

Description de l'étude de cas « Open de Tennis de Lyon »



Fondé en 1987, l'Open de Tennis de Lyon fait partie des tournois internationaux comptant pour le classement ATP 250. En 2017, il devient l'Open Parc Auvergne-Rhône-Alpes Lyon (ou Open Parc ARAL) et est organisé sur terre battue au Parc de la Tête d'Or, durant la semaine précédant Roland-Garros. Le gagnant remporte un prix d'un peu plus de 500 000€ (tournois de Master 1000 : plus de 3M€).

Le parc ne disposant pas de courts de tennis, un court central de 4500 places est construit sur le Vélodrome Georges-Préveral, tandis que 3 courts annexes sont installés sur les parkings situés à proximité. Il existe différents types de billets : grand public, licenciés, promotions (associations...), invitations, et billets pour la Finale.



Un village de 2000m² est associé au Grand Prix, où se trouvent des restaurants gastronomiques et des stands liés au monde du tennis, ainsi qu'un village VIP.

L'Open comprend deux tournois : le **Simple** Messieurs et le **Double** Messieurs. Les qualifications (en Simple) permettent de qualifier 4 joueurs qui pourront participer au tournoi même sans avoir le classement requis. Elles ont lieu les 2 premiers jours, sur 2 tours (*16^e de finales + 1/8^e de finales donnant 4 qualifiés*). Puis le tournoi officiel démarre sur 8 jours. Durant les jours de qualification et de tournoi, un quotidien et un programme officiel relatent la vie de l'Open Park.

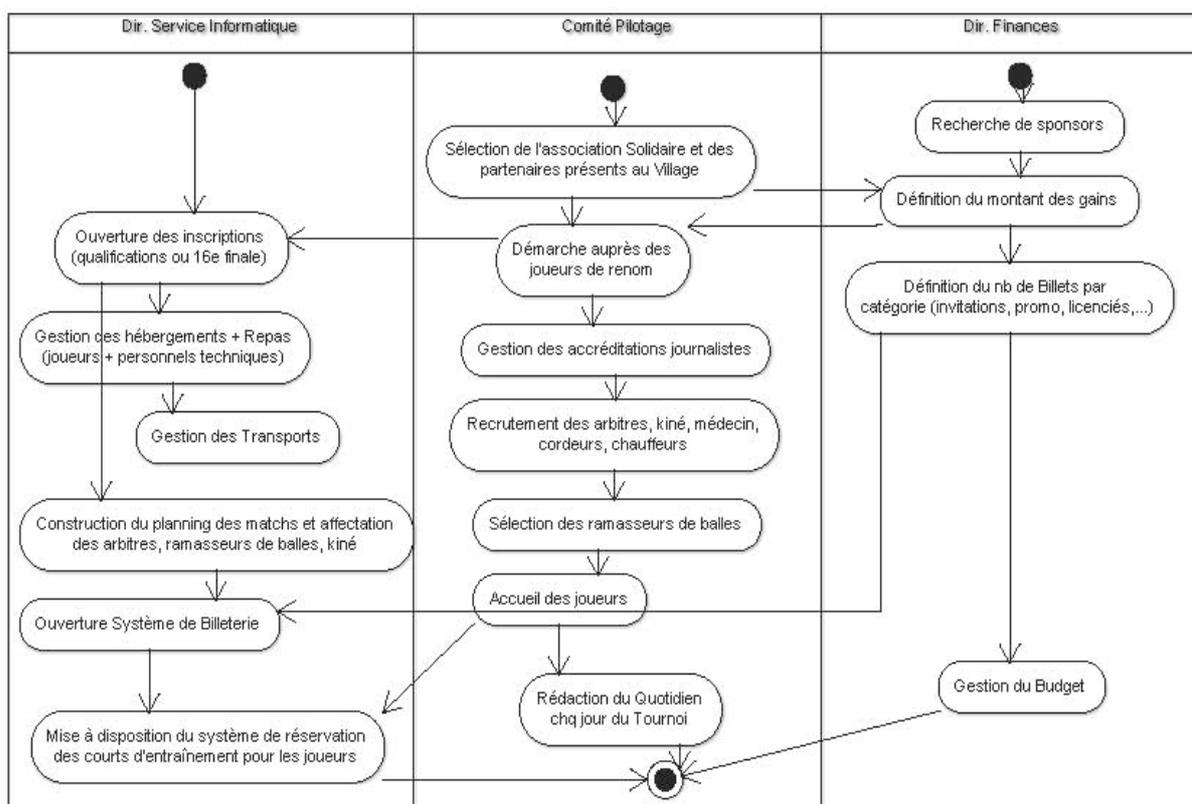
Les matchs à organiser sont ceux de la Qualification (16 joueurs tirés au sort parmi les inscrits pour 8 matchs¹), puis en parallèle les matchs du tournoi Simple et du Double. Le Simple démarre avec les

