

Description de l'étude de cas « Grand Prix de Tennis de Lyon »



Fondé en 1987, le Grand Prix de Tennis de Lyon (GPTL) fait partie des tournois internationaux comptant pour le classement ATP¹. Le Grand Prix de Lyon est classé, avec Roland Garros et l'Open de Paris, parmi les trois tournois les plus renommés de France. La dotation globale 2014 était de 427 000 euros, avec des prix allant de 2200€ en double en 16e de finale, à 77 000€ au vainqueur du tournoi simple. Chaque gagnant d'un niveau de match se voit attribué un gain.

Le GPTL comprend deux tournois : le Simple Messieurs et le Double Messieurs. Les matchs à organiser sont ceux de la Qualification (32 joueurs en 2010 dont 4 d'entre eux ont rejoint le tableau final), puis du tournoi avec les 16e de finale (32 joueurs en Simple, 64 en Double), puis les 8e de finale etc., jusqu'à la finale, appelée *the Big Match*.

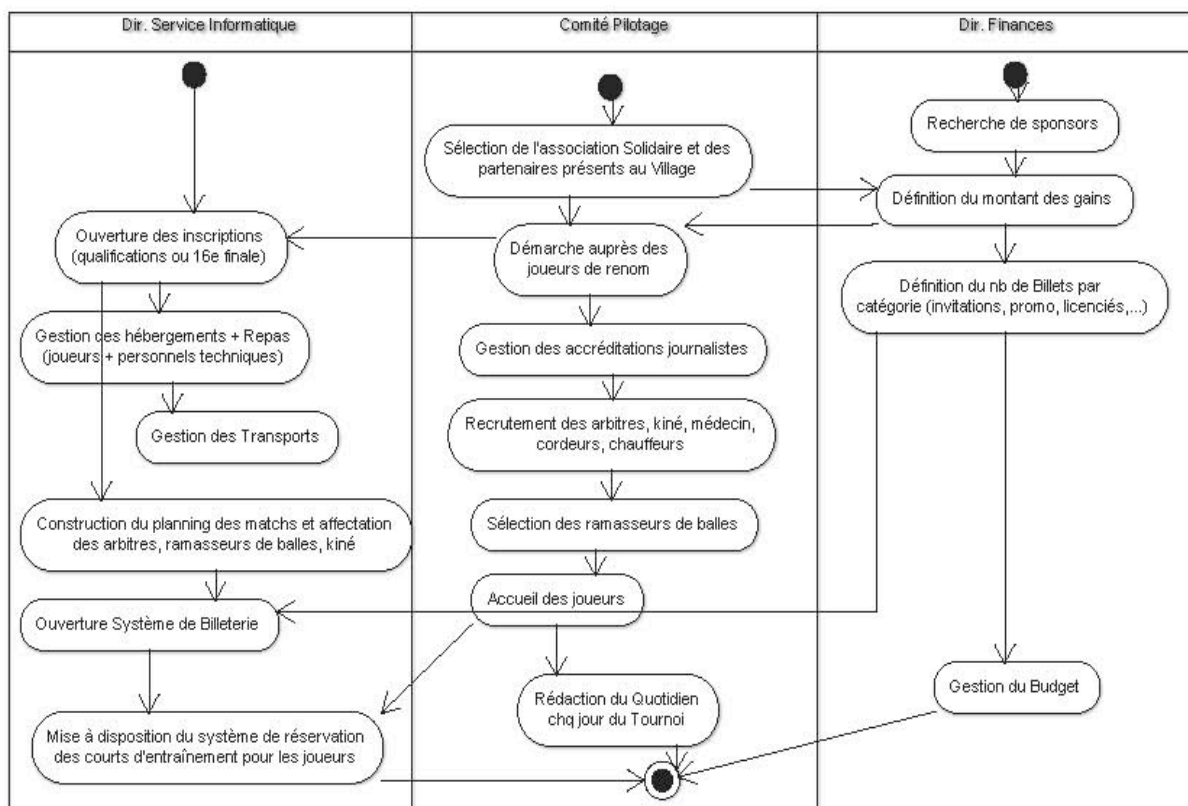
Le grand Palais des Sports de Gerland dispose d'un court central et de 5 autres courts. Le court central est organisé en trois niveaux pour une capacité totale de 5600 spectateurs. Il existe différents types de billets : grand public, licenciés, promotions (associations...), invitations, et billets pour la Finale.

Un village de 3000m² est associé au Grand Prix, où se trouvent des restaurants gastronomiques et des stands liés au monde du tennis.

