Association entre Jeu et Tournoi. Un tournoi est créé pour un seul jeu.

---

### EST CLASSÉ (UTILISATEUR – RÉSULTAT)

Association entre Utilisateur et Résultat. Un résultat appartient à un seul utilisateur.

---

<sup>10</sup> Anglicisme pour dire que l'on confirme sa présence, voir glossaire pour plus de détails

<sup>11</sup> Anglicisme pour dire abandonner un tournoi, voir glossaire pour plus de détails.

---

## AU (RÉSULTAT – TOURNOI)

Association entre Résultat et Tournoi. Un résultat appartient à un seul tournoi.

---

## DECKJOUEUR (UTILISATEUR – DECK – TOURNOI)

Association entre Utilisateur, Deck et Tournoi. Un *deck* est soumis par un seul utilisateur pour un seul tournoi.

---

## APPARTIENT (DECK – JEU)

Association entre Deck et Jeu. Un *deck* est créé pour un seul jeu.

---

## À (TOURNOI – ROUND)

Association entre Tournoi et Round. Une ronde est créée pour un seul tournoi.

---

## EST COMPOSÉ (ROUND – MATCH)

Association entre Round et Match. Un match est créé pour une seule ronde.

---

## PARTICIPE COMME JOUEUR1 (UTILISATEUR – MATCH)

Association entre Utilisateur et Match. Un match a toujours un seul joueur 1.

---

## PARTICIPE COMME JOUEUR2 (UTILISATEUR – MATCH)

Association entre Utilisateur et Match. Un match a maximum un seul joueur 2.

---

## MANCHE (MATCH – PARTIE)

Association entre Match et Partie. Une partie est créée pour un seul match.

---

## EST JOUÉ COMME DECK1 (DECK – PARTIE)

Association entre Deck et Partie. Une partie est jouée avec un seul *deck* du joueur 1.

---

## EST JOUÉ COMME DECK2 (DECK – PARTIE)

Association entre Deck et Partie. Une partie est jouée avec un seul *deck* du joueur 2.

---

## (UTILISATEUR – PARTIE)

Association entre Utilisateur et Partie. Une partie a toujours un seul joueur 1 (cette association n'est pas visible sur le schéma pour des soucis de lisibilité).

Utilisateur 0-n, Partie 1-1

---

## (UTILISATEUR – PARTIE)

Association entre Utilisateur et Partie. Une partie a toujours un seul joueur 2 (cette association n'est pas visible sur le schéma pour des soucis de lisibilité).

Utilisateur 0-n, Partie 1-1

# MODÈLE RELATIONNEL

## DIAGRAMME

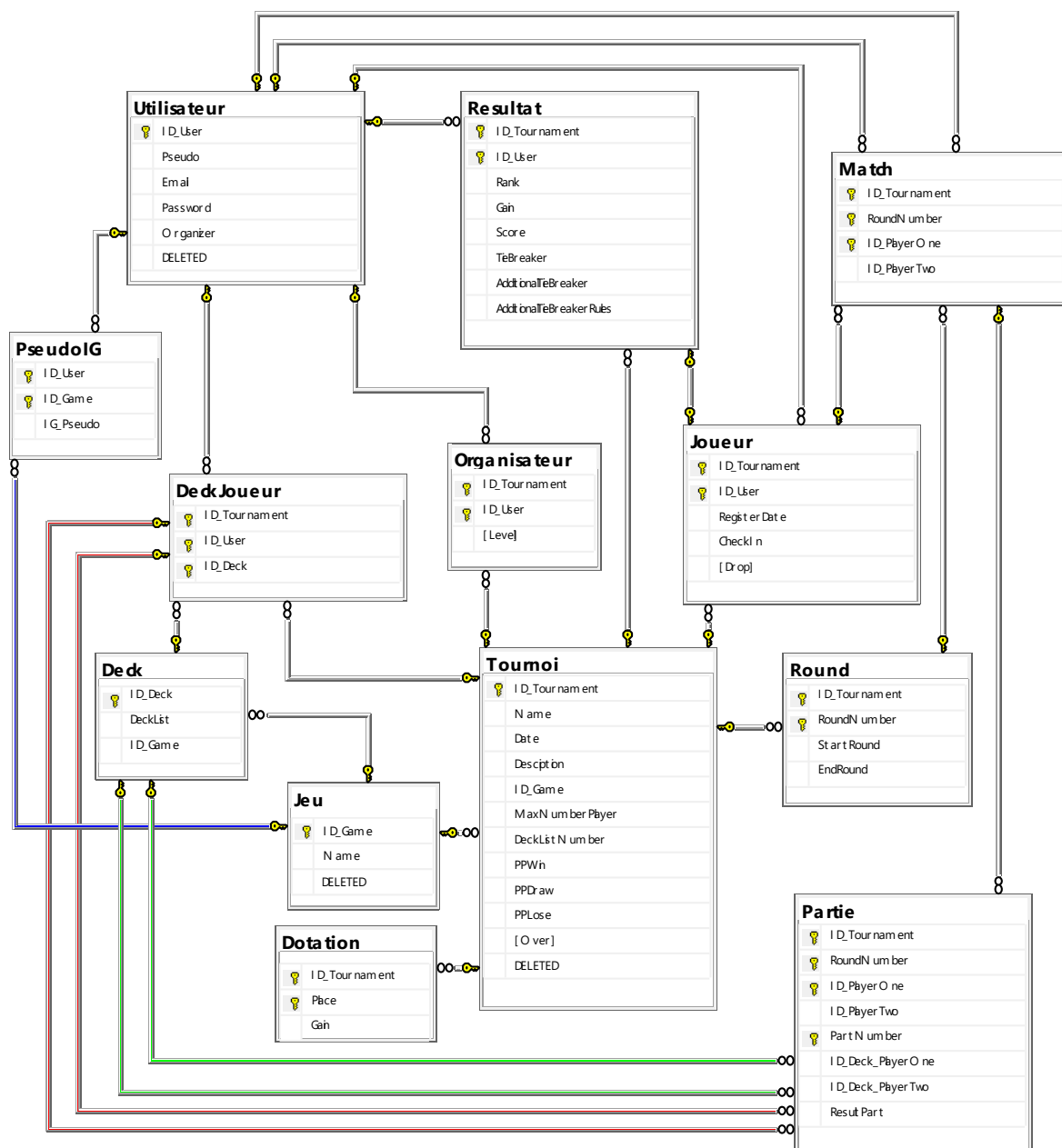


Figure 2 - Relationnel

---

## TABLES

---

### UTILISATEUR

- ID\_User : entier, non-null, identité auto-incrémentée
- Pseudo : chaîne de caractères de longueur maximale 50, non-null
- Email : chaîne de caractères de longueur maximale 256, non-null
- Password : données binaires de longueur 64, non-null
- Organizer : booléen, non-null, valeur par défaut : « faux »
- Deleted : date-heure, nullable, valeur par défaut : « null »

Contrainte de clé primaire : ID\_User

Contrainte d'unicité : Email

---

### JEU

- ID\_Game : entier, non-null, identité auto-incrémentée
- Name : chaîne de caractères de longueur maximale 50, non-null
- Deleted : booléen, non-null, valeur par défaut : « faux »

Contrainte de clé primaire : ID\_Game

Contrainte d'unicité : Name

---

### TOURNOI

- ID\_Tournament : entier, non-null, identité auto-incrémentée
- ID\_Game : entier, non-null
- Name : chaîne de caractères de longueur maximale 50, non-null
- Date : date-heure, non-null
- Description : texte, nullable
- MaxNumberPlayer : entier, nullable, valeur par défaut : « null »
- DeckListNumber : entier, non-null, valeur par défaut « 3 »
- PPWin : entier, non-null, valeur par défaut « 2 »
- PPDraw : entier, non-null, valeur par défaut « 1 »
- PPLose : entier, non-null, valeur par défaut « 0 »
- Over : booléen, non-null, valeur par défaut « faux »
- Deleted : date-heure, nullable, valeur par défaut « null »

Contrainte de clé primaire : ID\_Tournament

Contrainte de clé étrangère : ID\_Game, qui pointe vers la table Jeu



---

## PSEUDOIG

- ID\_User : entier, non-null
- ID\_Game : entier, non-null
- IG\_Pseudo : chaîne de caractères de longueur maximale 50, non-null

Contrainte de clé primaire : ID\_User + ID\_Game

Contraintes de clé étrangère : ID\_User, qui pointe vers la table Utilisateur

ID\_Game qui pointe vers la table Jeu

---

## DOTATION

- ID\_Tournament : entier, non-null
- Place : entier, non-null
- Gain : entier, non-null, valeur par défaut : « 0 »

Contrainte de clé primaire : ID\_Tournament + Place

Contrainte de clé étrangère : ID\_Tournament, qui pointe vers Tournoi

---

## DECK

- ID\_Deck : entier, non-null, identité auto-incrémentée
- ID\_Game : entier, non-null
- DeckList : texte, non-null

Contrainte de clé primaire : ID\_Deck

Contrainte de clé étrangère : ID\_Game, qui pointe vers Jeu

---

## RÉSULTAT

- ID\_Tournament : entier, non-null
- ID\_User : entier, non-null
- Rank : entier, non-null
- Gain : entier, non-null, valeur par défaut : « 0 »
- Score : entier, non-null
- TieBreaker : entier, non-null
- AdditionalTieBreaker : entier, nullable
- AdditionalTieBreakerRules : chaîne de caractères de longueur maximale 50, nullable

Contrainte de clé primaire : ID\_Tournament + ID\_User

Contraintes de clé étrangère : ID\_Tournament, qui pointe vers Tournoi

ID\_User, qui pointe vers Utilisateur

---

## ORGANISATEUR

- ID\_Tournament : entier, non-null
- ID\_User : entier, non-null
- Level : entier, non-null, valeur par défaut : « 1 »

Contrainte de clé primaire : ID\_Tournament + ID\_User

Contraintes de clé étrangère : ID\_Tournament, qui pointe vers Tournoi

ID\_User, qui pointe vers Utilisateur

---

## JOUEUR

- ID\_Tournament : entier, non-null
- ID\_User : entier, non-null
- RegisterDate : date-heure, non-null, valeur par défaut : maintenant
- CheckIn : date-heure, nullable, valeur par défaut : « null »
- Drop : booléen, non-null, valeur par défaut : « faux »

Contrainte de clé primaire : ID\_Tournament + ID\_User

Contraintes de clé étrangère : ID\_Tournament, qui pointe vers Tournoi

ID\_User, qui pointe vers Utilisateur

---

## DECKJOUEUR

- ID\_Tournament : entier, non-null
- ID\_User : entier, non-null
- ID\_Deck : entier, non-null

Contrainte de clé primaire : ID\_Tournament + ID\_User + ID\_Deck

Contraintes de clé étrangère : ID\_Deck, qui point vers la table Deck

ID\_Tournament + ID\_User, qui pointe vers la table Joueur

Contrainte d'unicité : ID\_Deck

---

## ROUND

- ID\_Tournament : entier, non-null
- RoundNumber : entier, non-null
- StartRound : date-heure, nullable
- EndRound : date-heure, nullable

Contrainte de clé primaire : ID\_Tournament + RoundNumber

Contrainte de clé étrangère : ID\_Tournament, qui pointe vers Tournoi

---

## MATCH

- ID\_Tournament : entier, non-null
- RoundNumber : entier, non-null
- ID\_PlayerOne : entier, non-null
- ID\_PlayerTwo : entier, nullable

Contrainte de clé primaire : ID\_Tournament + RoundNumber + ID\_PlayerOne

Contraintes de clé étrangère : ID\_Tournament + RoundNumber, qui pointe vers Round

ID\_PlayerOne, qui pointe vers Utilisateur

ID\_PlayerTwo, qui pointe vers Utilisateur (si non-null)

---

## PARTIE

- ID\_Tournament : entier, non-null
- RoundNumber : entier, non-null
- PartNumber : entier, non-null
- ID\_PlayerOne : entier, non-null
- ID\_PlayerTwo : entier, non-null
- ID\_Deck\_PlayerOne : entier, non-null
- ID\_Deck\_PlayerTwo : entier, non-null
- ResulPart : entier, nullable

Contrainte de clé primaire : ID\_Tournament + RoundNumber + ID\_PlayerOne + PartNumber

Contraintes de clé étrangère : ID\_Tournament + RoundNumber+ ID\_PlayerOne, qui pointe vers Match

ID\_PlayerOne, qui pointe vers Utilisateur

ID\_PlayerTwo, qui pointe vers Utilisateur

ID\_Deck\_PlayerOne, qui pointe vers Deck

ID\_Deck\_PlayerTwo, qui pointe vers Deck

---

## RÈGLES DE VALIDATION

Si la mention procédure est mise entre parenthèses, cela signifie que cette vérification se trouve au niveau des procédures qui affectent la table auxquelles la règle de validation se rapporte.

Si la mention « non-implementé » est mise entre parenthèses, cela signifie que cette vérification ne sera pas implémentée dans le projet final (sauf changement d'avis ultérieur).

Les autres mentions mises entre parenthèses sont là à titre informatif.

---

## UTILISATEUR

Email :

- Le champ email doit respecter le format d'une adresse email, à savoir 1 caractère ou plus suivi par un « @ » suivi d'au moins 2 caractères puis d'un « . » suivi d'au moins 2 caractères (« %\_@\_%.\_% »)
- Le champ Email doit être unique

---

## JEU

Name :

- Le champ Name doit être unique

---

## TOURNOI

Date :

- Dans le cadre de la création d'un tournoi, la date renseignée ne doit pas être passée (procédure)

MaxNumberPlayer :

- Le nombre de joueurs maximum doit être supérieur à 6 ou null.
- Dans le cadre de la création d'un tournoi, si le nombre renseigné est inférieur ou égal à 0, la valeur sera null (procédure)

DeckListNumber :

- La valeur doit être comprise entre 3 et 5 (bornes comprises)

PPWin :

- Logiquement la valeur de PPWin doit être supérieure ou égale à PPDrow (non-implémenté)

PPDraw :

- Logiquement la valeur de PPWin doit être supérieure ou égale à PPDraw (non-implémenté)
- Logiquement la valeur de PPDraw doit être supérieure ou égale à PPLose (non-implémenté)

PPLose :

- Logiquement la valeur de PPDraw doit être supérieure ou égale à PPLose (non-implémenté)

Over :

- Si un tournoi est fini il ne peut plus être modifié (procédure)

Deleted :

- Si un tournoi est annulé il ne peut plus être modifié à moins de rétracter son annulation (la rétractation n'est pas implémentée)

Autre :

- Un tournoi doit toujours avoir au moins 1 organisateur (procédure)

---

## PSEUDOIG

---

## DOTATION

---

## DECK

DeckList :

- Un *deck* ne peut plus être mis à jour après le début du tournoi pour lequel il a été soumis (procédure)

---

## RÉSULTAT

ID\_Tournament + ID\_User :

- Le résultat doit se reporter à un joueur

---

## ORGANISATEUR

ID\_Tournament :

- Tous les tournois doivent au moins avoir un organisateur de niveau 0 (procédure)
- Les organisateurs de niveau 0 ne peuvent pas être supprimés (procédure)

---

## JOUEUR

RegisterDate :

- Un joueur ne peut plus s'enregistrer ou se désister après la date de début d'un tournoi (Tournoi.Date > RegisterDate) (procédure)
- Un joueur ne peut plus s'inscrire à partir du moment où au moins un match a été créé pour un tournoi (non-implémenté)

---

## DECKJOUEUR

ID\_Deck :

- Doit être unique

ID\_Tournement + ID\_User :

- Le nombre de *decks* soumis ne doit pas dépasser le nombre de *decks* demandés (procédure) (cas possible dans le cas d'un changement du nombre de *decks* demandés => problème non géré)

---

## ROUND

StartRound :

- La date de début de ronde doit être avant la date de fin de ronde
- La date de début doit être après la date de fin de ronde précédente ou le début de tournoi si c'est la première ronde (procédure), (non-implémenté)

EndRound :

- La date de fin de ronde doit être supérieure à la date de début de ronde

---

## MATCH

ID\_Player (One/Two) :

- Doivent être des joueurs du tournoi
- Les 2 ID\_Player doivent être différents (procédure)
- A la création du match les joueurs ne doivent pas avoir *dropper*<sup>12</sup> (procédure)

---

## PARTIE

ID\_Tournement + RoundNumber + ID\_PlayerOne (+ ID\_PlayerTwo) :

- La partie doit se reporter à un match du tournoi (qui possède 2 joueurs)

ID\_Deck\_Player (One/Two) :

- Le *deck* doit être l'un de ceux soumis par le joueur correspondant

---

<sup>12</sup> Anglicisme pour dire abandonner, voir glossaire pour plus de détails