¹ Equivalent aux '8^{ème} de finales', en qualif, on arrête les matchs aux '1/4 de finale' (4 joueurs sélectionnés)

16e de finale (donc 32 joueurs en Simple), les 8e de finale etc., jusqu'à la finale, appelée *the Big Match*. Le tournoi Double démarre avec les 8e de finales (donc 32 joueurs en Double). Un même joueur peut jouer à la fois en Simple et en Double.

Le tournoi est constitué de 3 principaux comités : un Comité de Pilotage, un service Informatique chargé du site du Grand Prix et des applications nécessaires à la gestion du tournoi, et d'un Comité de Finances. Leurs tâches respectives se déroulent 6 mois avant l'Open et sont décrites dans le diagramme d'activités ci-dessous. Il y a d'autre part un comité OTG (*on the ground*), présent sur le tournoi chaque jour de match, composé de 5 interlocuteurs (superviseur technique -balles, filets, cordage-, service Joueurs, Communication, Marketing et Kinésithérapeute), qui font l'interface avec les comités précédents.

Diagramme d'activité général de l'Open Parc



L'ARBITRAGE

Pour chaque tournoi, il y a un juge-arbitre et 30 arbitres de différentes catégories. Par match, il y a un arbitre « de chaise » (qui est par ailleurs responsable des juges de lignes), et 8 juges de lignes. Sont affectées d'autre part deux équipes de 6 ramasseurs de balles par match.

JOURNALISTES

Des accréditations de 3 niveaux sont distribuées aux journalistes pour la durée du tournoi, qui leur permettent l'accès aux gradins du court pour tous les matchs (niveau 1), de prendre des photos en bord de court (niveau 2), d'accéder à l'espace VIP (niveau 3). L'attribution des niveaux d'accréditation est décidée par le Comité de Pilotage, en fonction du média dont dépend le journaliste et du nombre d'accréditations demandées. On appelle VIP les journalistes ou personnalités du tennis qui sont invitées ou pour lesquelles l'Open Parc offre des services particuliers (notamment l'hébergement). C'est le Comité de Pilotage qui gère les VIP.

SPONSORS

L'Open de Lyon est un évènement fortement médiatisé (TLM, l'Equipe, etc.). Les appels et offres de parrainage sont gérés par le Comité de Pilotage. Le budget global du tournoi s'élevait à **3 millions**

d'euros en 2019. La Région participe à hauteur de 15-20%, 12% provient de la billetterie, et les 70% restants émanent de partenaires privés.

Pour le module CPOA, nous nous limitons à la description simplifiée des modules *gestion des VIP*, *gestion des hébergements*, le *système de Billetterie* et la définition du *planning des matchs*.

Gestion des VIP

Il s'agit ici d'une application qui permettra, entre autres :

- La constitution et le maintien de **fiches VIP** : ceci inclut la description du VIP (on appelle VIP les personnes prises en charge par le tournoi : joueur, accompagnant de joueur, coach technique, manager, etc.) avec leurs caractéristiques : photo, nationalité, classement ATP s'il s'agit d'un joueur, nb de tournois de grands chelems (gagnés, finaliste), ce qu'on lui paye (par ex. considérer des *niveaux* de prise en charge : totale, transport seulement, repas + hébergement, hébergement seulement, etc.), un compagnon attiré, etc., *à vous d'imaginer toutes les informations qui peuvent intéresser les organisateurs du tournoi*.
- La **gestion de la relation VIP** : comme pour l'entreprise avec la CRM (*Customer Relation Management*, Gestion Relation Client), le festival a un souci de qualité et garde donc une trace de tous les *échanges* effectués avec ses invités (courrier, appel téléphonique, FAX) ainsi que toutes les *actions* entreprises pour répondre à leurs attentes (par ex. arriver en Porsche) surtout avant, mais aussi pendant le tournoi.

Ce module sera proposé via une application WEB, accessible depuis n'importe quel poste relié à internet (après authentification du responsable).