Les qualifications ont lieu le samedi et le dimanche précédant le tournoi. Durant les 7 jours qui suivent, un quotidien et un programme officiel relatent la vie du tournoi.

Le tournoi est constitué de 4 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 le Grand Prix 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 du GPTL



¹ En 2010, le tournoi de tennis de Lyon a été "relocalisé" à Montpellier, puis à partir de 2017, il devient l'Open Parc Auvergne-Rhône-Alpes Lyon et est organisé sur terre battue au Parc de la Tête d'Or, durant la semaine précédant Roland-Garros.

L'ARBITRAGE

Pour chaque tournoi, il y a un juge-arbitre et 45 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 *lounge* (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 le GPTL offre des services particuliers. C'est le Comité de Pilotage qui gère les VIP.

SPONSORS

Le Grand Prix de Lyon est un évènement fortement médiatisé (Canal+, etc.). Les appels et offres de parrainage sont gérés par le Comité de Pilotage. Le budget global du tournoi s'élevait à 14 millions d'euros en 2010.

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 autre :

- 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, nb de tournois de grands chelems (gagnés, finaliste), auxquelles vous ajouterez des éléments plus subjectifs de votre choix (par ex. un coefficient de popularité -sur une échelle de 10, une probabilité de désistement, etc. Vous pouvez imaginer ce que vous voulez).
- 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) avant ou 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

Cette application permettra entre autre :

- La **saisie des caractéristiques** de l'hébergement par les responsables des hébergements : le type (hôtel, auberge jeunesse, etc.) ; 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.), et le nombre de places (chambres/lits) disponibles chaque jour.
- La **saisie d'une réservation** dans l'établissement : nombre de personnes, dates et vérification des contraintes (arbitres et joueurs ne doivent pas être logés dans le même établissement, l'équipe accompagnant un joueur peut préférer se retrouver au même endroit), et affichage de la capacité disponible (MàJ régulières). En fait les réservations se font majoritairement AVANT le tournoi, 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 de l'équipe du Tournoi, de même que la gestion des réservations des VIP.

Système 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. 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 Promo* : le Comité distribue un certain nombre de codes promotionnels (une dizaine de codes, chacun donnant droit à un certain nb de billets) permettant d'assister au tournoi à des tarifs préférentiels. Le code promotionnel devra être saisi par l'acheteur. La réduction est de 20 à 30% du prix Grand Public.
- *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, comme pour les billets Promo. Cela concerne en général 30% 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.
- *Billets « the Big Match »* : ce sont les places pour les 2 finales et leur prix est aussi fonction des emplacements.

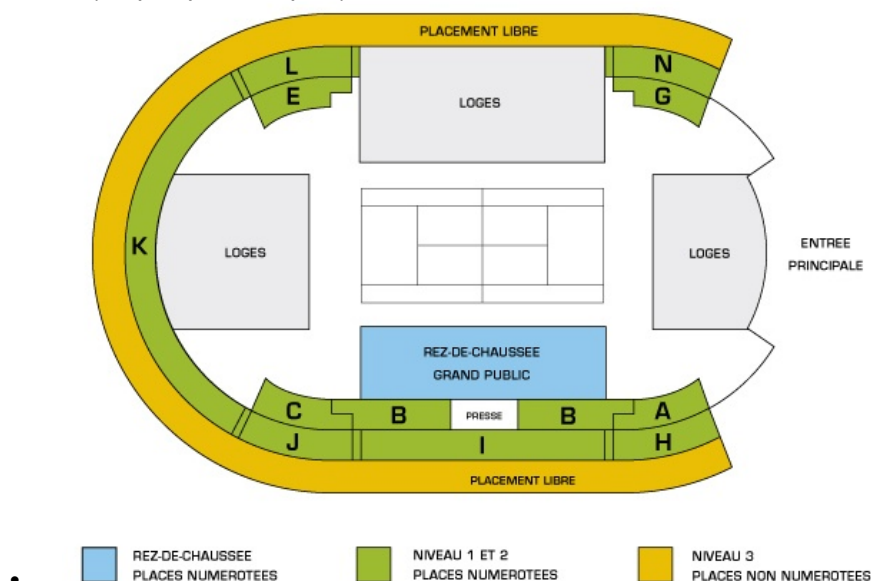
On ne gèrera pas le paiement (juste prévoir une page « Vous allez être redirigé vers le site de paiement ») mais on pensera aux pages de back-office permettant de définir le nombre affecté à chaque catégorie de billets, et le prix de chacun.