# CAS D'UTILISATION

## DIAGRAMME DE CAS D'UTILISATION GLOBALE

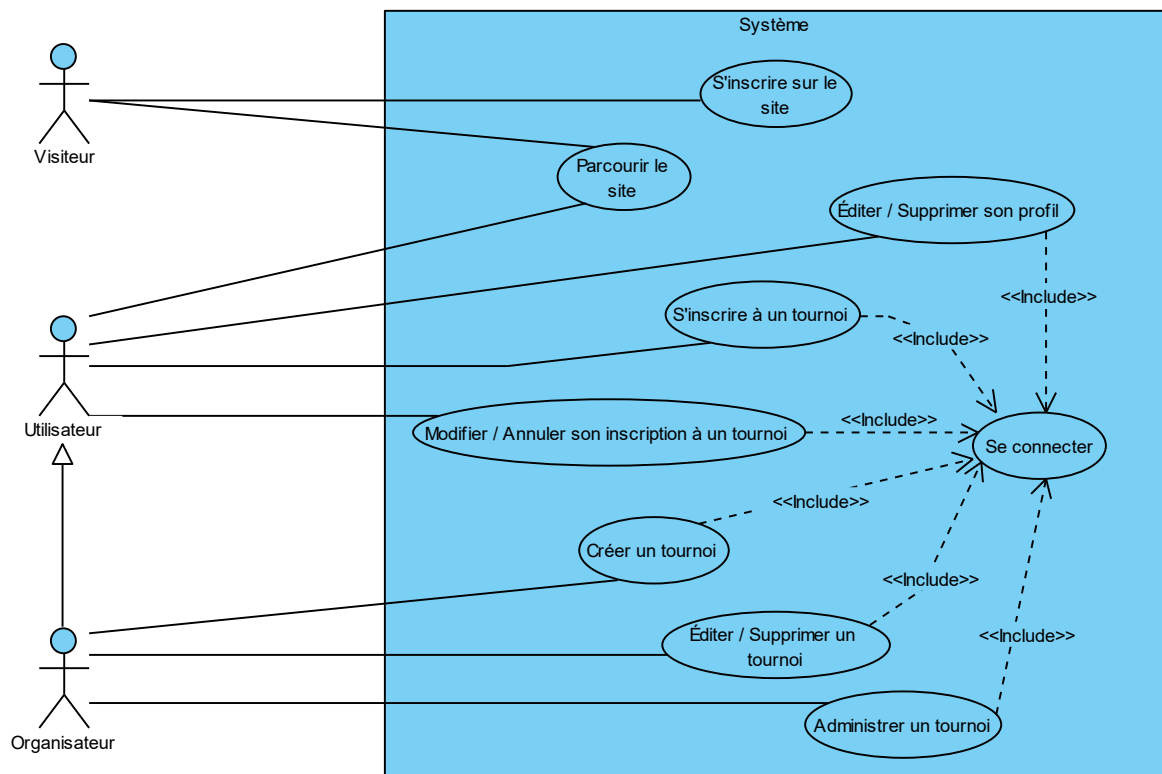


Figure 3 - Diagramme cas d'utilisation : Globale

---

## SCÉNARIOS DE CAS D'UTILISATION

---

### CAS D'UTILISATION : CONNEXION

Acteur : visiteur

But : se connecter comme Utilisateur / Organisateur

Précondition : avoir un compte

#### Scénario nominal

1. Le système demande l'email et le mot de passe de l'utilisateur
2. Le visiteur entre son email et son mot de passe
3. Le système recherche un utilisateur qui a pour email celui renseigné et dont le mot de passe correspond
4. Le système donne son accord pour la connexion
5. Le système met en mémoire les informations de l'utilisateur connecté

#### Scénario d'erreur

*E1 : Le système ne trouve pas de correspondance au point 3*

4. Le système refuse la connexion
5. Le système affiche un message d'erreur
6. Retour au point 1



## Diagramme de séquence

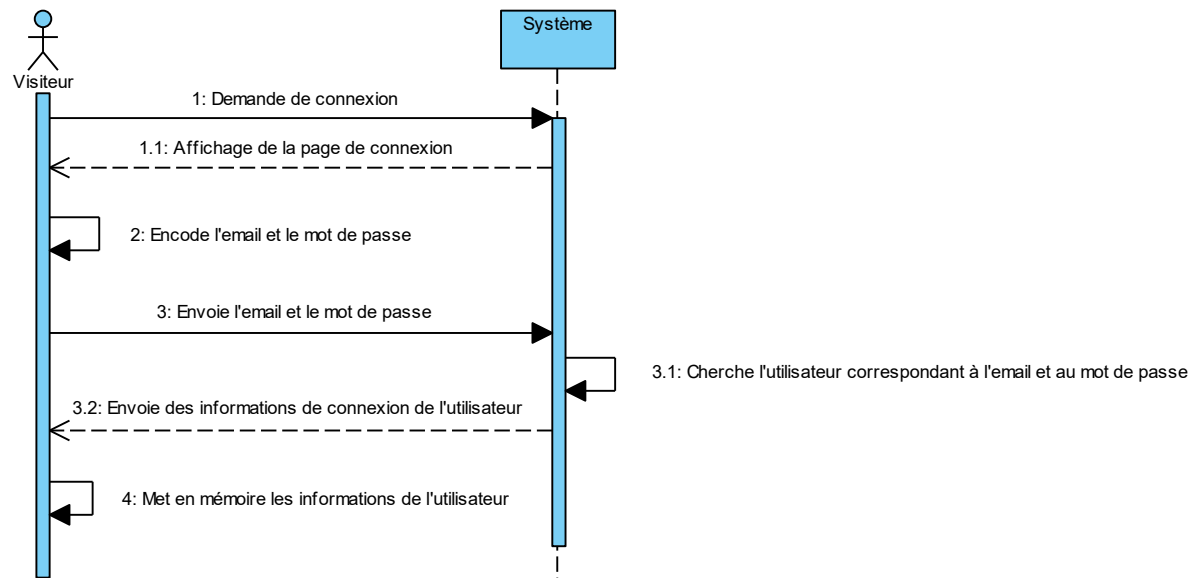


Figure 4 - Diagramme de séquence : Connexion

## Diagramme d'activité

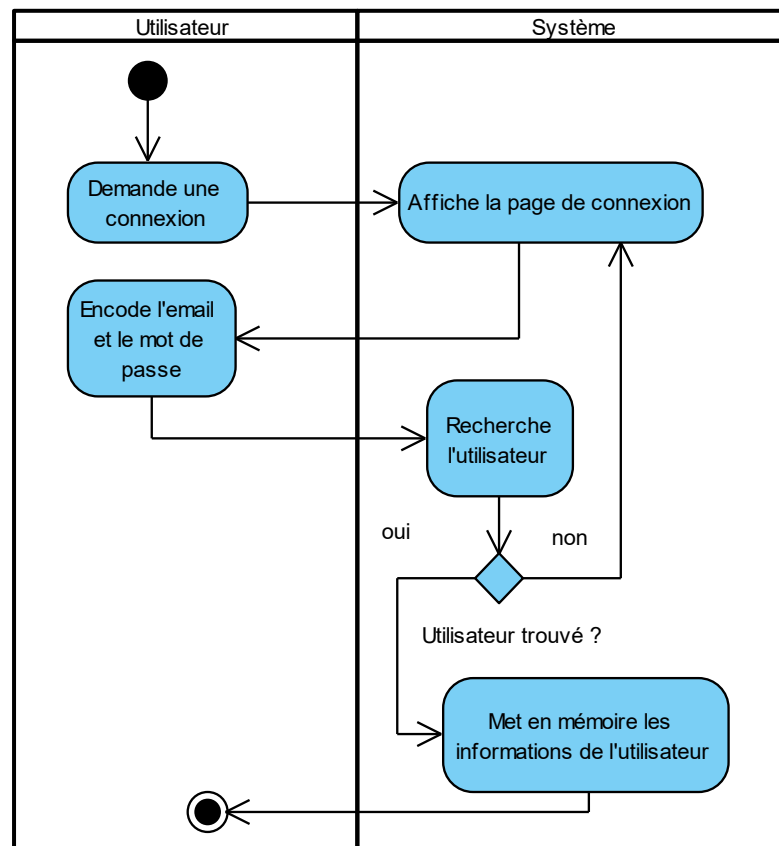


Figure 5 - Diagramme d'activité : Connexion

## CAS D'UTILISATION : ADMINISTRATION D'UN TOURNOI

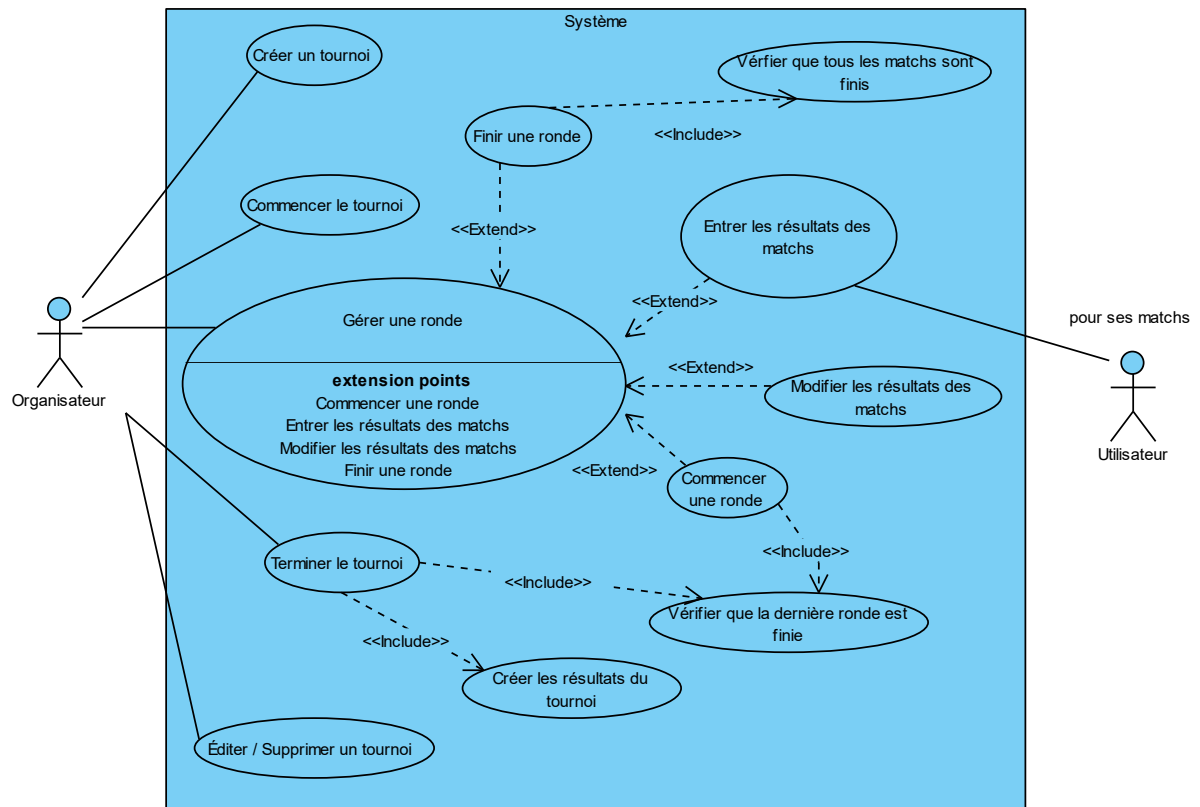


Figure 6 - Diagramme cas d'utilisation : Administration d'un tournoi

Acteur : organisateur

But : administrer un tournoi

Précondition : être organisateur du tournoi

### Scénario nominal

1. Le système récupère les données du tournoi
2. Le système affiche les données du tournoi et propose des actions.

### Scénario d'erreur

E1 : Le système ne trouve pas le tournoi au point 1

2. Le système affiche un message d'erreur

## Diagramme d'activité

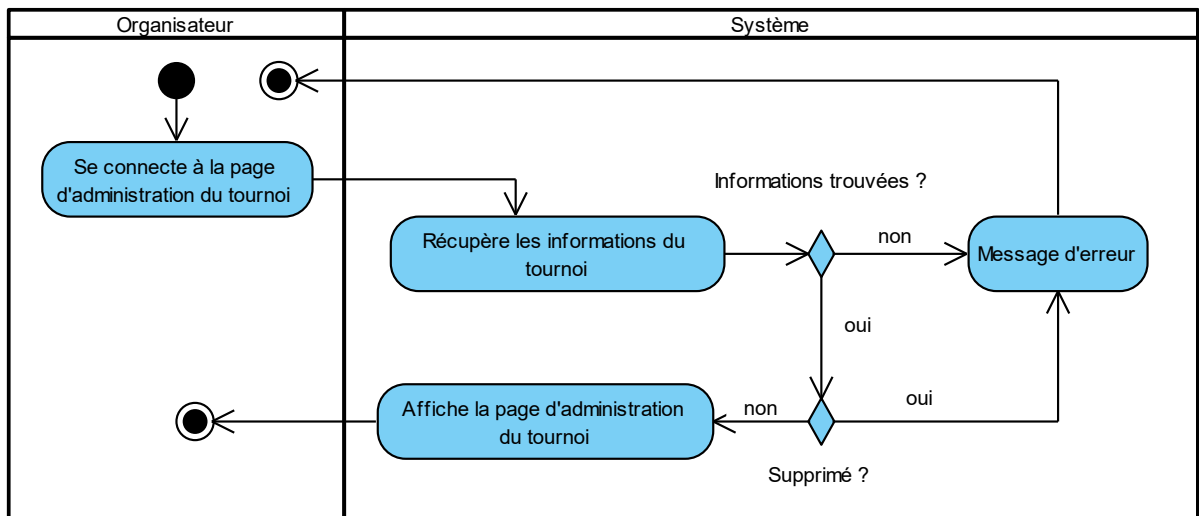


Figure 7 - Diagramme d'activité : Administration d'un tournoi

## GÉRER UNE RONDE : COMMENCER UNE RONDE

Acteur : organisateur

But : commencer une nouvelle ronde pour le tournoi

Précondition : être organisateur du tournoi

### Scénario nominal

3. L'organisateur choisit l'action « commencer une nouvelle ronde »
4. Le système vérifie que la dernière ronde est terminée, s'il y en a une
5. Le système crée une nouvelle ronde avec la date actuelle comme date de début de ronde
6. Le système redirige l'organisateur au point 1 et propose de faire le *pairings*

### Scénario d'erreur

E1 : La ronde précédente n'est pas encore finie, au point 4

5. Le système renvoie un message d'erreur
6. Retour au point 1

## Diagramme de séquence

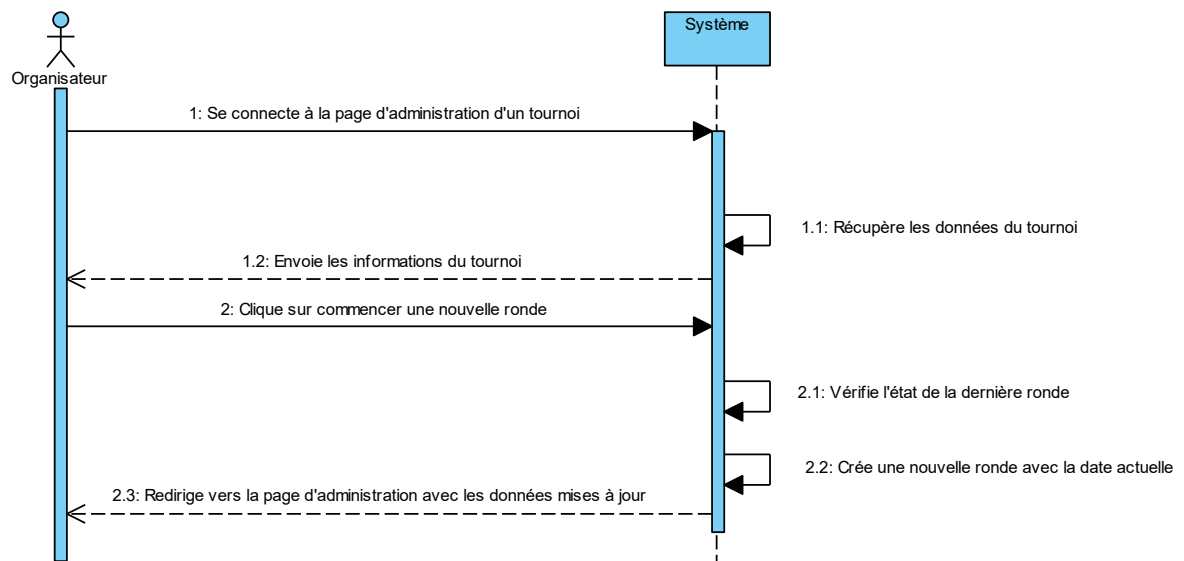


Figure 8 - Diagramme de séquence : Commencer une ronde

## Diagramme d'activité

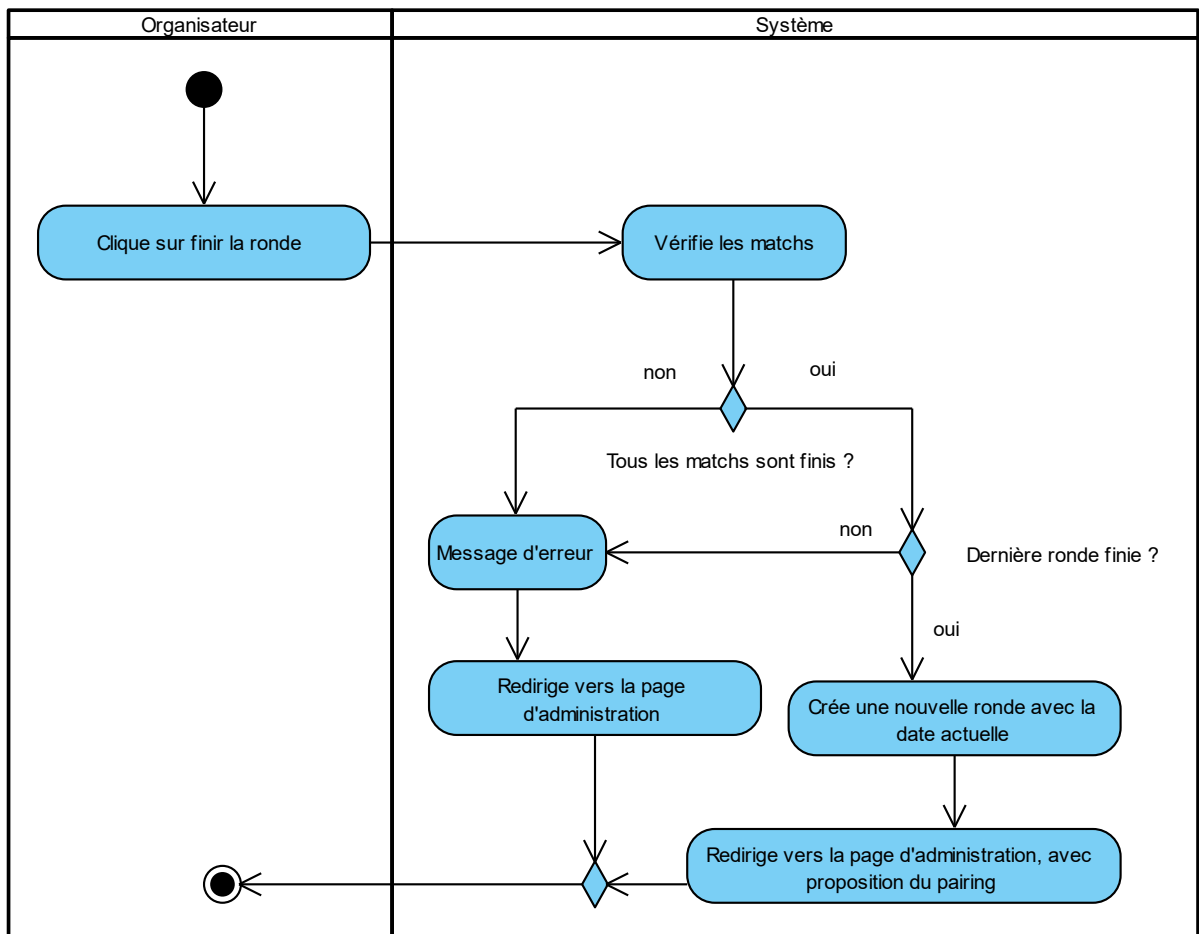


Figure 9 - Diagramme d'activité : Commencer une ronde

## GÉRER UNE RONDE : ENTRER / MODIFIER LES RÉSULTATS D'UN MATCH

Acteur : organisateur

But : encoder les résultats d'un match

Précondition : être organisateur du tournoi

### Scénario nominal

3. L'organisateur choisi l'action « Entrer / Modifier le résultat d'un match »
4. Le système récupère les données du match
5. Le système affiche les données du match dans la page de modification
6. L'organisateur encode les données
7. L'organisateur valide ses modifications
8. Le système enregistre les données encodées
9. Le système redirige l'organisateur au point 1