Gestion des hébergements

Chaque année le tournoi sélectionne des établissements hôteliers pour les joueurs, la presse, les partenaires, et autres personnes liées au tournoi. Les joueurs peuvent faire eux-mêmes leur réservation sur la centrale de réservation : le tournoi s'occupe de tout (libre à eux évidemment d'opter pour d'autres solutions). L'application Hébergement permettra donc entre autres :

- La **saisie des caractéristiques** de l'hébergement par le gérant de l'hébergement : le nom de l'hôtel ; la localisation ; le nombre d'étoiles..., ainsi que la description des services offerts (bar, restaurant, petit-déjeuner, sauna, salle de sport, coiffeur, pressing, hammam, etc.). On souhaiterait aussi connaître le nombre de places (chambres/lits) disponibles chaque jour du tournoi.
- La **saisie d'une réservation de logement VIP** dans un établissement par un joueur : nombre de personnes, dates et vérification des contraintes. Ainsi dans un souci d'éthique, arbitres et joueurs ne seront pas logés dans le même établissement. On loge généralement l'équipe accompagnant le joueur dans le même hôtel.
- La validation des demandes de réservations par le responsable Hébergement du Tournoi : il valide les réservations et les transfère aux hôtels partenaires. En cas de problème (indisponibilité, conflit éthique, problème de place, etc.), il gère le problème directement avec le joueur et annule la réservation.

On gère l'hébergement uniquement *pour les VIP* du Tournoi. Les personnes *non invitées* se logent par leurs propres moyens. La réservation d'hôtel s'effectue surtout avant le tournoi, une fois le planning des matchs défini. Seuls quelques besoins supplémentaires de logement peuvent arriver pendant le tournoi.

Là aussi, l'application sera de type WEB, accessible directement par les responsables des hébergements afin notamment qu'ils mettent à jour la disponibilité (nb de chambres/lits) pour les jours à venir. La création d'un hébergement (et du compte associé) est sous la responsabilité du responsable Hébergement de l'équipe du Tournoi, de même que la validation des réservations des VIP.

Systeme de Billetterie²

Ce module devra permettre aux internautes d'acheter les billets sur Internet en fonction du nombre de places affectées à chaque catégorie par le Comité de Pilotage. Un billet correspond à une **journée**. Le prix est dépend du type de tournoi (qualification, Simple ou Double) et de la date dans le tournoi (1^{er} tour, demi-finales, finales). Il existe 5 types de billets différents :

- *Billets Grand Public* : (hors finales) le prix de ces places est fonction des jours et des catégories de places (cf. schéma Court Central ci-dessous).
- *Billets Licenciés* : réservée à tout possesseur d'une licence de Tennis. Le joueur doit saisir son numéro de Licence pour acheter le billet. On suppose disposer du fichier national des licences pour vérification. Là aussi, la réduction appliquée est un % défini par le Comité de Pilotage (17% en 2019). Cela concerne près de la moitié des ventes de billets.
- *Billets Journée de la Solidarité* : chaque année une association différente bénéficie de la vente de ces billets à des tarifs préférentiels. Un code promotionnel devra être saisi. C'est le comité de Pilotage qui choisit l'association bénéficiaire de l'année, et les billets prennent le nom de l'association (ex. : Billets RêvesDeClown).
- *Billets « the Big Match »* : ce sont les places pour les 2 finales et leur prix est aussi fonction des emplacements.

On ne gèrera pas le paiement (juste prévoir une page « Vous allez être redirigé vers le site de paiement ») mais on pensera aux **pages de back-office** permettant de définir le nombre affecté à chaque catégorie de billets, et le prix de chacun.

Chaque année et selon le budget obtenu par les sponsors, l'équipe du tournoi définit : le taux de réduction appliqué, le nombre de billets affectés à chaque code promotionnel, et bien entendu le prix des billets (finales incluses).



Les places des matchs du Court Central ont un tarif qui est fonction de leur emplacement (cf schéma ci-contre).

Un billet acheté pour un match du court central permet au spectateur, toute la journée concernée, d'assister également à tous les autres matchs (selon l'affluence : accès géré par un vigile sur place).

Par contre un billet acheté pour un match d'un court annexe ne donne accès à aucune place attitrée du court central.

² Avec l'Open Parc Auvergne-Rhône-Alpes Lyon, la billetterie est maintenant différente. Vous suivrez les indications données ici.

Tarifs des places (central) : il existe 3 catégories de places (loges : bleu, cat.1 : rouge, cat.2 : gris).

	Cat. 1	Cat. 2
Dimanche	30,00 €	25,00 € (-12 ans : 20,00 €)
Lundi	30,00 €	25,00 € (-12 ans : 20,00 €)
Mardi	30,00 €	25,00 € (-12 ans : 20,00 €)
Mercredi	40,00 €	30,00 € (-12 ans : 25,00 €)
Jeudi	45,00 €	35,00 € (-12 ans : 30,00 €)
Vendredi	60,00 €	48,00 € (-12 ans : 38,00 €)
Samedi	60,00 €	48,00 € (-12 ans : 38,00 €)

Planning des Matches

Ce module consiste à **organiser les matchs** de qualification et des 2 tournois. Il s'agit d'abord de répartir les arbitres et les ramasseurs de balles sur les matchs, d'affecter les courts pour les matchs et de gérer les réservations de créneaux d'entraînements. On veillera ensuite, lorsqu'on affecte les joueurs aux matchs, à faire preuve de bon sens : un joueur qui joue à la fois en Simple et en Double ne doit pas être planifié sur 2 matchs qui ont lieu en même temps. D'autres contraintes existent qui sont détaillées maintenant.

Compte tenu des 8 jours de matchs (du dimanche au dimanche avec 2 jours de qualification), il faut placer 8 arbitres de lignes et l'arbitre de chaise de chaque match, en fonction de leurs compétences respectives, ainsi que les équipes de ramasseurs de balles. On affecte en général 2 équipes de 6 ramasseurs par match. Un arbitre de chaise est un arbitre de catégorie ITT1, les arbitres de ligne sont des arbitres de catégorie minimale JAT2. **Pour des raisons d'équité, un même arbitre de chaise ne doit pas juger plus de 4 matchs sur la durée du tournoi (2 en Simple et 2 en Double).**

D'autre part, l'arbitre de chaise doit être d'une **nationalité différente** de celle des joueurs.

Le tournoi dispose d'un *court central* où ont lieu les matchs de Simple et la finale Double, et de 3 *courts annexe* pour les matchs de Double (et quelques Simples).

La réservation des courts annexes pour les entraînements sera disponible pour les joueurs pendant tout le tournoi quand ils sont libres (un seul nom suffit pour la réservation).

Hormis les finales, les matchs de qualification et de tournoi sont planifiés sur 4 tranches horaires : matinée (10h), midi (12h), après-midi (14h), soirée (16h). Les finales Simple et Double ont lieu le Dimanche (cf programme 2020 en Annexe).

Assistance au Planning

Pour le Planning des Matches, l'application devra **aider l'organisateur à placer un match** sur un créneau donné, en vérifiant les contraintes. Les contraintes relèvent du type de match : match de tournoi ou finale, et pour les matchs de tournoi : type de tournoi (Qualifications, Simple ou Double). Elles concernent aussi les nationalités et catégories d'arbitres, les équipes de ramasseurs de balles, la date et le créneau horaire, etc. Pour la réservation d'un court d'entraînement par un joueur, il suffit d'afficher de façon claire les courts disponibles, et de saisir le nom du joueur sur l'horaire choisi.

NOTA : le tableau final du tournoi Simple comprend 32 entrées (1^{er} tour = 16^{es} de finale) : au min 22 entrées directes, au max 3 *wild Cards* (invitations), 4 qualifiés et 2 *special Exempts* (joueurs admis directement en dépit de leur classement insuffisant). Pour la qualification, on a 16 joueurs, tirés au hasard parmi les joueurs du top 200 du classement ATP qui souhaitent venir. Pour le tournoi Double, on a aussi besoin de 32 joueurs (on démarre aux 1/8^e de finale : 8 matchs x 4 joueurs).

Une solution pour élaborer le planning serait par exemple de procéder en 2 temps :

(1) Créer un 1^{er} planning des matchs d'un type donné (Simple Messieurs, Double Messieurs, Qualifications) à partir d'une liste des joueurs qu'on imagine fournie (à vous de créer une liste test), des courts et des tranches horaires proposées. Cela signifie **insérer un match** avec les joueurs, les arbitres et les 2 équipes de ramasseurs, sur un court, avec un horaire donné, en

respectant les contraintes mentionnées. Prévoir aussi de mentionner le **nom des gagnants** une fois le match terminé (au meilleur des 3 manches), ainsi que le **score**.