### Scénarios d'erreurs

*E1 : Le système ne trouve pas le match ou celui-ci ne peut plus être modifié, au point 4*

5. Le système affiche un message d'erreur
6. Retour au point 1

*E2 : L'organisateur a rentré des données incorrectes, au point 6*

8. Le système affiche un message d'erreur
9. Retour au point 5

*E3 : Le match ne peut plus être modifié, au point 7*

8. Le système affiche un message d'erreur
9. Retour au point 1

## Diagramme de séquence

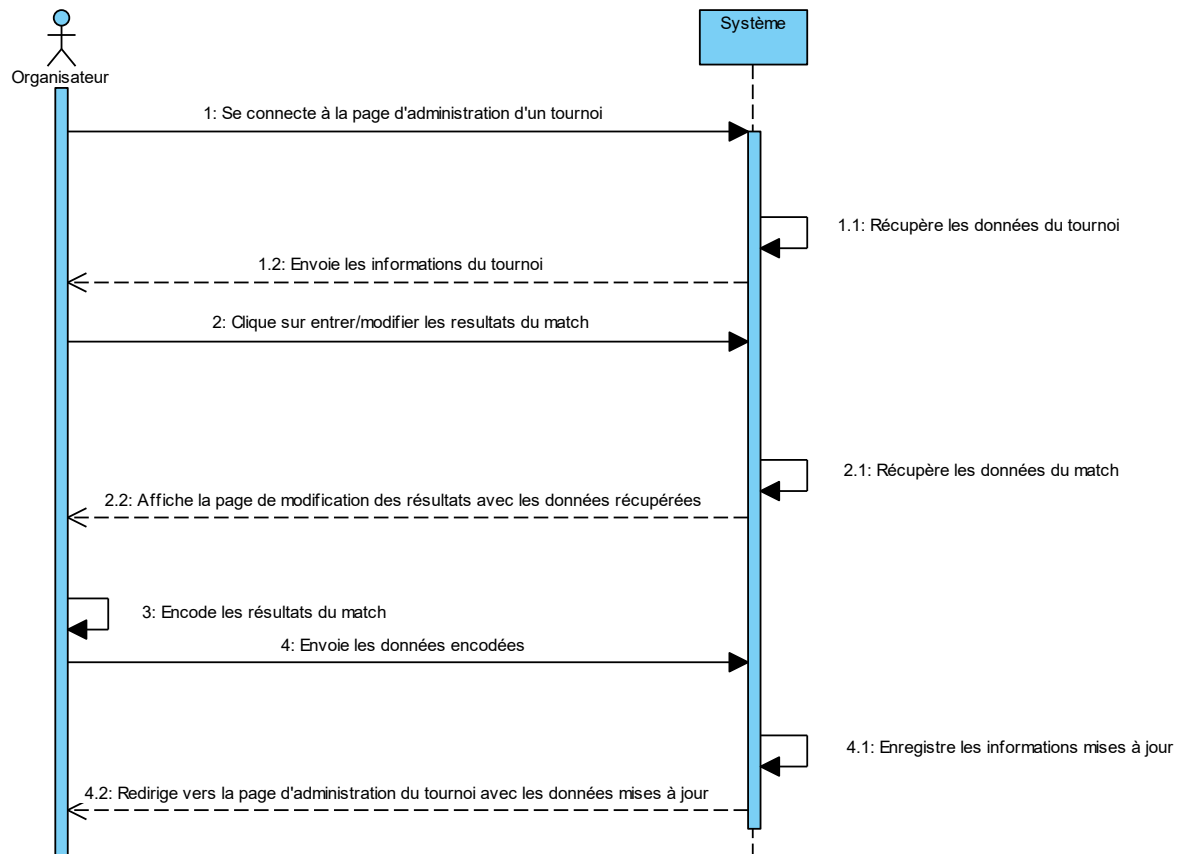


Figure 10 - Diagramme de séquence : Résultat match

## Diagramme d'activité

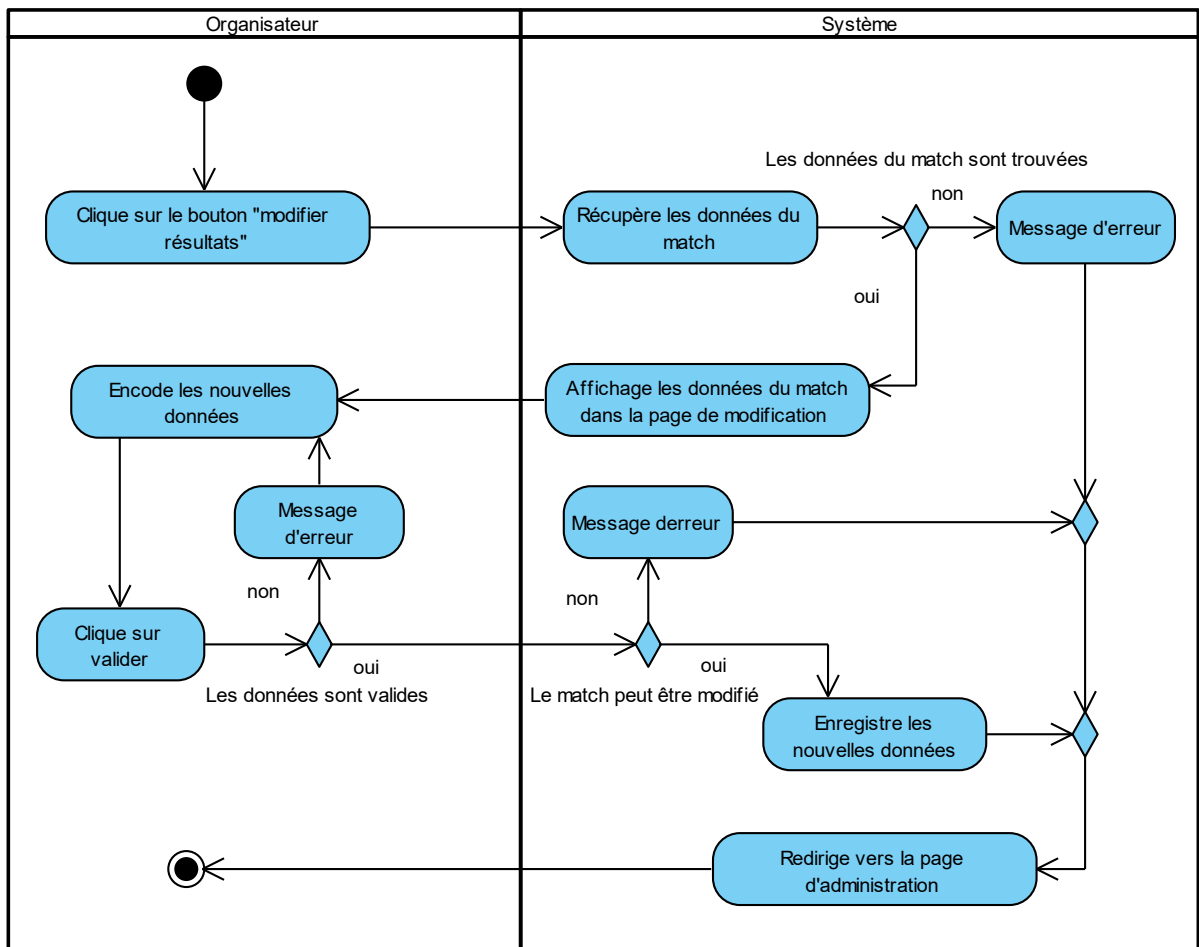


Figure 11 - Diagramme d'activité : Résultat match

## GÉRER UNE RONDE : FINIR UNE RONDE

Acteur : organisateur

But : finir la ronde en cours du tournoi

Préconditions : être organisateur du tournoi

avoir une ronde commencée

### Scénario nominal

3. L'organisateur choisit l'action « Finir la ronde »
4. Le système vérifie que tous les matchs sont finis
5. Le système enregistre la date actuelle comme date de fin de ronde
6. Retour au point 1

### Scénarios d'erreurs

E1 : Tous les matchs ne sont pas finis, au point 4

5. Le système affiche un message d'erreur
6. Retour au point 1

E2 : La ronde est déjà finie, au point 5

5. Le système ne modifie pas la ronde
6. Retour au point 1

### Diagramme de séquence

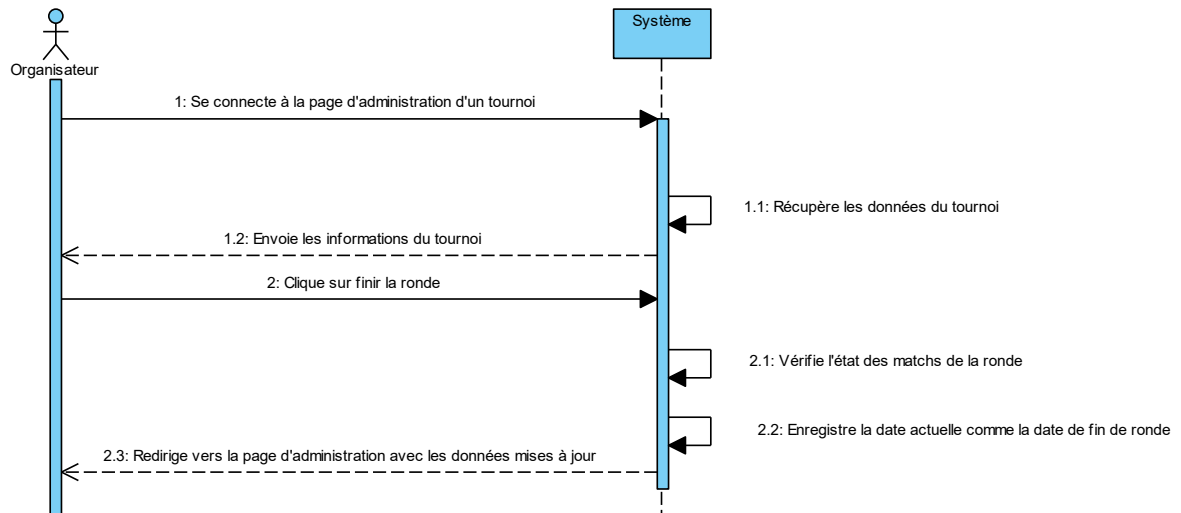


Figure 12 - Diagramme de séquence : Finir une ronde

### Diagramme d'activité

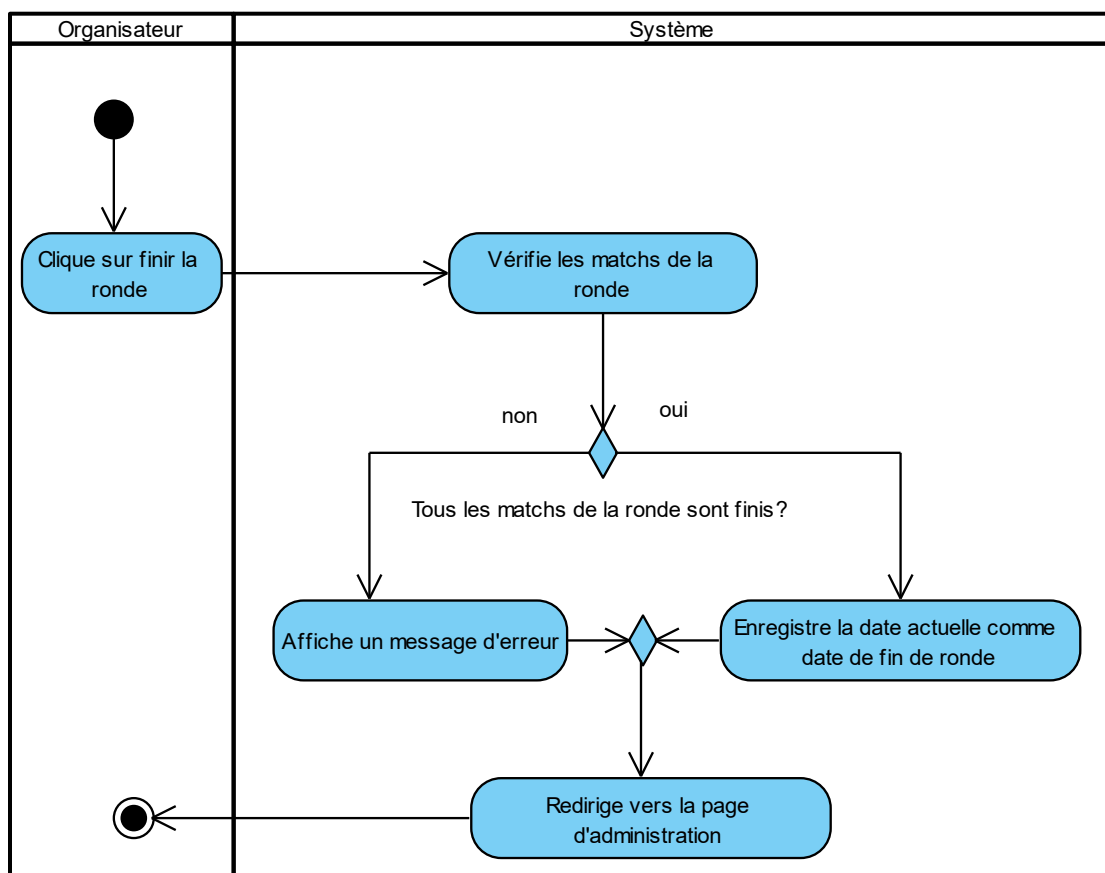


Figure 13 - Diagramme d'activité : Finir une ronde



## ALGORITHME : *PAIRINGS*

---

### FORMALISATION DU PROBLEME

---

#### OBJECTIF

---

- But : appareillé les joueurs de la ronde 2 à 2.
  - Restrictions/Conditions : cet appareillage doit être fait sur base du score de chaque joueur, dans le but que celui-ci affronte des joueurs avec un score équivalent. Mais il est aussi demandé qu'un joueur ne rencontre jamais 2 fois le même adversaire sur un tournoi.
  - Demande additionnelle : l'appareillage doit rester aléatoire entre les adversaires de même score qui ne se sont pas encore rencontrés.
- 

#### DONNEES NECESSAIRES

---

- Liste des joueurs de la ronde
  - Liste des précédents adversaires de chaque joueur
  - Score du joueur
- 

#### DONNEES A DISPOSITIONS

---

- Liste des joueurs du tournoi
- Jeu du tournoi
- Liste des organisateurs du tournoi
- Classement (temporaire) du tournoi
- Dotation du tournoi
- Liste des matchs du tournoi
- Liste des rondes du tournoi
- Nom du tournoi
- Description du tournoi
- Date du tournoi
- Point par victoire, défaite, égalité
- ...

---

## RESULTAT VOULU

- La liste des joueurs appareillé 2 à 2.

---

## CONDITIONS INITIALES PARTICULIERES OU PROBLEMATIQUES

- Si le nombre de joueurs est impair, un joueur ne pourra pas être appareillé : si ce n'est le cas que d'un seul et unique joueur pour la ronde, il peut être considéré comme étant appareillé.
- Un ou plusieurs joueurs ont déjà affronté tous les autres joueurs de la ronde : cela provoquera l'impossibilité de faire l'appareillement, sauf s'il coïncide avec le premier cas particulier évoqué.
- Lors de la première ronde le score de tous les joueurs est le même.

---

## ERREUR POSSIBLE.

- Les conditions initiales rendent le *pairings* impossible.

---

## INPUT

- Liste des ID des joueurs de la ronde
- Classement (temporaire) du tournoi
- Liste des matchs du tournoi
- Numéro de la ronde

---

## OUTPUT

- La liste des *pairings*

---

## OUTPUT EN CAS D'ERREUR

- Une liste de *pairings* vide

---

## EXPLICATION ET SCHEMATISATION DE L'ALGORITHME PRELIMINAIRE

Pour commencer notre algorithme reçoit la liste des joueurs de la ronde (Players), le classement des joueurs du tournoi (Classements), la liste des matchs du tournoi (Matches) et le numéro de la ronde (RoundNumber).

Ensuite, on initialise la variable de retour qui contiendra la liste des *pairings* (retour), on extrait du classement ('Classements') ceux des joueurs de la ronde ('Players') dans une nouvelle variable (playerClassement) et l'on crée une variable (playerOpponentList) qui contiendra la liste des adversaires de chaque joueur.

Pour ce faire, pour chaque joueur ('Players'), on va rechercher chaque match ('Matches') auquel il a participé puis ajouter son adversaire à la liste de ses adversaires

Puis, on va initialiser une variable qui contiendra la liste des joueurs reportés et créer des groupes de joueurs (groupPlayer) en les regroupant suivant leur score au classement temporaire ('playerClassement').