(2) Ensuite le module doit permettre au responsable du Planning de **modifier, supprimer** ou **déplacer** un match du planning existant sur un autre créneau (parmi ceux possibles), à sa convenance. On peut aussi imaginer que des déplacements sont nécessaires par exemple quand un match dure trop longtemps en soirée, ou en cas de pluie.

On aura noté que ce module comprend de fait 2 parties distinctes : l'une à destination du **responsable des Planning** pour la définition du planning initial des match, et une autre à destination des **joueurs**, pour ce qui concerne la réservation d'un court pour un entraînement à un horaire donné pendant le tournoi. Ces 2 applications seront développées en **Java**.

Travail demandé

On vous demande de réaliser, en trinôme, l'analyse et l'implémentation de l'application OPEN PARC de Lyon avec au minimum 2 modules :

- au choix, **un module WEB** pour la *Gestion des VIP*, la *Gestion de l'hébergement* ou la *Billetterie* ;
- le **module Java** constitué des 2 applications, pour le *Planning des Matches*.

L'analyse sera effectuée en UML et l'implémentation en Java avec un SGBD relationnel pour les données persistantes, en PHP/MySQL pour les modules WEB. On veillera à bien scinder les 3 couches : *Présentation*, *Métier* et *Persistence* des données, de manière à permettre un changement d'interface ou de BD, sans avoir à modifier tout l'ensemble.

- **Couche Présentation** : elle concerne tout ce qui est affichage permettant les interactions avec les utilisateurs (modules WEB et Java). Attention, **20% de la note de la démonstration concernera l'ergonomie** (les critères seront rappelés en séance).
- **Couche Métier** : cela concerne tous les *processus métiers* à effectuer pour les différents modules : création d'un planning de matchs pour un tournoi donné (ex. Simple Messieurs) en fonction des contraintes mentionnées ; saisie des joueurs/ équipes du match, saisie des scores, insertion d'un match dans un planning en cours, vérification du nombre de billets disponibles à l'achat, basculer vers d'autres types de billets en cas d'indisponibilité, etc.

Au niveau de la modélisation, il faudra décrire toutes les classes participant à l'application.

Théoriquement **le back End devrait aussi être implémenté** dans le logiciel livré. Cela concerne les *gestions des entités Métier* : création / modification / suppression des différentes entités de l'application (joueurs, VIP, hôtels, matchs, etc.). On vous propose de les modéliser en UML, mais c'est inutile de les implémenter (vous utiliserez des données Test saisies dans les tables).

- **Couche Données** : stockage des données persistantes dans des tables relationnelles. On veillera à faire une classe de connexion indépendante avec un fichier de propriétés pour les paramètres de connexion. Certaines tables sont communes aux 2 modules. Il faut donc concevoir une seule BD. On pourra la faire en local pour faciliter le développement (serveur Apache et phpMyAdmin), mais vous devez partager les scripts de création / insertion sur GIT, afin de travailler toujours sur la **même** BD.

IHM

Parallèlement à l'analyse, vous **élaborez les IHM des 2 modules** : après avoir fixé les cas d'utilisation, développez les maquettes d'écran et leur enchaînement illustrant le fonctionnement choisi, à l'aide de l'outil de votre choix (WireFrame ou autres).

Données

Comme précisé précédemment, on supposera que la liste des 25 joueurs sélectionnés pour le tournoi Simple est connue, ainsi que la liste des 16 joueurs passant les qualifications et les 3 invités (*wild card*), et les 32 joueurs (ou moins) de Double. La liste des arbitres sera aussi déjà affectée avant le tournoi, ainsi que les ramasseurs de balles sélectionnés.

Vous pouvez inventer des noms et nationalités pour tous ces éléments.

Il existe un site appelé **Generate Data**, <http://www.generatedata.com/#generator> qui permet de créer un script SQL pour remplir automatiquement vos tables de données fictives (ID, noms, prénoms, classement ATP, etc.). *Attention, ne passez pas du temps à remplir la BD et rédiger les rapports, c'est le reste qui est important pour ce module.*

Cf. <https://openparc.com/> pour un accès aux informations réelles.

Annexes - Sommaire

L'annexe 1 propose un programme prévisionnel pour l'édition 2020 (<https://openparc.com/tournoi/programme-des-matches/>).

L'annexe 2 propose un guide méthodologique qui devrait vous permettre de mieux cadrer le déroulement de votre projet.

En Annexes 3 et 4, on spécifie clairement ce qu'on attend des deux rapports à rendre (en analyse/conception puis en programmation). En Annexe 5, on donne les attentes de la démonstration.

Annexe 1 – Planning prévisionnel des matchs

Dates : du Dimanche au Dimanche

Qualifications (uniquement en Simple)

4 qualifiés → 16 joueurs, 8 matchs au 1^{er} tour, 2 tours uniquement, à planifier sur les 3 premières demi-journées (Dimanche et Lundi matin).

Tournoi Simple

1^{er} tour = 16^e de finales : 16 x 2 joueurs = 32 joueurs, 16 matchs le 1^{er} tour, à planifier dès le Lundi.

Tournoi Double

1^{er} tour = 8^e de finales : 8 x 4 joueurs = 32 joueurs, 8 matchs au 1^{er} tour à planifier à partir du Mardi.

LE PROGRAMME DE LA SEMAINE	
DIM 16 MAI QUALIFICATIONS SIMPLE	À PARTIR DE 10H00 8 MATCHS Téléchargez le programme de dimanche 16 mai
LUNDI 17 MAI QUALIFICATIONS SIMPLE + 1E TOUR SIMPLE	À PARTIR DE 10H00 10 MATCHS Téléchargez le programme de lundi 17 mai
MARDI 18 MAI 1ER TOUR SIMPLE 1/8E DE FINALES DOUBLE	À PARTIR DE 10H30 10 MATCHS Téléchargez le programme de mardi 18 mai
MERCREDI 19 MAI 1ER TOUR SIMPLE 1/8E DE FINALES DOUBLE	À PARTIR DE 10H30 10 MATCHS Téléchargez le programme du mercredi 19 mai

JEUDI 20 MAI 1/8E DE FINALES SIMPLE 1/4 DE FINALE DOUBLE	À PARTIR DE 10H30 9 MATCHS Téléchargez le programme du jeudi 20 mai
VENDREDI 21 MAI 1/4 DE FINALE SIMPLE 1/4 DE FINALE DOUBLE	À PARTIR DE 10H30 5 MATCHS Téléchargez le programme du vendredi 21 mai
SAMEDI 22 MAI DEMI-FINALES SIMPLE DEMI-FINALES DOUBLE	À PARTIR DE 11H00 4 MATCHS Téléchargez le programme du samedi 22 mai
DIMANCHE 23 MAI FINALE SIMPLE FINALE DOUBLE	À PARTIR DE 11H45 2 MATCHS Téléchargez le programme du dimanche 23 mai

ANNEXE 2 – GUIDE METHODOLOGIQUE

- **L'utilisation d'un AGL** est conseillée pour la **génération automatique** de code. L'application demandée comporte différents modules, mais tous relèvent de la même application : ils partagent des données communes (il faut déterminer lesquelles). Ne faites pas de développements indépendants. Notamment sous PowerAMC, les 2 modules doivent **faire partie du même projet**, avec 2 packages.
- Pour les données persistantes, vous choisirez une implémentation par BD avec **JDBC** et les DAO, ou un framework d'accès aux données (tel HIBERNATE) si vous le connaissez. Dans SQL*Plus, vous pouvez valider votre script SQL (notamment chaque script généré par PowerAMC) en l'exécutant avec start. **Rappelez-vous qu'une appli WEB n'est en général pas capable d'accéder à une BD Oracle** (une appli WEB a des contraintes d'ouverture que le SGBD Oracle n'autorise pas).
- Le module a une durée de 44h. En mode traditionnel, **l'analyse requiert environ 12h**, la conception **4h**, la programmation **24h**. En mode Agile, vous définissez la durée de sprint, et les éléments ciblés pour chaque itération. Les dernières 4h sont consacrées à l'évaluation du travail réalisé par chaque trinôme sous la forme de démonstrations, devant le groupe entier.
- Le travail concerne deux parties bien distinctes : l'analyse/conception et l'implémentation de l'application. Vous devrez donc **rendre deux dossiers distincts**, l'un concernant l'analyse/conception, et l'autre la programmation. Ce dernier comportera en outre un bilan et une réflexion *a posteriori* du travail d'analyse effectué, avec mention des *points forts* : qu'est-ce qui a permis d'aller plus vite, de faciliter la construction des classes, etc. et des *points faibles* : erreurs commises, incomplétudes, éléments non utilisés. **L'implémentation pourra être différente de ce que vous aviez envisagé en analyse, mais les écarts devront être argumentés.**
- Comme le temps risque d'être court, il est important de **prioriser ce qui risque d'être difficile**, i.e. ce qui concerne le « métier » (planifier les matchs, affecter les hébergements) plutôt que le back end lié aux « gestions » (gestion des joueurs, des matchs, des arbitres, etc.).
- Pour faciliter le développement à plusieurs, vous utiliserez un **système de gestion des versions** type CVS (*Concurrent Versions System*), par ex. GIT avec le GitLab de l'Université (avec notamment sa gestion de tickets). Le correcteur aura accès à votre dépôt.
- Pour le développement WEB et l'accès aux données, ne sont autorisés que les **Frameworks connus**.
- **Gestion de projet** : le développement de cette application se fera selon une méthode de GP de votre choix : traditionnelle ou agile, dont vous devrez *rendre compte* pour la présentation finale. On veillera à constituer un **planning prévisionnel des tâches** pour la programmation, avec durée et affectation aux membres du binôme. Outre les tâches de codage des différentes parties, vous penserez aux tâches annexes comme la configuration d'un outil collaboratif (ex. CVS), la création des tables, les tests, la préparation de la démonstration, etc. Vous noterez ensuite le temps passé (à la louche) à réaliser chacune des activités pour pouvoir, en fin d'implémentation, **comparer avec un planning réel**.
Pour la conduite en méthode agile, il faut donner précisément la durée des itérations choisie, mais surtout la trace de tous les backlogs et des tâches précises envisagées en début de chaque sprint, afin que l'on puisse évaluer votre façon de gérer le développement agile.
- Fonctionnement en **mode projet** : les encadrants sont là en tant que « **client** » et pour vous aider avec UML et Java (moins en web), pas pour diriger votre travail. Il y a plusieurs façons de voir et de procéder pour l'application proposée, chaque trinôme construira son système selon son point de vue qu'il présentera lors de la démonstration finale.