Après cela, pour chaque groupe de joueurs (Group) parmi les groupes créés ('groupPlayer'), et tant qu'il reste au moins un groupe (non traité), on va dans l'ordre :

1. Tenter d'appairer les joueurs reportés ('reportedPlayers') avec les joueurs du groupe ('Group') et pour chaque appariement réussi, on l'ajoute à la liste des *pairings* ('retour').
2. Supprimer les joueurs présents dans la liste des *pairings* du groupe ('Group') et des joueurs reportés ('reportedPlayer').
3. Tenter d'appairer les joueurs restants du groupe ('Group') entre eux et pour chaque appariement réussi, on l'ajoute à la liste des *pairings* ('retour').
4. Supprimer les joueurs présents dans la liste des *pairings* du groupe ('Group').
5. Ajouter les joueurs restants du groupe ('Group') à la liste des joueurs reportés ('reportedPlayers').

Pour appairer un joueur, on prend la liste des joueurs du groupe à laquelle on retire les adversaires déjà rencontrés par le joueur, et parmi les joueurs restants on en choisit un, au hasard. Les joueurs ainsi appairés ne doivent plus être considérés pour les appariements suivants.

Une fois, tous les groupes de joueurs ('Group') traités, on regarde le nombre de joueurs restants dans la liste des joueurs reportés ('reportedPlayers') :

- S'il n'est resté aucun, l'appareillement est fini.
- S'il en reste un seul, on l'ajoute à la liste des *pairings*, sans adversaire, on lui donne une *Bye*, l'appareillement est fini.
- S'il en reste plus d'un, l'appareillement n'a pas réussi à aboutir.

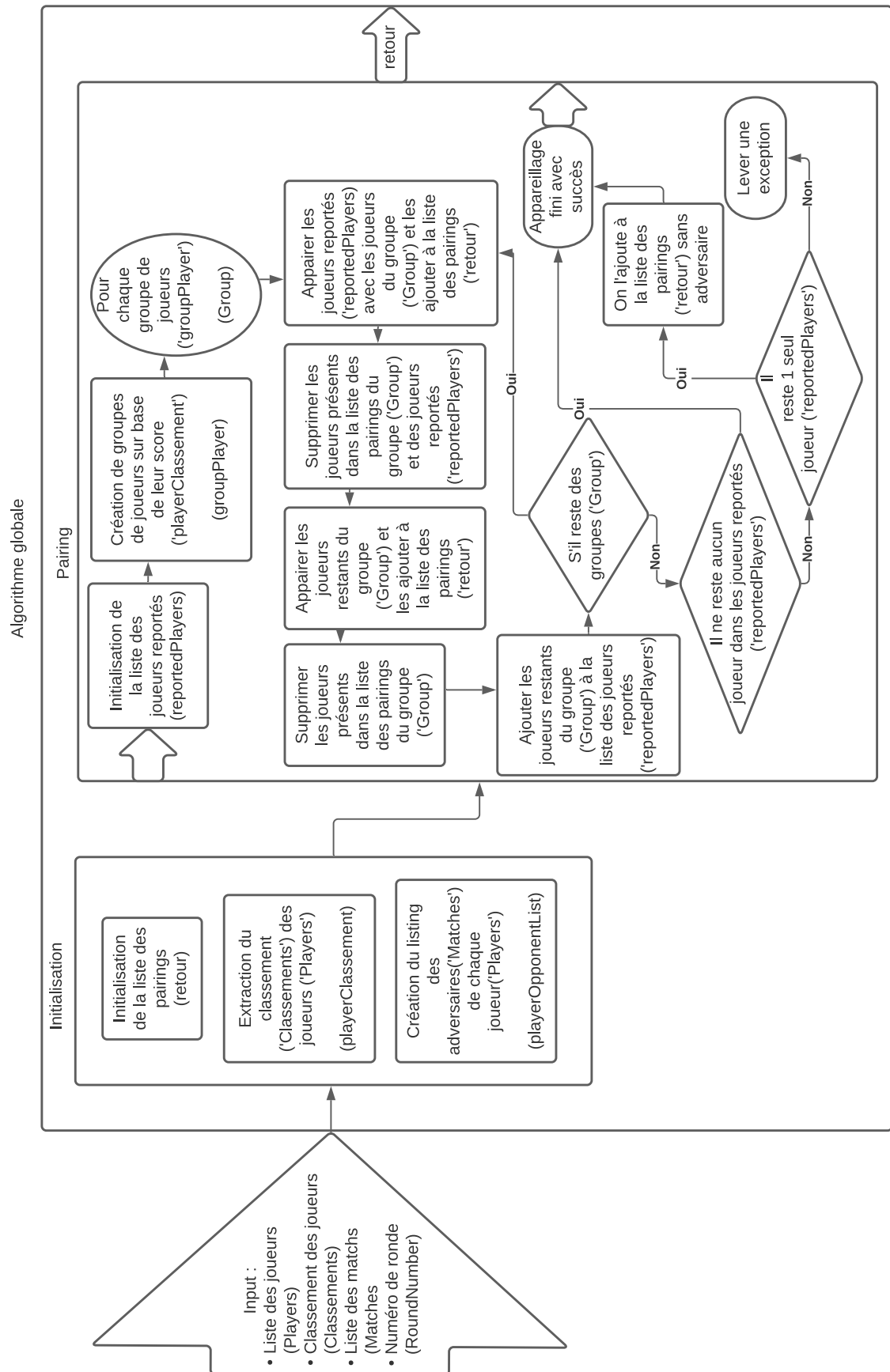


Figure 14 - Algorithme de pairings

# ARCHITECTURE DE L'APPLICATION

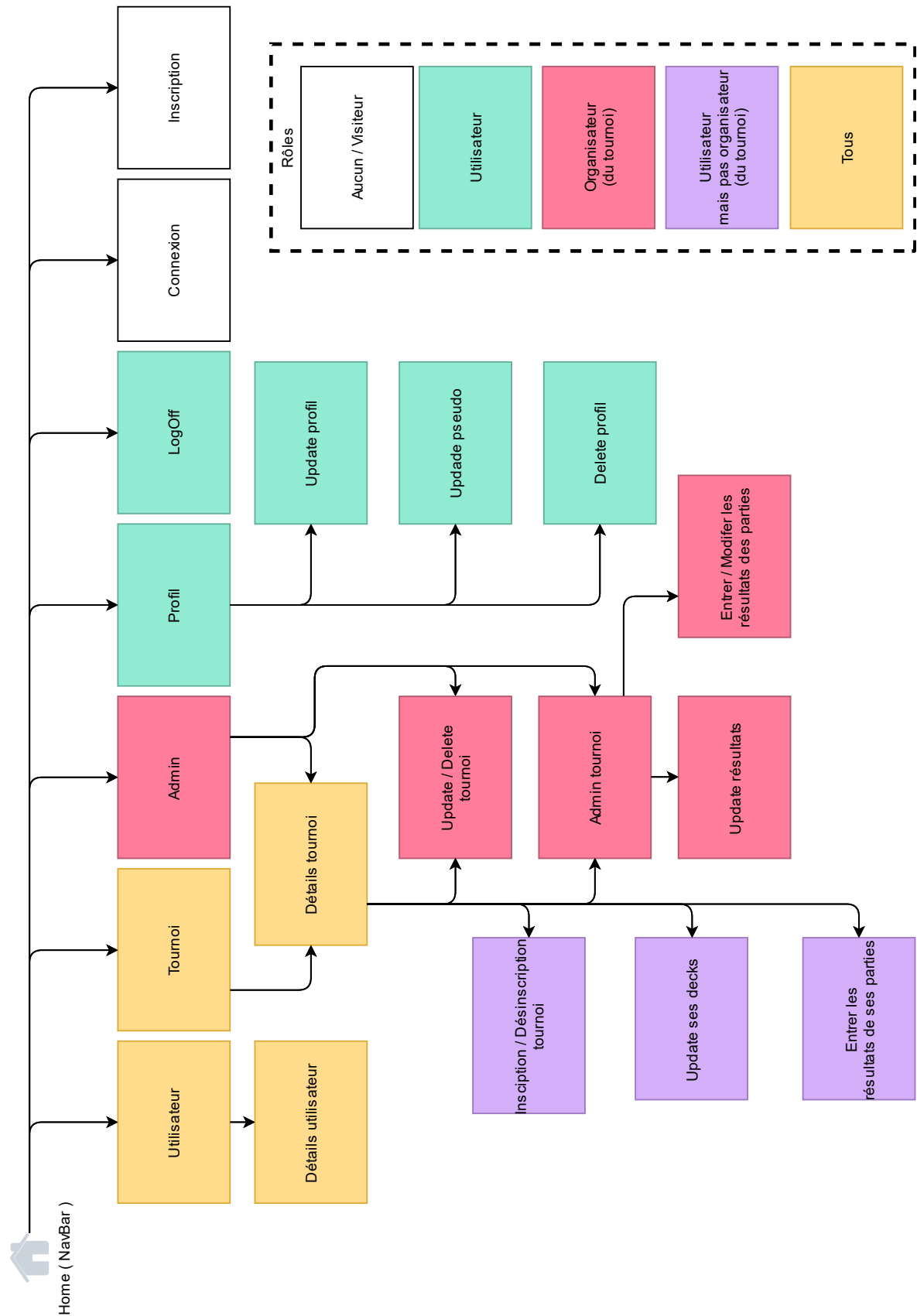


Figure 15 - Architecture du site web

# PROTOTYPES DE PAGES WEB

## HOME

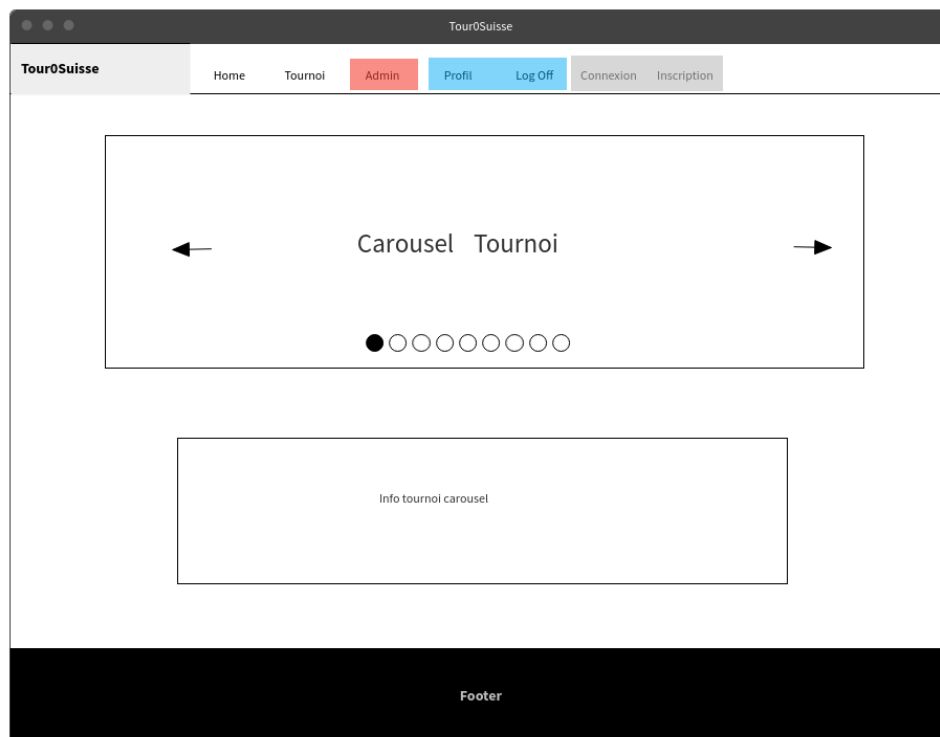


Figure 16 - Mockup : Home

## TOURNOI

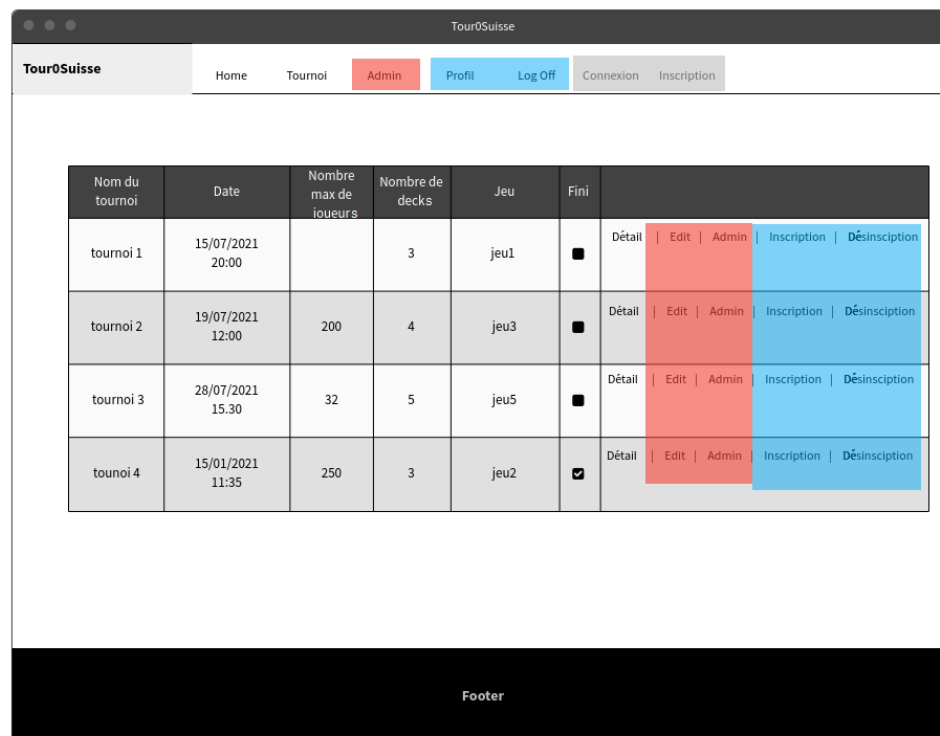


Figure 17 - Mockup : Tournoi

## PROFIL

Tour0Suisse

HomeTournoiAdminProfilLog OffConnexionInscription

Nom utilisateur

NomUtilisateur 1

Organisateur☐

Liste pseudos

Jeu	Pseudo
Jeu1	Pseudo1
Jeu3	Pseudo3
Jeu4	Pseudo4

Ajouter pseudo

Edit | liste joueurs

Footer

Figure 18 - Mockup : Profil

## ADMIN TOURNOI

Tour0Suisse

HomeTournoiAdminProfilLog OffConnexionInscription

Nom du tournoi	Date	Nombre de decks	Jeu
tournoi 1	15/07/2021 20:00	3	jeu 1

ronde 1ronde 2ronde 3ronde 4

joueur1	joueur2	
Janice Monahan	Koch and Sons	modifier
Rollin Fadel	Steuber LLC	modifier
Lera Stroman	Konopelski Group	modifier
Adan Schiller	Harber Inc	modifier

Créer la prochaine ronde

Terminer le tournoi

retour detail | liste

Footer

Figure 19 - Mockup : Admin Tournoi

Légende :

- Visiteur / Tout le monde
- Profil Utilisateur
- Profil organisateur
- Profil non connecter



## CONCLUSION

En réalisant mon travail de fin d'études, j'ai pu me rendre compte de l'importance d'effectuer une analyse claire et complète du projet, avant de me lancer dans sa phase de développement.

Une étude approfondie permet d'éviter certaines erreurs de logique, de conception voire de programmation, qui apparaissent, de manière plus ou moins évidente à la rédaction ou la lecture de l'analyse, alors qu'elles peuvent passer inaperçues au cours du développement.

Procéder ainsi permet d'anticiper d'éventuelles utilisations inutiles de ressources tant humaines que matérielles. L'impact économique de cette méthode de travail est par conséquent non négligeable.

Qui plus est, au fur et à mesure de l'avancement de mon projet, je me suis rendu compte qu'il aurait mieux valu ne pas être trop précis dans l'élaboration des détails de l'annexe 1. Certaines conditions présentent dans cette annexe, comme la restriction de 3 à 5 *decks*, en plus d'être superflue, empêche les utilisateurs de créer des tournois nécessitant plus ou moins de *decks*. Il aurait mieux valu que je dise « de 1 à plusieurs *decks* », cela aurait été une condition nettement moins restrictive.

À travers ce travail j'ai pu me rendre compte de l'intérêt de disposer de plusieurs modèles de données pour une même entité en fonction de la finalité de ce modèle.

Un premier modèle pour enregistrer les données, un second pour partager les données propres à l'entité affichée ainsi qu'un troisième pour partager les données propres et relatives à cette entité.

Cette pluralité des modèles montre également l'intérêt des interfaces. Celles-ci facilitent la mise en commun des données récurrentes pour les différents modèles et permettent ainsi de passer de l'un à l'autre plus simplement.

Cependant, j'ai pris conscience de l'importance de la multiplicité des modèles alors que le projet était déjà trop avancé pour que j'ai le temps de le recommencer en revoyant mon approche. J'ai donc quand cela était nécessaire, créé plusieurs modèles, sans pour autant le faire de manière systématique.

D'un point de vue personnel, ce projet m'a permis d'une part de me conforter dans l'intérêt que je porte au développement et d'autre part, d'approfondir mes connaissances en la matière.

J'ai pris plaisir à développer cet algorithme d'appareillement automatique, en le décomposant et le simplifiant, d'un problème complexe à plusieurs sous problèmes plus simples à résoudre.

J'ai également pu me rendre compte que ce qui me paraissait évident dans ma réflexion logique et intellectuelle ne s'appliquait pas toujours de manière si évidente au moment de passer à l'implémentation. Mettre en évidence les différentes problématiques que j'ai pu rencontrer dans ce type de projet me permet d'envisager le développement de manière plus sereine.

Au final, avec l'aide et le soutien de tous les intervenants de mon cursus, enrichi de nouvelles connaissances et grandi grâce aux rapports humains, j'en sors plus motivé que jamais !