ANNEXE 3 – ATTENTES DU RAPPORT D'ANALYSE

Le dossier d'analyse/conception est un dossier complet. **Les 2 modules doivent faire partie du même projet** (avec 2 packages). Il doit comprendre tous les éléments explicatifs afin de ressembler à un **rapport professionnel**. L'énoncé sera fourni en format éditable, pour que vous puissiez récupérer des paragraphes. Pour le rapport vous vous aiderez pour cela de la **génération automatique de rapport d'analyse** à partir de l'AGL.

Le rapport comportera 5 éléments :

1. La **présentation de l'application** complétée et présentée selon vos soins : vous détaillerez ce que vous avez compris à partir de l'énoncé et des explications que l'on vous aura fournies en tant que "client".
2. Le diagramme de **Packages** et leur dépendance (quelle classe a besoin de quelle autre classe). Les dépendances ne sont souvent visibles qu'en fin de modélisation.
3. **Par package** :
 - Le **Diagramme des Cas d'Utilisation** (DCU) ;
 - Puis, **par cas d'utilisation** jugé intéressant ou complexe (min 3) :
 - o Un **Diagramme d'Activités** (DIT) illustrant le fonctionnement du Cas ;
 - o Eventuellement quelques **scénarios** associés aux différentes situations (écrits dans les propriétés du Cas dans l'AGL, onglet Spécifications) ;
 - o Un ou plusieurs **Diagrammes de Séquences Objet** (**pas de DSQ système**) qui modélisent certains points spécifiques ;
 - Le **Diagramme de Classes du package** (DCL).
4. La **partie IHM** avec les maquettes d'écran intéressants, et leur enchaînement illustrant le fonctionnement choisi (outil de votre choix : WireFrame ou autre).
5. Enfin pour la **partie conception**, on donnera :
 - les **Diagrammes de Classes de Conception** (enrichis des éléments de conception) ;
 - les **scripts SQL** de création d'au moins 2 tables relationnelles (par ex. issus de PowerAMC) ;
 - le squelette de **2 classes Java** issues de la génération automatique de code.

IMPORTANT : les diagrammes UML doivent être **intégrés** dans le corps du rapport (ne pas les mettre en Annexe) et **complétés d'explications textuelles** : ils participent à la compréhension générale de l'analyse autant que le texte qui est là pour guider la lecture.