## DÉFINITIONS ET GLOSSAIRE :

Une ronde suisse (système suisse) :

*« Le principe du tournoi suisse est que chaque joueur sera opposé à un adversaire qui a fait, jusqu'à présent, aussi bien (ou mal) que lui.*

*Le premier tour est déterminé soit par tirage au sort soit par têtes de série déterminées*

*Les joueurs qui gagnent reçoivent un point et les perdants ne reçoivent aucun point. Qu'ils gagnent, perdent, tous les joueurs poursuivent le tournoi au tour suivant où les gagnants seront opposés aux gagnants, les perdants aux perdants et ainsi de suite.*

*Le tournoi continue pendant un nombre de tours (appelés « rondes » aux échecs) déterminé avant le début du tournoi (généralement entre 3 et 9 tours). Après le dernier tour, les joueurs sont classés suivant leur score, en cas d'égalité, ils peuvent être départagés par la somme des scores de leurs adversaires ou par un autre système de départage déterminé à l'avance.*

*Au début du tournoi, les  $n$  joueurs sont classés selon leur force ou par tirage au sort. Pour le premier tour, les joueurs sont divisés en deux sous-groupes : un sous-groupe  $S1$  composé des joueurs 1 à  $n/2$ , et un sous-groupe  $S2$  composé des joueurs  $(n/2)+1$  à  $n$ . Le premier de  $S1$  joue contre le premier de  $S2$ , le deuxième de  $S1$  contre le deuxième de  $S2$ , et ainsi de suite de manière que le dernier joueur de  $S1$  joue contre le dernier joueur de  $S2$ .*

*En cas de nombre impair de participants le dernier est exempté et marque un point.*

*Après chaque ronde, on regroupe les participants ayant le même nombre de points et on recommence le processus décrit ci-dessus en veillant à ne jamais faire se rencontrer deux fois les mêmes adversaires et en s'arrangeant pour alterner au plus juste (nombre de parties avec les blancs proche de celui avec les noirs).*

*Lorsqu'un participant ne peut être apparié dans son groupe de points (s'il est seul ou s'il a déjà rencontré tous les autres) il est apparié dans le groupe de points le plus proche. »*

*(Système suisse - Wikipédia, 2021)<sup>13</sup>*

<sup>13</sup> [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_suisse#Principes](https://fr.wikipedia.org/wiki/Syst%C3%A8me_suisse#Principes)

### Système de départage SPA :

*« Est un système de départage qui se base sur les adversaires rencontrés. On somme le score de chaque adversaire que l'on a rencontré. Ces systèmes concernent les tournois qui admettent plus d'adversaires possibles que de rondes. Les participants ne rencontrent pas les mêmes adversaires. »*

*(Système de départage - Wikipédia, 2021)<sup>14</sup>*

### Deck :

*« Un deck est un terme utilisé par les joueurs de jeux de cartes à collectionner (ou de jeu de cartes évolutif) pour désigner le paquet de cartes qu'ils composent pour affronter d'autres joueurs.*

*La composition d'un deck dépend des cartes qu'un joueur possède dans sa collection.*

*Tous les jeux conseillent une limite au nombre de cartes à mettre dans un deck pour éviter les déséquilibres entre les joueurs. Par exemple, un deck de Magic the Gathering sera composé d'au moins 60 cartes alors qu'un deck de Yu-Gi-Oh! sera composé d'au moins 40 cartes. Pour Pokémon, il en faut exactement 60. »*

*(Deck (jeu) - Wikipédia, 2021)<sup>15</sup>*

### Framework :

*« En programmation informatique, un framework (appelé aussi infrastructure logicielle, infrastructure de développement, environnement de développement, socle d'applications, cadre d'applications ou cadriceil) désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture). »*

*(Framework - Wikipédia, 2021)<sup>16</sup>*

### Pseudo :

*« Diminutif de pseudonyme*

*Un pseudonyme est un nom d'emprunt adopté par une personne pour exercer une activité sous un autre nom que celui de son identité officielle. »*

*(Pseudonyme - Wikipédia, 2021)<sup>17</sup>*

---

<sup>14</sup> [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_d%C3%A9partage](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_d%C3%A9partage)

<sup>15</sup> [https://fr.wikipedia.org/wiki/Deck\\_\(jeu\)](https://fr.wikipedia.org/wiki/Deck_(jeu))

<sup>16</sup> <https://fr.wikipedia.org/wiki/Framework>

<sup>17</sup> <https://fr.wikipedia.org/wiki/Pseudonyme>

**Dropper :**

*« Un joueur peut dropper un tournoi (le quitter) à tout moment. »*

*(MTG\_MTR\_21Jan19\_FR.pdf, 2021)<sup>18</sup>*

**Bye :**

*« Le terme bye est un anglicisme parfois utilisé pour signifier qu'un joueur ou une équipe ne participe pas à un tour de jeu. »*

*(Bye - Wikipédia, 2021)<sup>19</sup>*

**Implémenter :**

*« En ingénierie et plus particulièrement en informatique, la mise en œuvre désigne la création d'un produit fini à partir d'un document de conception, d'un document de spécification, voire directement depuis une version originelle ou un cahier des charges.*

*L'utilisation de l'anglicisme « implémentation », de l'anglais to implement, est courante (et acceptée) et reflète la volonté de ne pas tomber dans la traduction ambiguë que serait l'utilisation du verbe implanter. »*

*(Mise en œuvre - Wikipédia, 2021)<sup>20</sup>*

**Check-In :**

*« Confirmation de sa présence et ça participation à un tournoi/ronde au début de celui-ci. »*

**Tiebreaker :**

*« Un système de départage est une manière de départager plusieurs joueurs à égalité.*

*L'utilisation de l'anglicisme « tiebreaker » ou « tie-breaker » est courante (et acceptée) dans les tournois de cartes, ainsi que les tournois sur internet, dû à l'utilisation prédominante de l'anglais. »*

---

<sup>18</sup> [https://blogs.magicjudges.org/translatedrules/files/2019/01/MTG\\_MTR\\_21Jan19\\_FR.pdf](https://blogs.magicjudges.org/translatedrules/files/2019/01/MTG_MTR_21Jan19_FR.pdf)

<sup>19</sup> <https://fr.wikipedia.org/wiki/Bye>

<sup>20</sup> [https://fr.wikipedia.org/wiki/Mise\\_en\\_œuvre](https://fr.wikipedia.org/wiki/Mise_en_œuvre)

# SOURCES ET BIBLIOGRAPHIE

TFE consulté :

- 2019-2020 – Jamotte Éric – Application de gestion d'une fédération de basket-ball – Rapporteur F. Ferrant – Section Informatique de gestion EPHEC

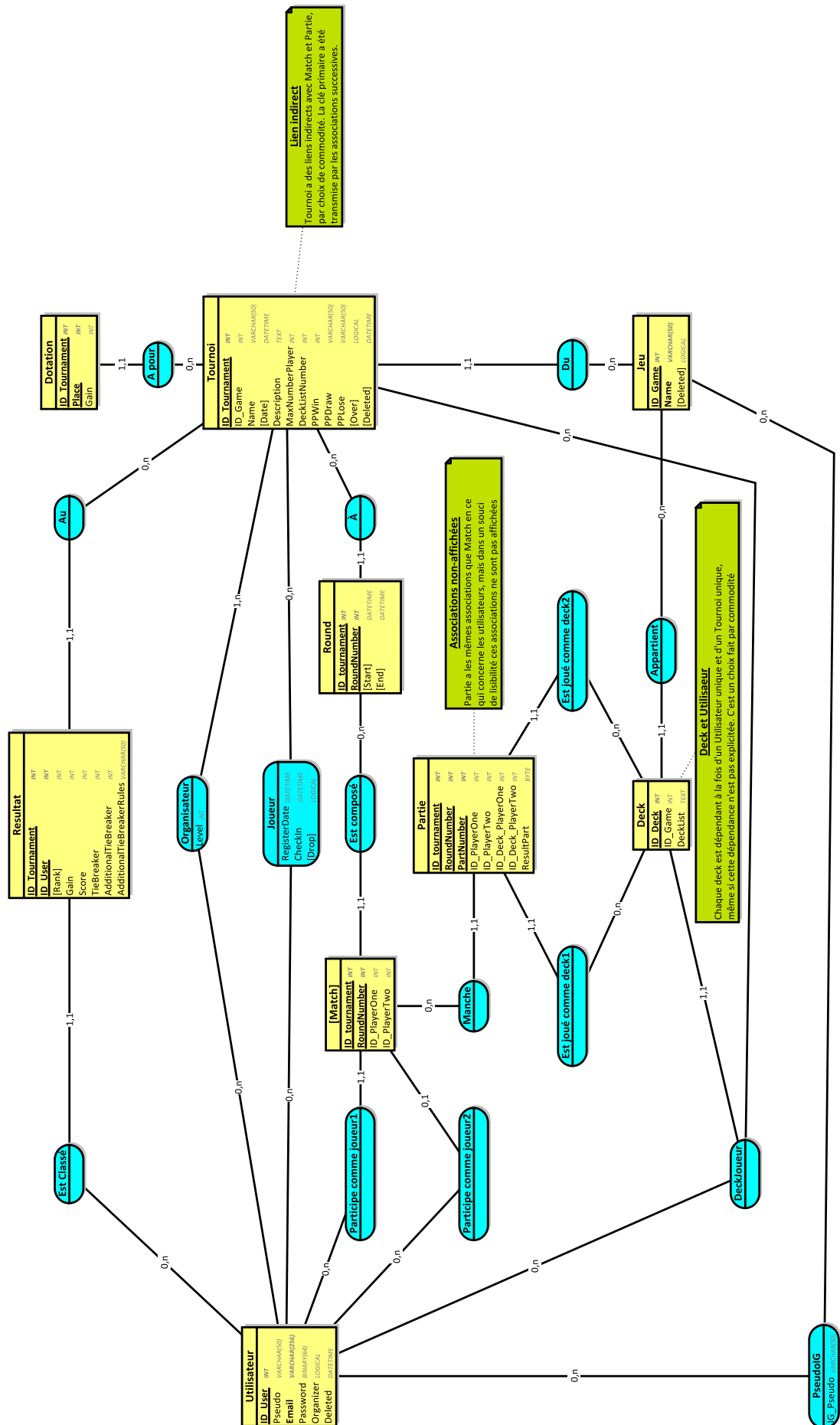
Site consultés ou utilisés :

- <https://www.wikipedia.org/>  
[site consulté le 28/04/2021]
- <https://docs.microsoft.com/>  
[site consulté le 28/04/2021]
- <https://stackoverflow.com/>  
[site consulté le 28/04/2021]
- <https://openclassrooms.com/>  
[site consulté le 28/04/2021]
- <https://www.c-sharpcorner.com/>  
[site consulté le 28/04/2021]
- <https://www.codegrepper.com/>  
[site consulté le 28/04/2021]
- <https://www.learnrazorpages.com/>  
[site consulté le 28/04/2021]
- <https://www.w3schools.com/>  
[site consulté le 28/04/2021]
- <https://getbootstrap.com/>  
[site consulté le 28/04/2021]
- <https://github.com/>  
[site consulté le 28/04/2021]
- <https://www.mockflow.com/>  
[site consulté le 28/04/2021]
- <https://battlefy.com/>  
[site consulté le 27/04/2021]
- <https://lucid.co/>  
[site consulté le 30/04/2021]

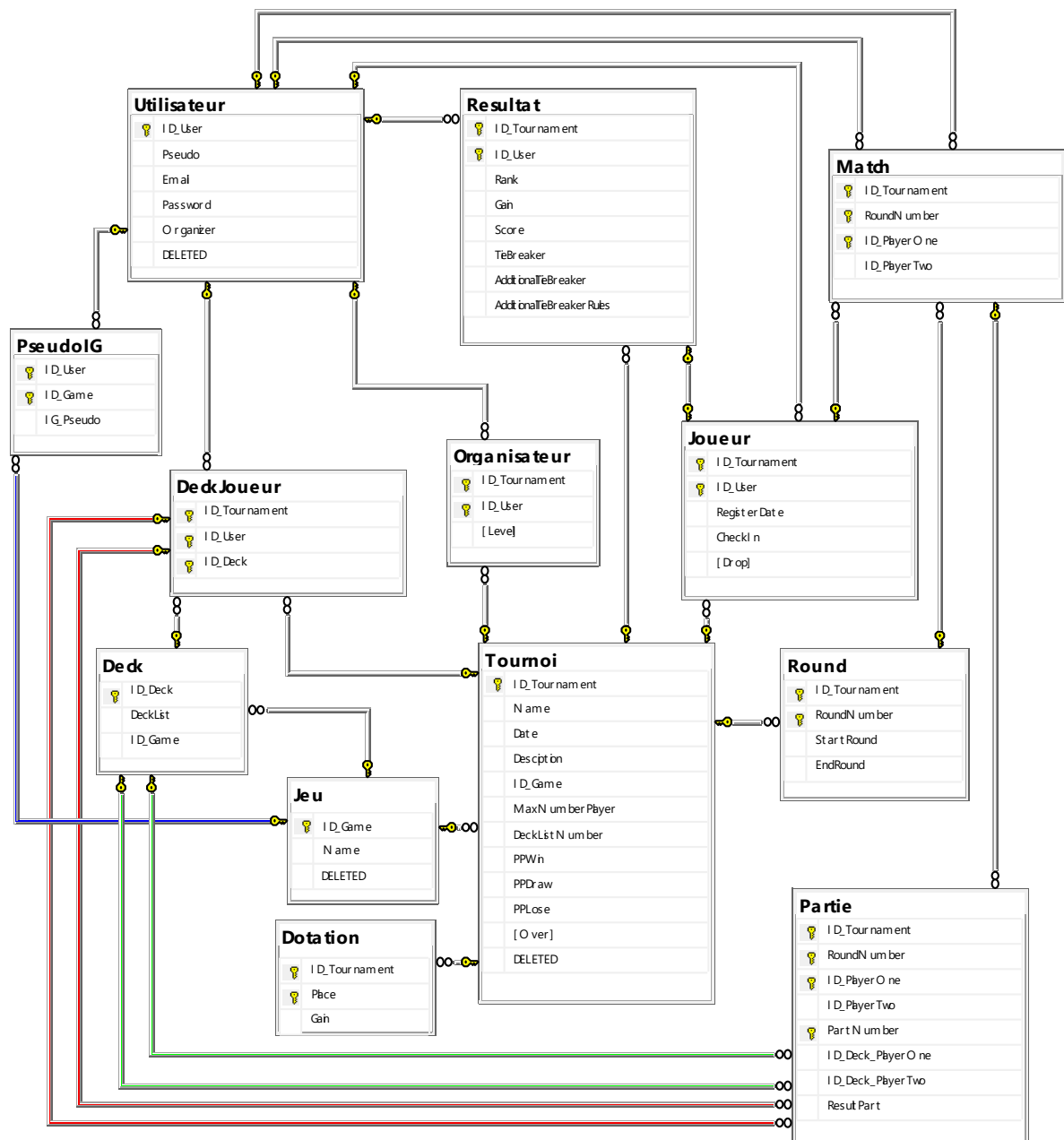
Source des citations :

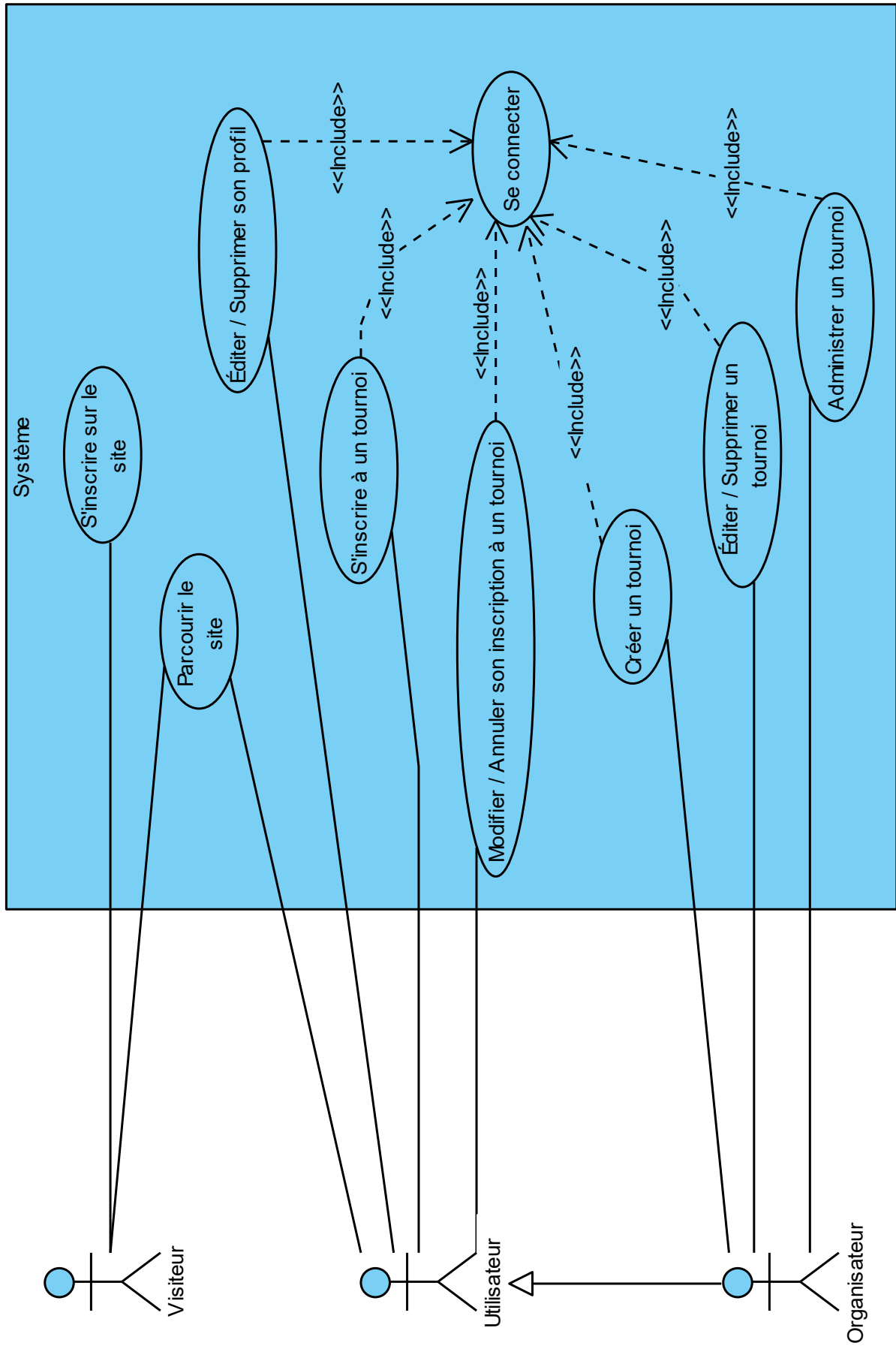
- MTG\_MTR\_21Jan19\_FR.pdf. (2021, 04 28). Récupéré sur Magic Judge News :  
[https://blogs.magicjudges.org/translatedrules/files/2019/01/MTG\\_MTR\\_21Jan19\\_FR.pdf](https://blogs.magicjudges.org/translatedrules/files/2019/01/MTG_MTR_21Jan19_FR.pdf)
- Bye - Wikipédia. (2021, 04 28). Récupéré sur Wikipédia :  
<https://fr.wikipedia.org/wiki/Bye>
- Deck (jeu) - Wikipédia. (2021, 04 25). Récupéré sur Wikipédia :  
[https://fr.wikipedia.org/wiki/Deck\\_\(jeu\)](https://fr.wikipedia.org/wiki/Deck_(jeu))
- Framework - Wikipédia. (2021, 04 25). Récupéré sur Wikipédia :  
<https://fr.wikipedia.org/wiki/Framework>
- Mise en œuvre - Wikipédia. (2021, 04 28). Récupéré sur Wikipédia :  
[https://fr.wikipedia.org/wiki/Mise\\_en\\_œuvre](https://fr.wikipedia.org/wiki/Mise_en_œuvre)
- Pseudonyme - Wikipédia. (2021, 04 25). Récupéré sur Wikipédia :  
<https://fr.wikipedia.org/wiki/Pseudonyme>
- Système de départage - Wikipédia. (2021, 04 02). Récupéré sur Wikipédia :  
[https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_d%C3%A9partage](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_d%C3%A9partage)
- Système suisse - Wikipédia. (2021, 04 02). Récupéré sur Wikipédia :  
[https://fr.wikipedia.org/wiki/Système\\_suisse#Principes](https://fr.wikipedia.org/wiki/Système_suisse#Principes)

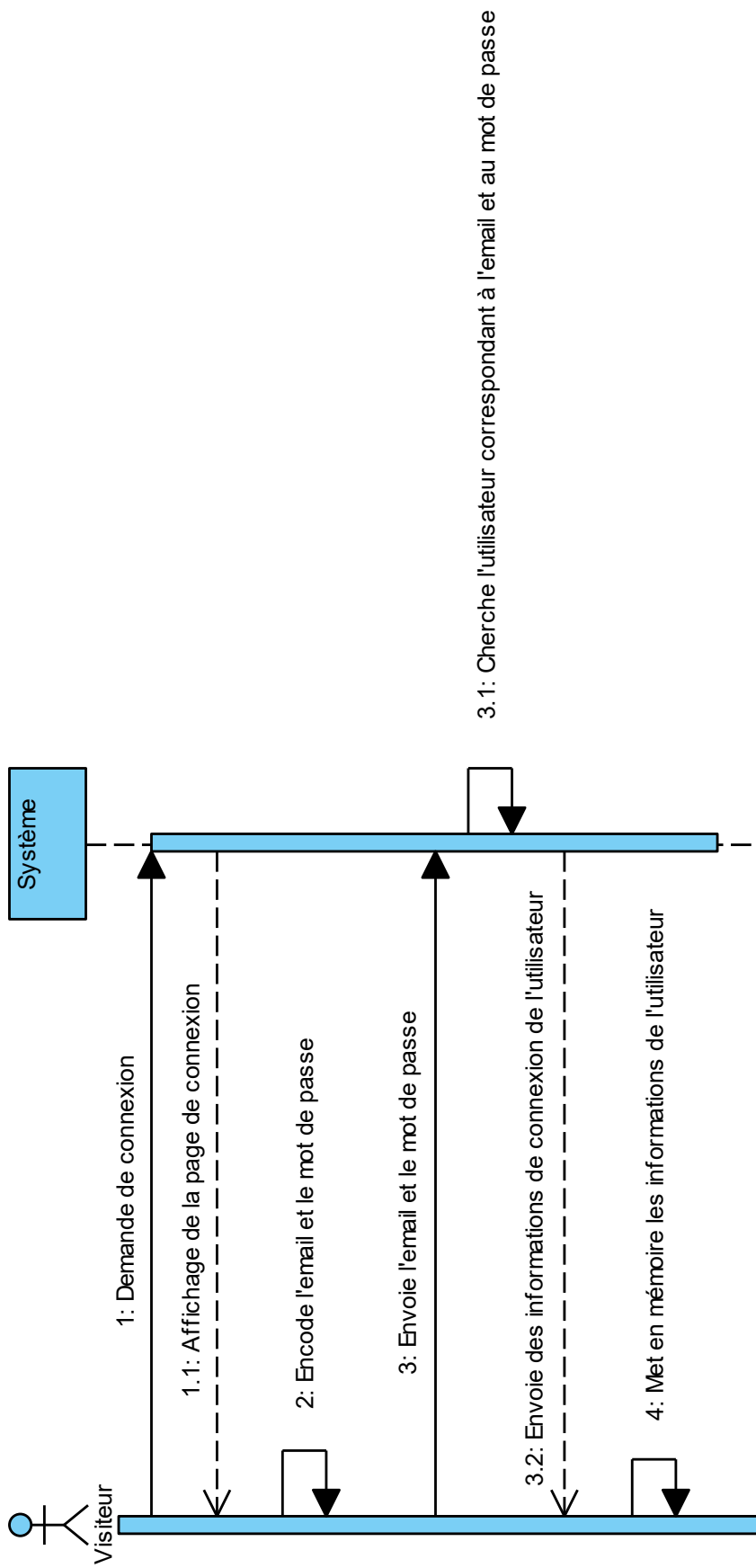
# ANNEXES

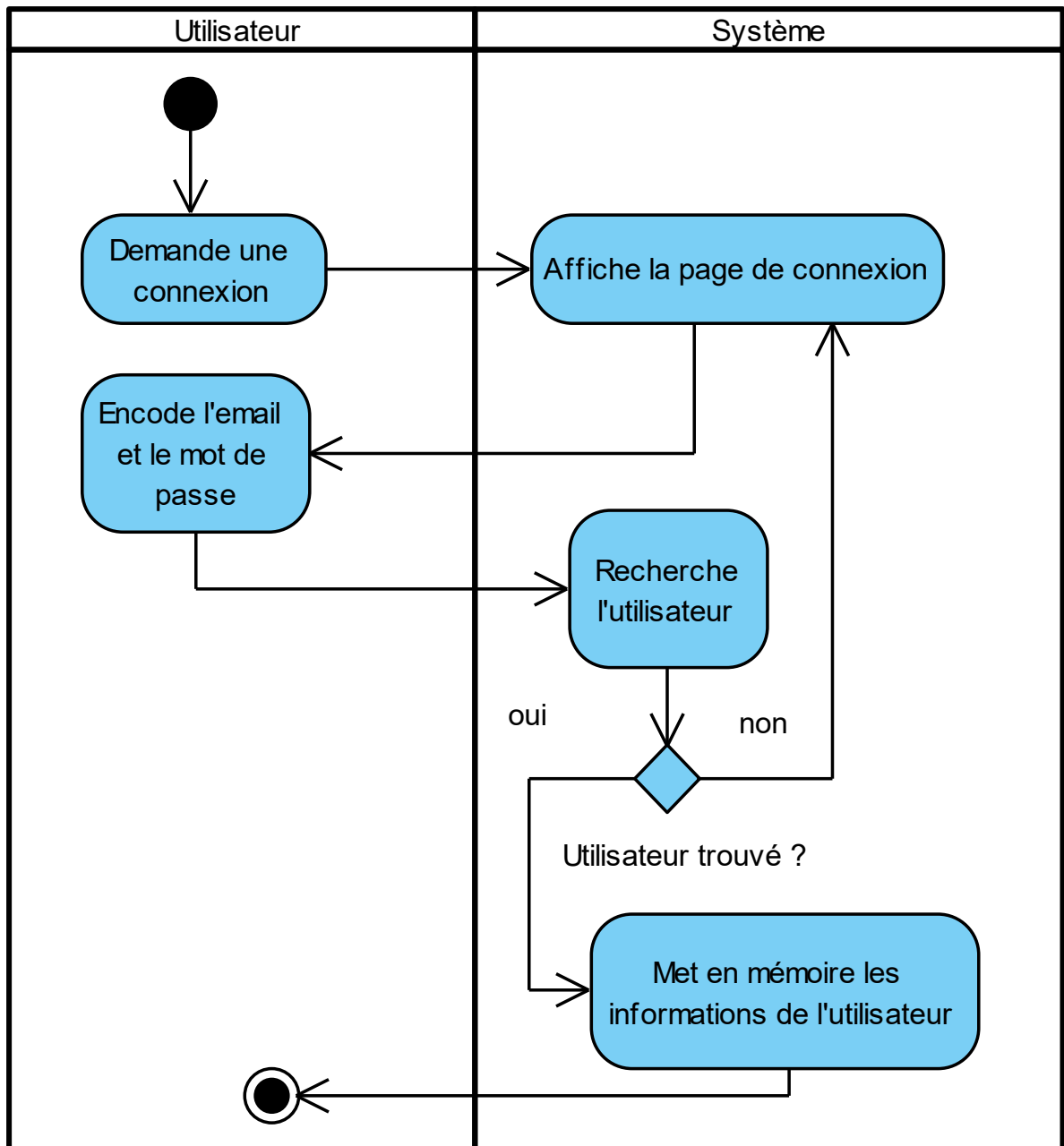


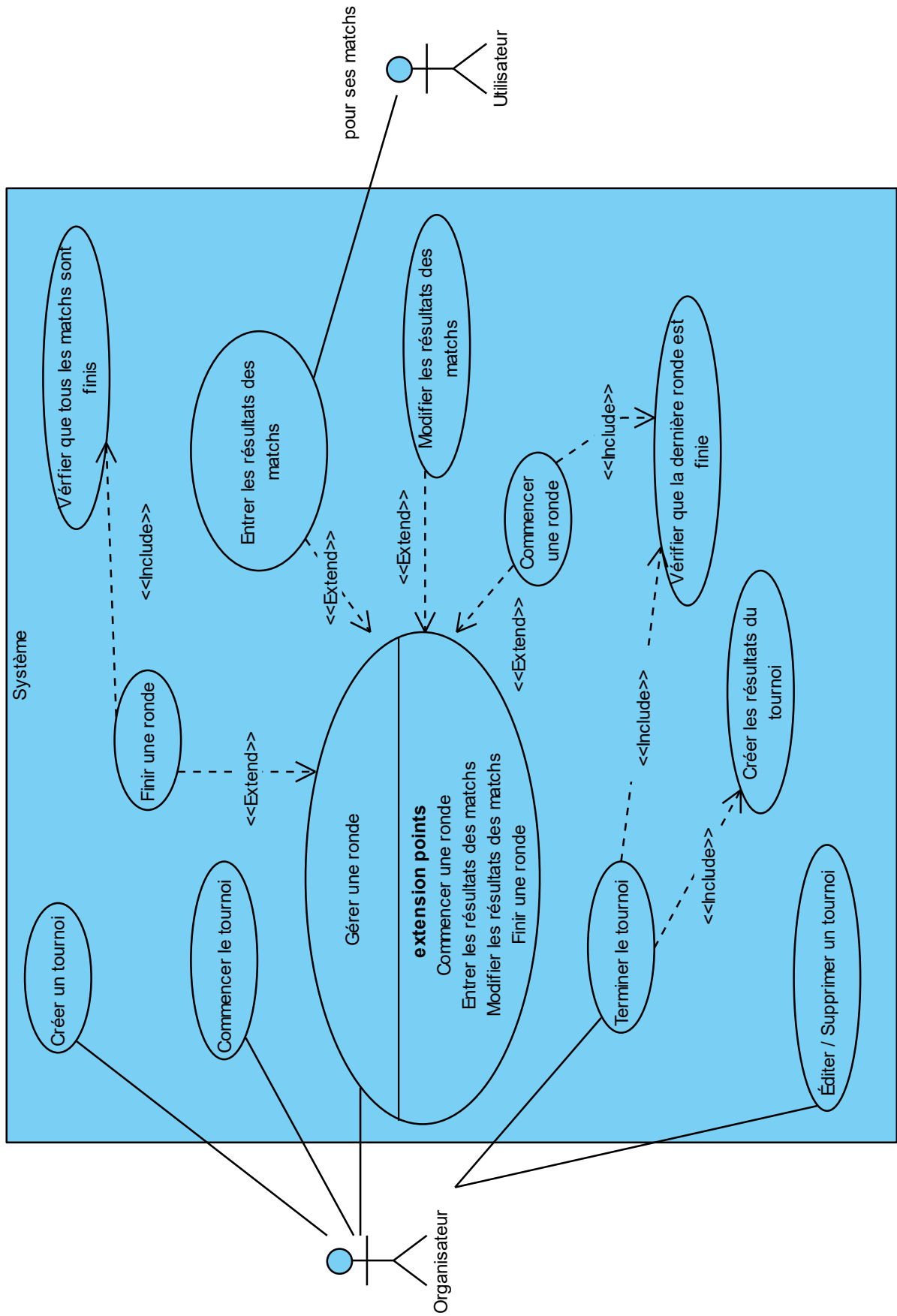


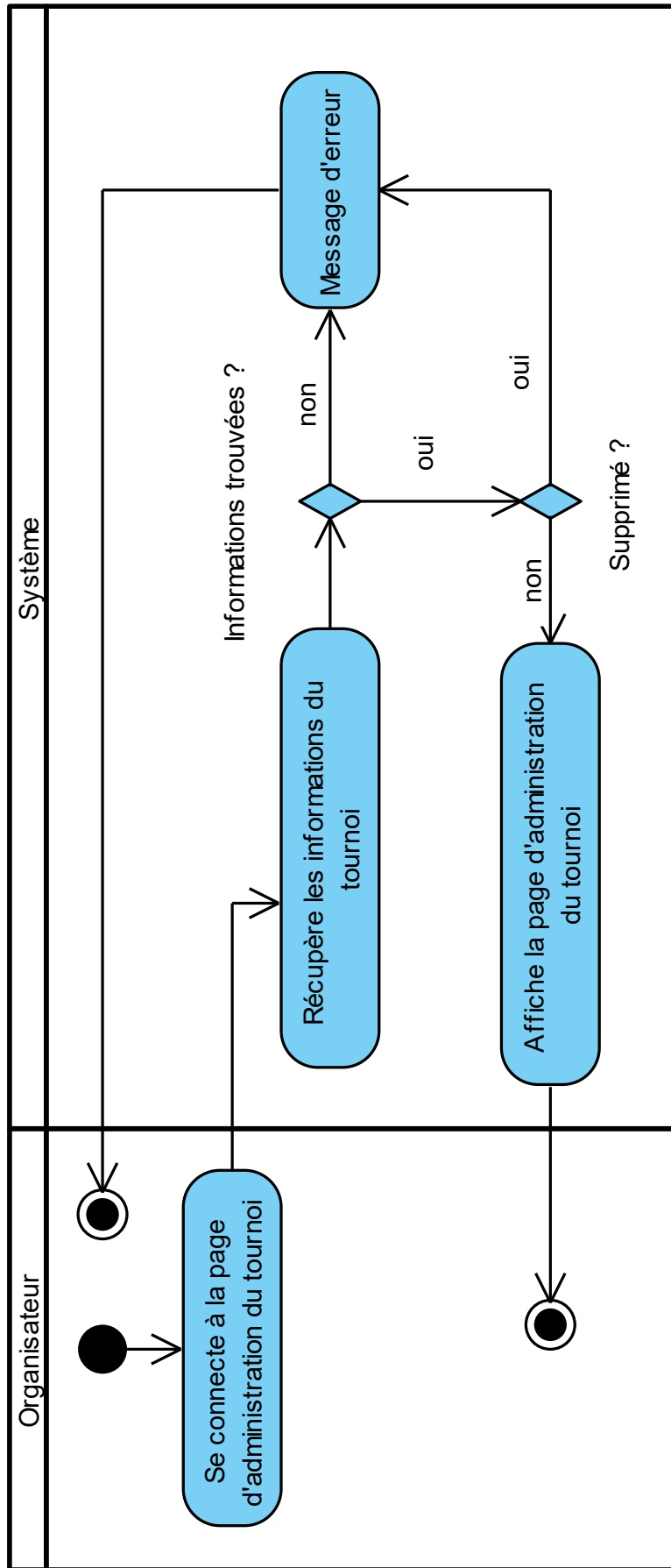


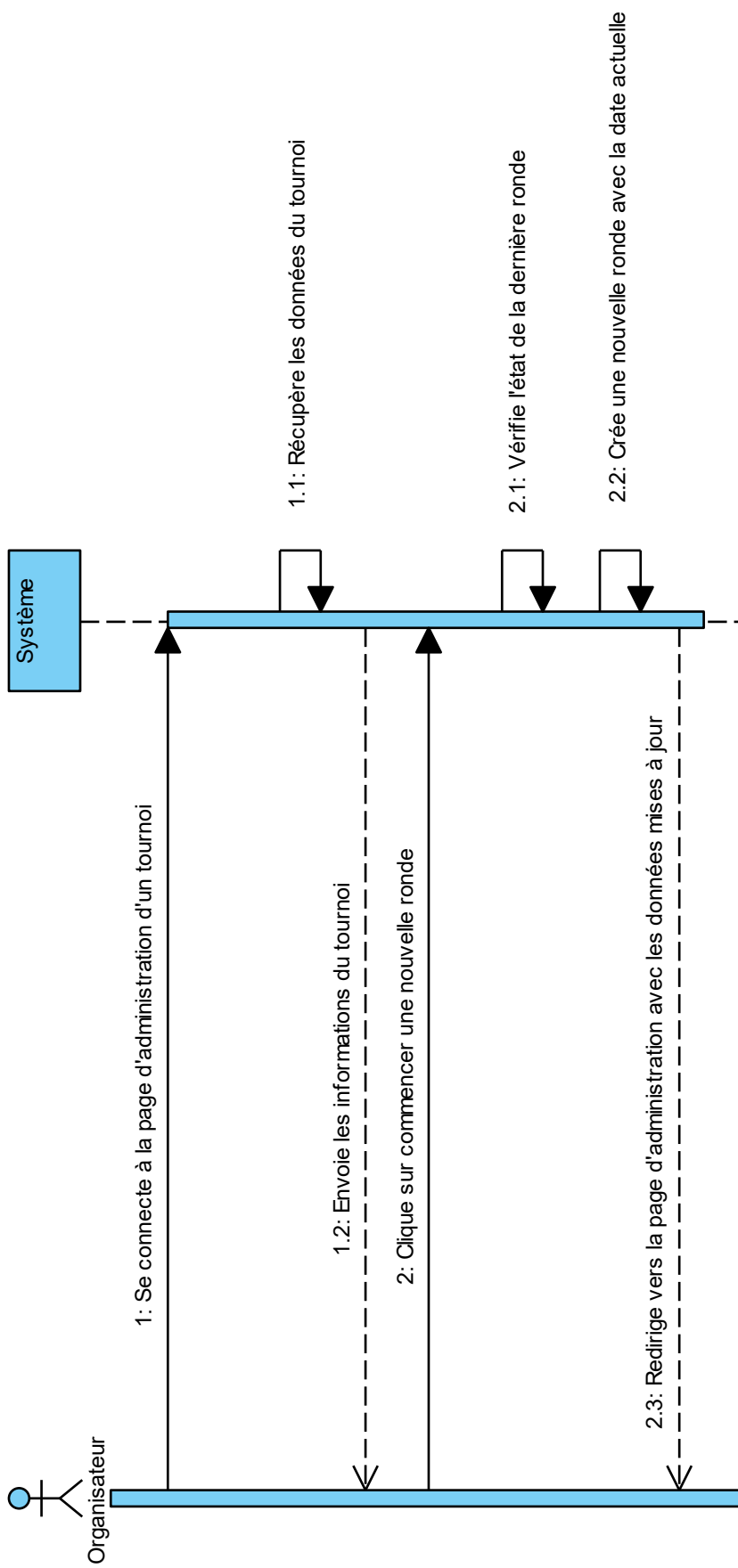


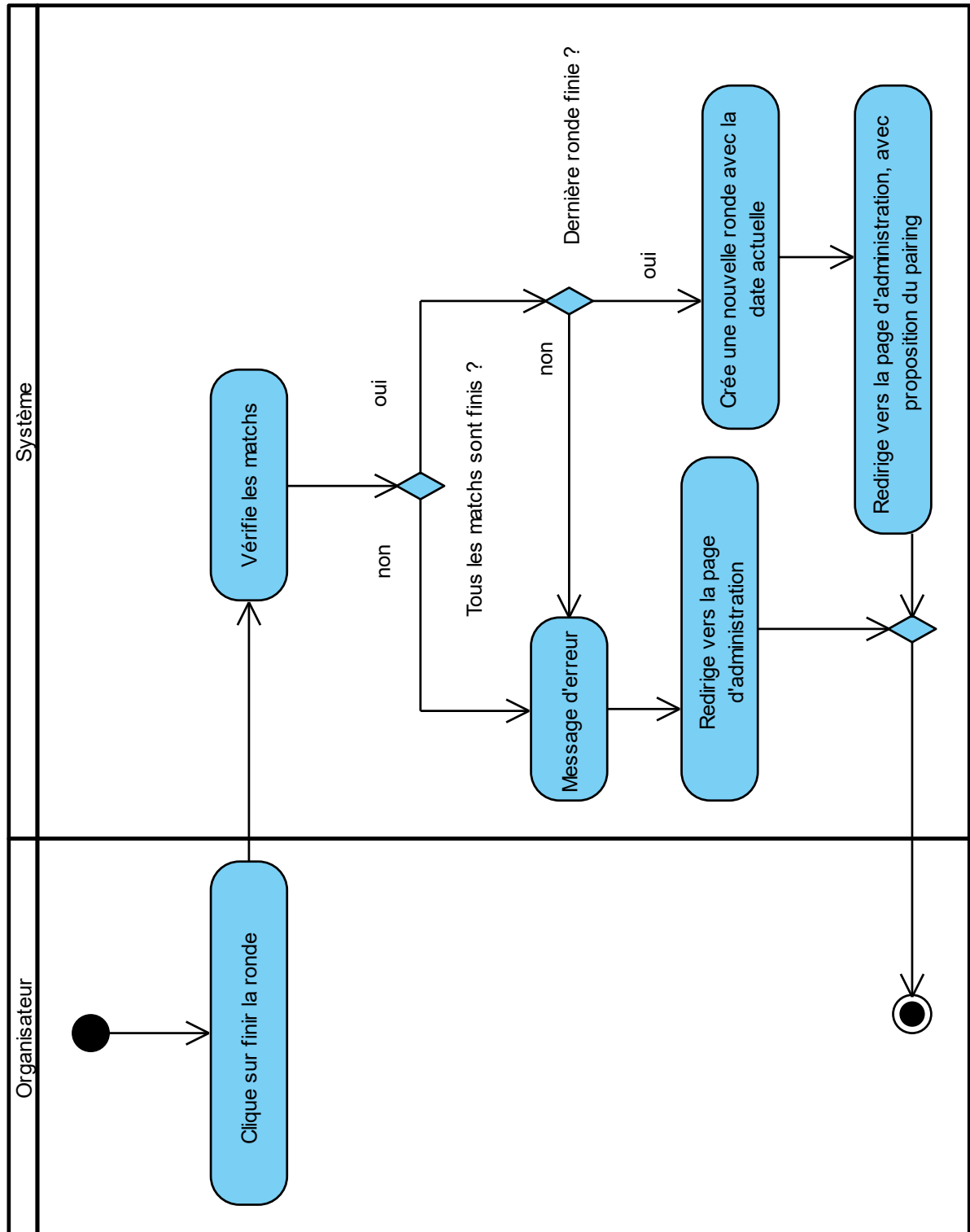




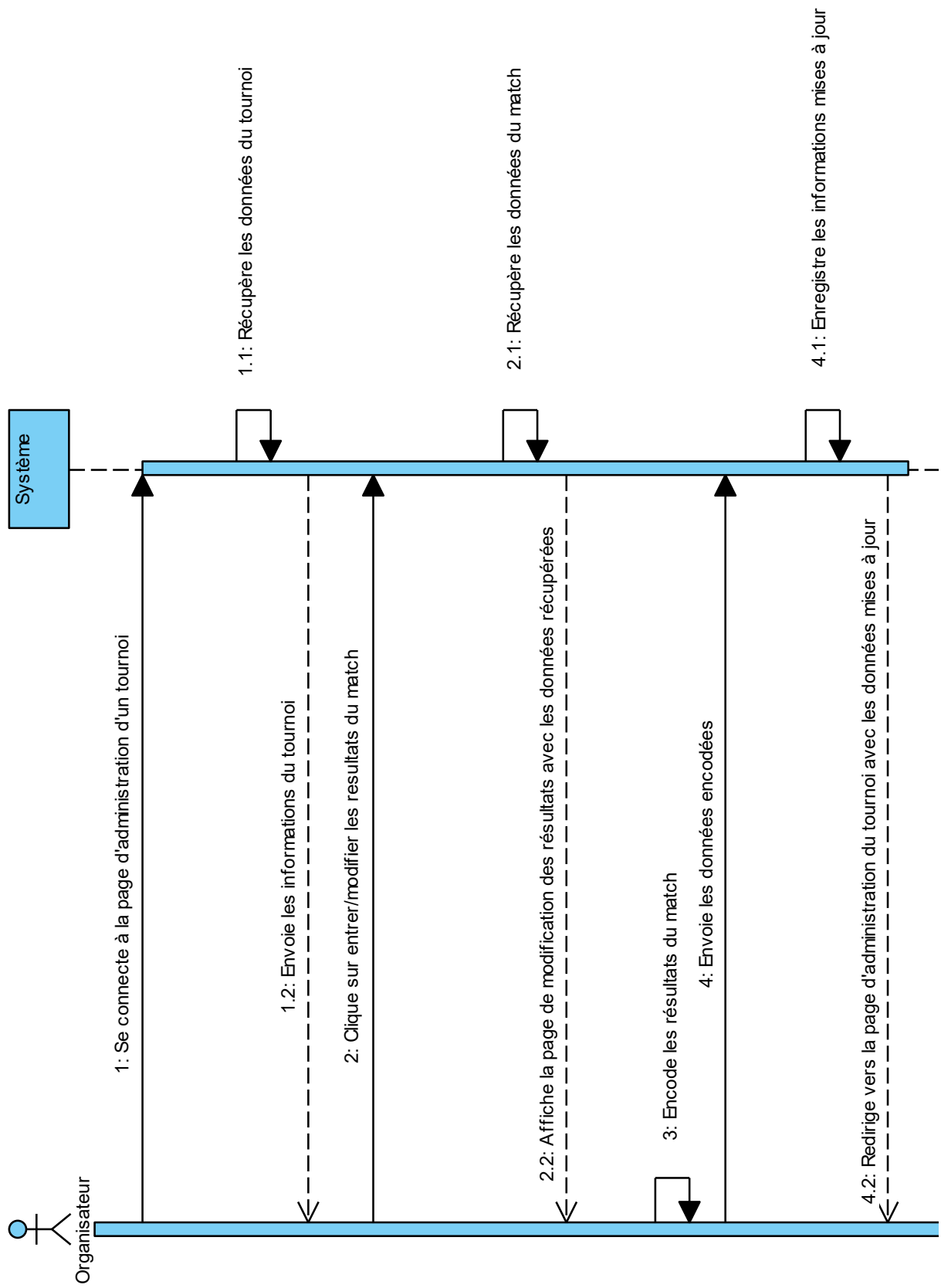


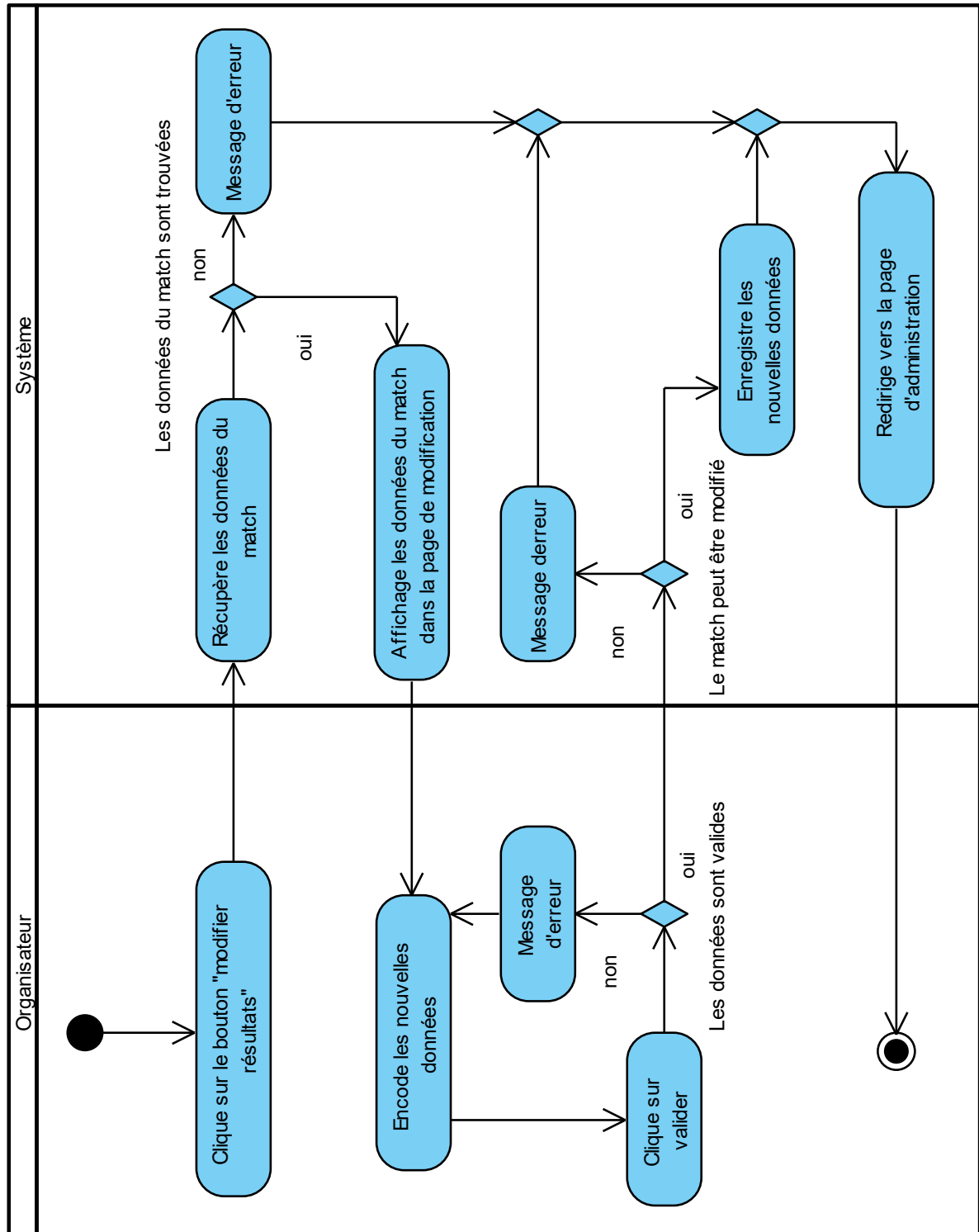