Par exemple, le choix des attributs mentionnés dans une classe Tournoi peut être justifié dans le texte : « *L'identifiant de la classe BudgetTournoi est composé d'une clef du tournoi couplée à l'année. En effet chaque année les budgets des tournois sont modifiés et le budget est géré à l'année ; on garde ainsi la trace des différents budgets annuels des tournois.* »

ANNEXE 4 – RAPPORT DE PROGRAMMATION



La programmation donnera lieu à une évaluation qui tient compte de la réalisation (démonstration) et de son rendu via le dossier de programmation. Ce dernier comportera deux volets : un volet Technique et un volet Gestion de projet, en voici le détail.

1. Éléments techniques

- **État d'avancement** : ce qui marche, ce qui reste à améliorer, ce qui n'a pas été commencé.
- **Description de l'architecture en 3 couches** : *présentation* (choix pour les IHM, classes frontières), *métier* (classes d'application, classes techniques) et *données* (argumenter les choix, lister les problèmes éventuellement rencontrés avec les tables).
- **Retour à l'analyse** : l'objectif du module étant de développer une application de synthèse Conception & Programmation, il est important de faire ce bilan une fois le codage terminé. Vous donnerez donc les éléments qui vous permettent d'en évaluer les **avantages** et les limites (ce qui vous a beaucoup servi, détails envisagés en analyse qui s'avèrent inutiles ou irréalisables lors du codage, contraintes de codage qui conditionnent l'analyse, modifications en cours de codage et justifications...).
- **« Si c'était à refaire »** : vous analyserez ici ce que vous feriez pour cette application s'il fallait la refaire, compte tenu de votre expérience.
- On pourra aussi faire une **rétro-conception** du code développé et comparer avec le modèle d'analyse qui avait été proposé.

2- Éléments de gestion de projet

- Donner les **2 plannings des tâches** (prévisionnel et réel) et les comparer. Ce travail devrait vous permettre de mieux chiffrer les durées des tâches pour vos futurs projets ;
- Préciser aussi le **partage des tâches** au sein du binôme (cela peut être mentionné dans le planning réel) ;
- Rendre compte de **l'utilisation du système de gestion des versions** (le correcteur devra avoir un accès à votre dépôt) ;
- Si vous avez pu tester, expliquer le **protocole de tests effectués** pour valider l'application (ex. billetterie), avec l'utilisation par ex. de JUnit sous NetBeans (jeux de tests).
- Donner un **bilan personnel** de ce module : ce que vous avez appris, les difficultés rencontrées, etc.

ANNEXE 5 – ATTENTES DE LA DEMONSTRATION

Pour la démonstration, il faut présenter les points suivants :

- (1) **Avancement** de l'implémentation : ce qui est fait, ce qui ne fonctionne pas encore.
- (2) **Répartition des tâches** dans le trinôme, et temps moyen requis pour chacune.
- (3) **Comment vous avez conçu l'application** (démonstration expliquée).

Pour cela, on reviendra aux schémas d'analyse pour illustrer l'implémentation. Par exemple :

"Pour le Planning des matchs, on a utilisé une classe X qui détaille l'horaire, le court, etc"... on montre le *diagramme de classes* avec tous ces éléments,

"et voici comment ça fonctionne" : démonstration d'une création de match, avec les arbitres, le court, les ramasseurs et les joueurs. On montre aussi le *diagramme d'activité* qui expliquait ce fonctionnement ;

"Le fonctionnement du module Hébergement est le suivant" : diagramme des cas d'utilisation pour montrer les utilisateurs/rôles, diagramme d'activités du module, puis démonstration.

(4) Aspect **Gestion de Projet** : méthode utilisée (ex. : développement itératif incrémental d'UML, développement agile avec priorisation des fonctionnalités et développement d'incréments de produit, etc.), comment vous avez organisé le projet (couches MVC, etc.), quelle utilisation du CVS a été particulièrement bénéfique, comment vous avez collaboré au sein du trinôme, comment vous avez testé les livrables, le niveau de commentaires du code, etc.

(5) **Retour sur l'analyse** : "tel et tel diagramme nous ont bien aidé pour ..." ..., "là on s'est trompé car ..." "si c'était à refaire, voilà comment on s'y serait pris...". **Bilan du module.**

80% de l'évaluation concerne tous ces points, **les plus importants étant les points (3) et (5).**

20% de la note de démo se rapporte à l'ergonomie (critères rappelés en séance).

Durée - Vous avez 30' par trinôme. On peut demander à voir le code et poser des questions techniques, tous les membres du trinôme s'expriment. La note de la soutenance (et du module) peut être individualisée si on constate une forte différence d'implication.