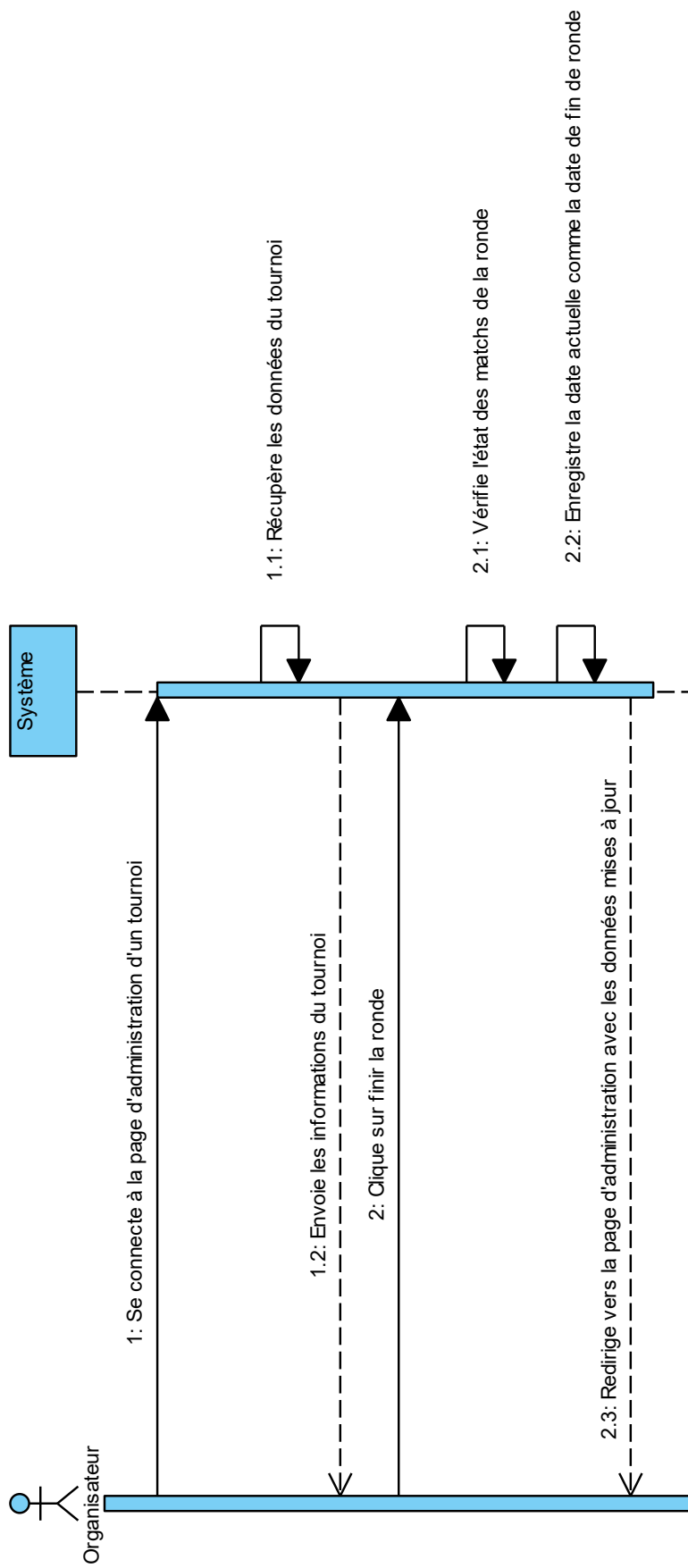


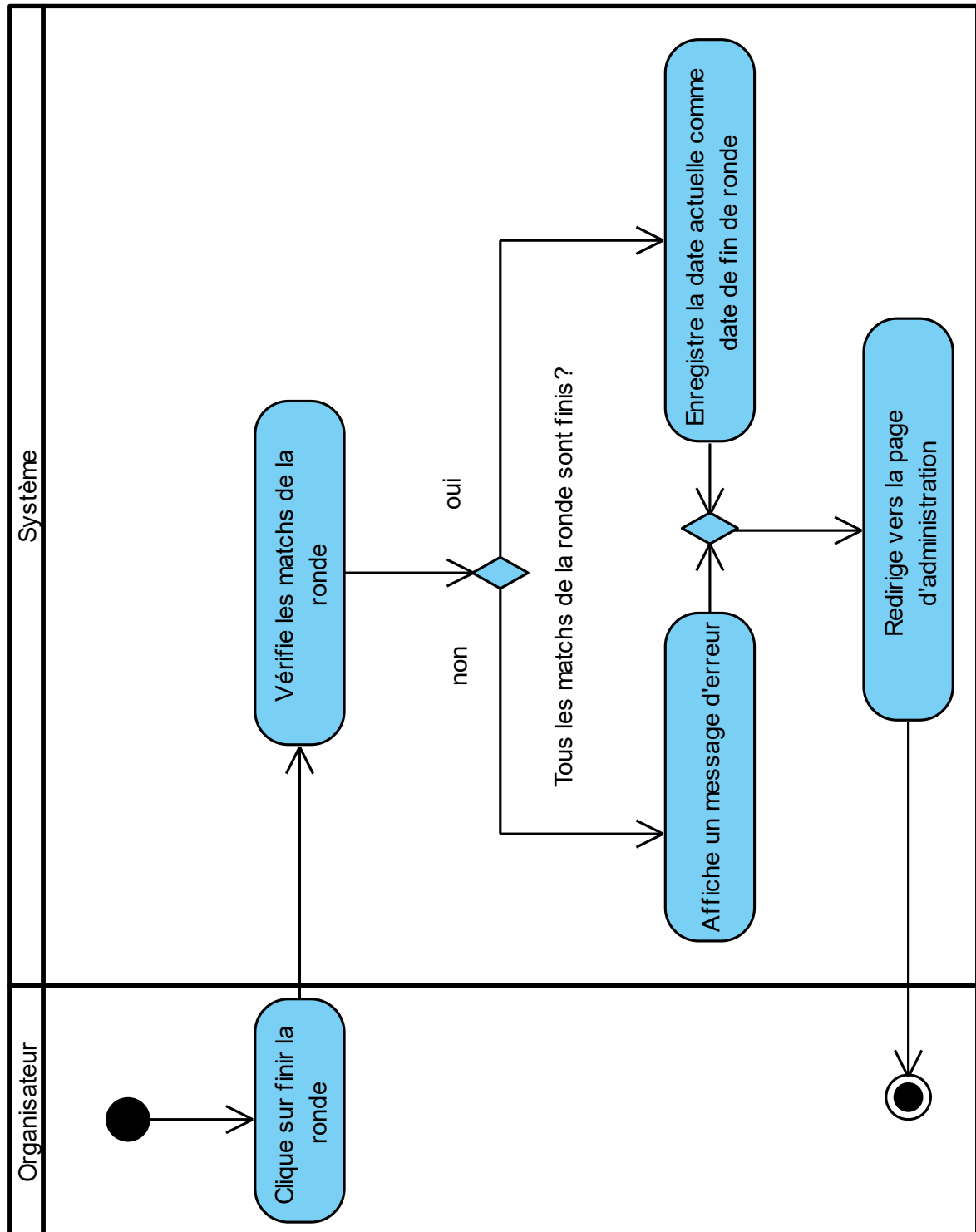




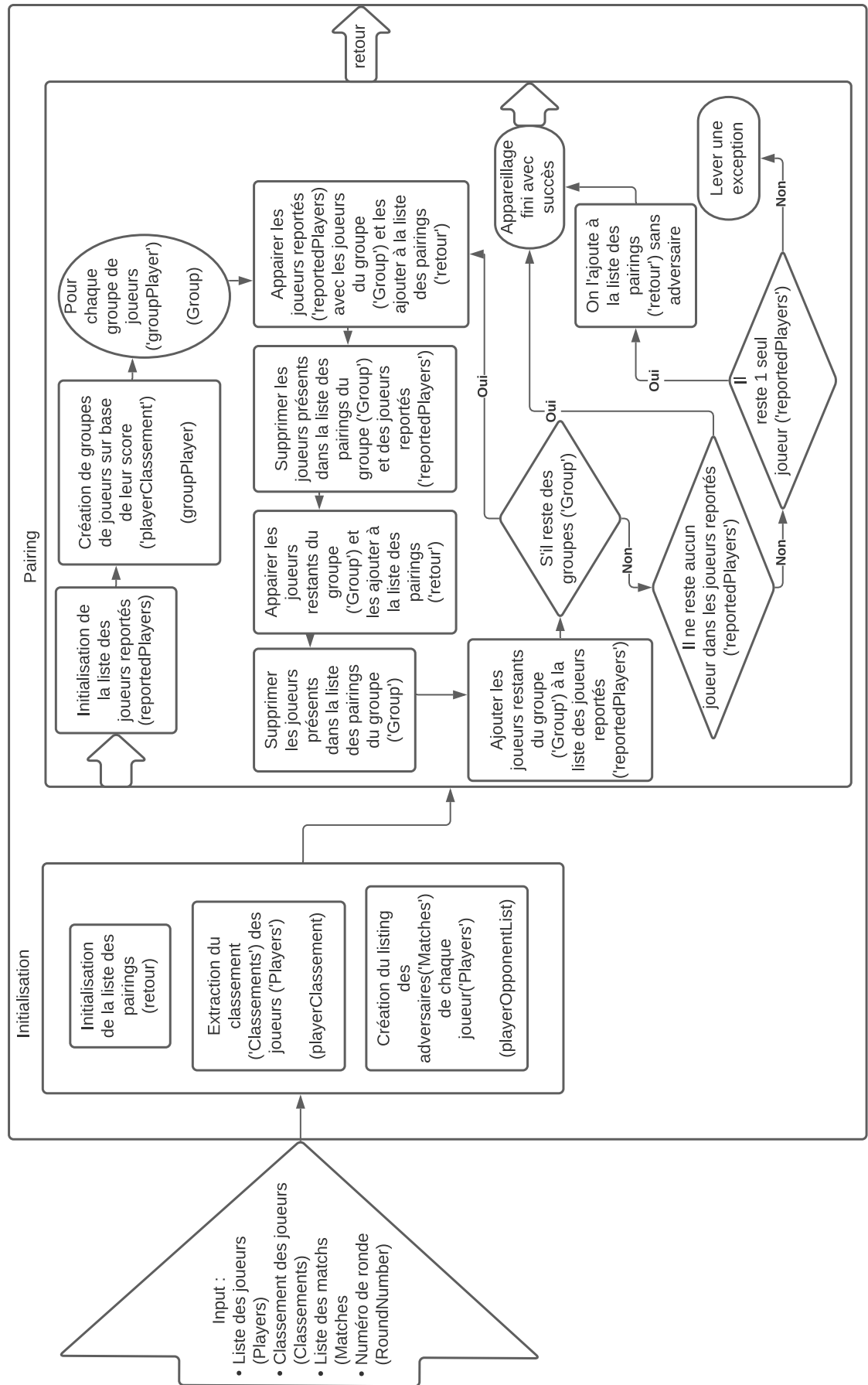


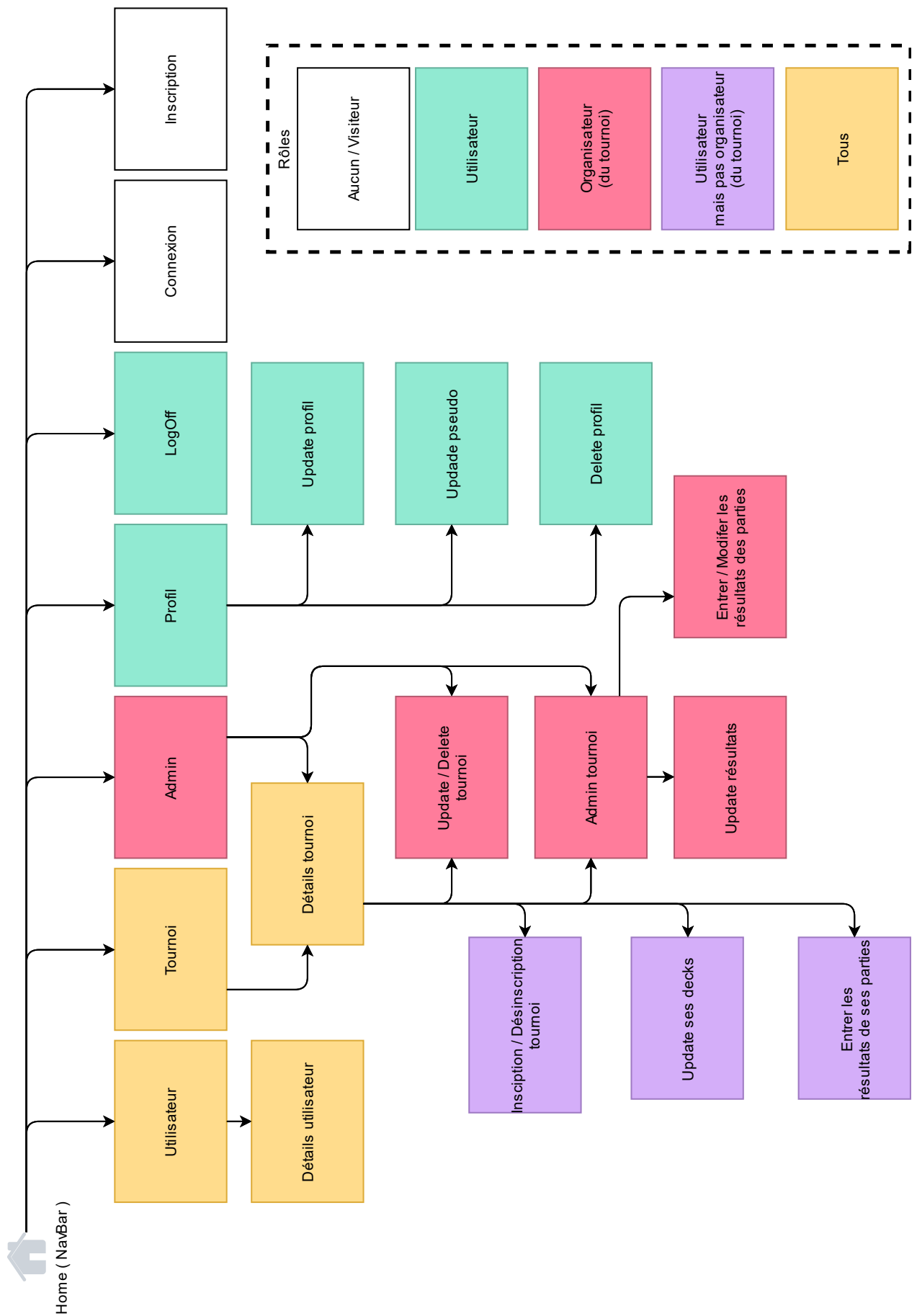


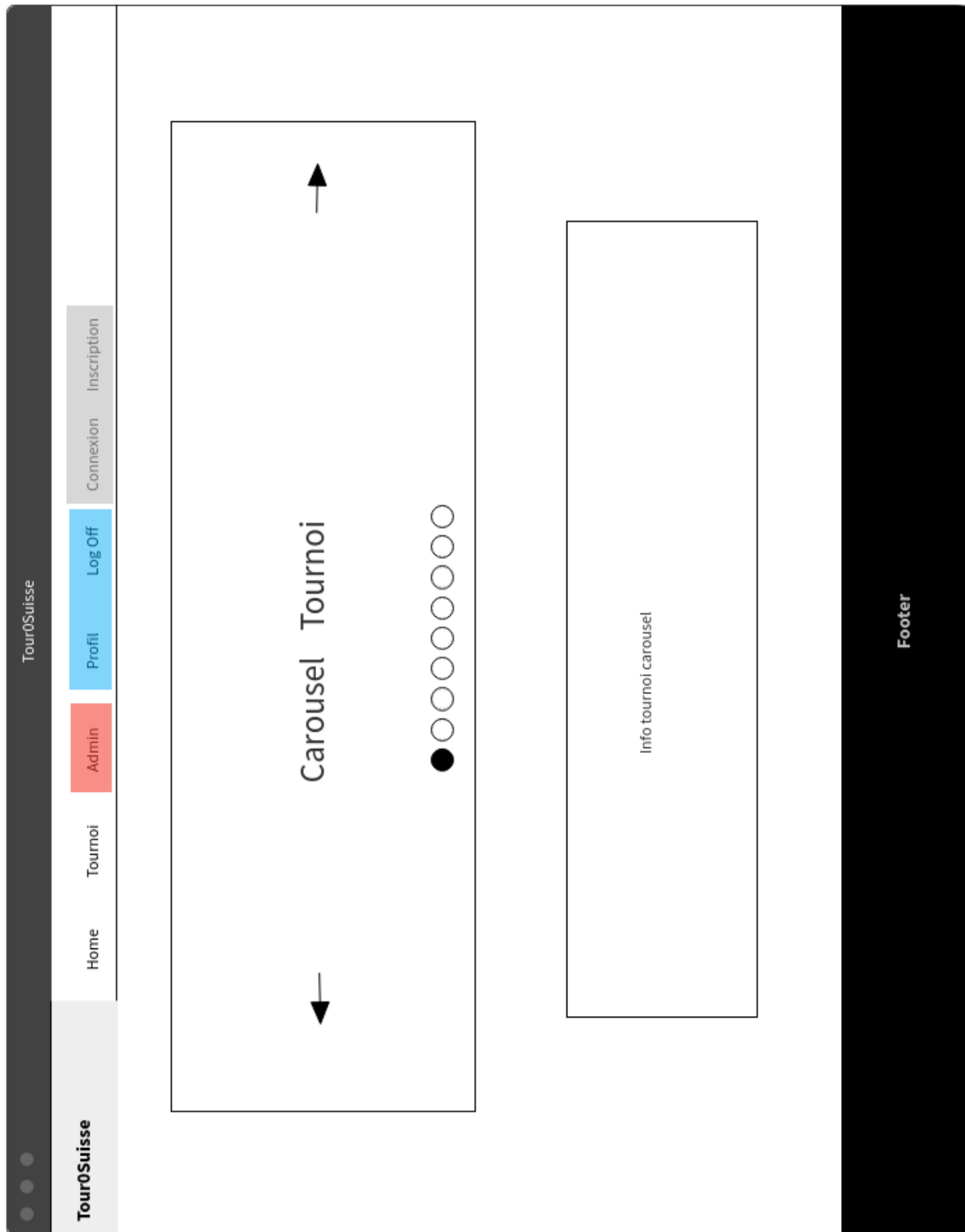




Algorithme globale

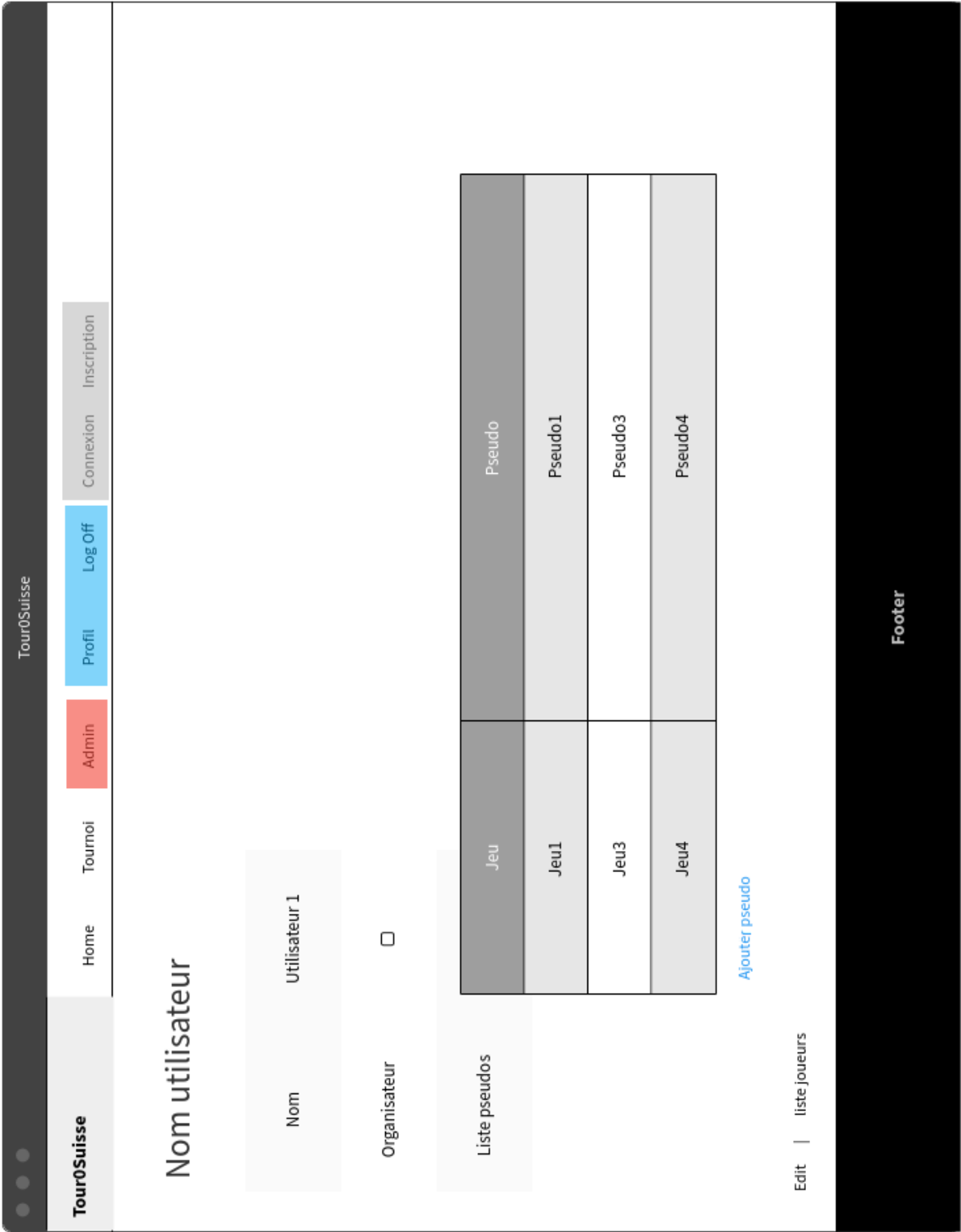












Tour0Suisse

Home

Tournoi

Admin

Profil

Log Off

Connexion

Inscription

Nom du tournoi	Date	Nombre de decks	Jeu
tournoi 1	15/07/2021 20:00	3	jeu 1

ronde 1

ronde 2

ronde 3

ronde 4

joueur1		joueur2	
Janice Monahan		Koch and Sons	modifier
Rollin Fadel		<b>Steuber LLC</b>	modifier
<b>Lera Stroman</b>		Konopelski Group	modifier
Adan Schiller		<b>Harber Inc</b>	modifier

Créer la prochaine ronde

Terminer le tournoi

retour detail | liste

Footer