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).

Planning des Matches

Répartition des arbitres et ramasseurs de balles sur les matchs, affectation des courts pour les matchs et les entraînements.

Compte tenu des 9 jours de matchs (premier WE de qualification puis 7 jours du Grand Prix), il s'agit de placer les 8 arbitres de lignes et l'arbitre de chaise sur les différents matchs 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). Le Palais des Sports dispose d'un *court central* où ont lieu les matchs de Simple et la finale Double, d'un *court annexe* pour les matchs de Double (et quelques Simples), et de 4 *courts d'entraînement*.



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

La réservation des courts d'entraînement sera disponible pour les joueurs pendant tout le tournoi (un seul nom suffit pour la réservation). Hormis les finales, les matchs sont répartis sur 5 tranches horaires : matin (8h), matinée (11h), midi (15h), après-midi (18h), soirée (21h). Les finales Simple et Double ont lieu le Dimanche (cf programme prévisionnel à titre d'illustration en Annexe). *Attention, des matchs peuvent être organisés aussi sur un court d'entraînement le premier jour, afin de faire tenir les matchs sur la semaine.*

Pour le Planning des Matches, l'application devra **aider l'utilisateur à placer un match** sur un créneau donné, en vérifiant les contraintes. Les contraintes relèvent du type de match : tournoi ou entraînement, pour les matchs de tournoi : Qualifications, tournoi Simple ou Double. Elles concernent aussi les nationalités et catégories d'arbitres (l'arbitre de chaise doit être d'une nationalité différente de celle des joueurs), 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 son nom sur l'horaire choisi.

NOTA : le tableau du tournoi comprend 32 entrées : au min 22 entrées directes, au max 4 *wild Cards* (invitations), 4 qualifiés et 2 *special Exempts* (joueurs admis directement en dépit de leur classement insuffisant).

On imagine par exemple procéder de manière suivante :

Créer un 1^{er} planning des matchs d'un type donné (Simple Messieurs, Double Messieurs, Qualifications) à partir de la liste des joueurs fournie, 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, sur un horaire donné, en respectant les contraintes spécifiées. Prévoir aussi de mentionner le nom des gagnants une fois le match terminé. Il ne doit pas y avoir de conflit entre match Simple et Double joué par un même individu.

Ensuite le module permet au responsable du Planning de **modifier, supprimer ou déplacer** un match du planning existant sur un autre créneau (parmi ceux possibles).

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 d'entraînement à un horaire donné pendant le tournoi. Ces 2 applications seront développées en Java.

(*facultatif*) Gérer les invitations VIP aux différents matchs, en fonction des places disponibles.

Travail demandé

On vous demande de réaliser l'analyse et l'implémentation de l'application GPTL 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 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 *Persistance* des données, de manière à permettre un changement d'interface ou de BD, sans avoir à modifier tout l'ensemble.

Couche Présentation : elle concerne toutes les interactions avec les utilisateurs (modules WEB et Java).

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 dans la description du sujet; 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. Et bien sûr, toutes les gestions des entités Métier seront également à implémenter : création/modification/suppression des différentes entités de l'application (joueurs, VIP, hôtels, matchs, etc.).

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.

Comme précisé précédemment, on supposera que la liste des joueurs sélectionnés pour les 16^e de finale Simple et Double est connue, ainsi que la liste des joueurs passant les qualifications. La liste des arbitres sera aussi déjà affectée. Vous inventerez des noms et des nationalités pour tous les autres éléments requis.

Cf. <http://www.opensuddefrance.com/> pour un accès à toutes les informations.

L'annexe 1 propose un programme prévisionnel (trouvé sur le site d'Open Sud France).

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).

Annexe 1 – (attention : ici le 2^{ème} tour du Double a manifestement été OUBLIE !)

Programme prévisionnel

DIM 31	QUALIFICATIONS : Simple
LUN 1	QUALIFICATIONS : Simple 1ER TOUR : Simple 1ER TOUR : Double
MAR 2	1ER TOUR : Simple 1ER TOUR : Double
MER 3	FIN DU 1ER TOUR : Simple DÉBUT DU 2ÈME TOUR : Simple 1/4 FINALE : Double
JEU 4	2ÈME TOUR : Simple 1/4 FINALE : Double
VEN 5	1/4 FINALE : Simple 1/2 FINALE : Double
SAM 6	1/2 FINALE : Simple 1/2 FINALE : Double
DIM 7	FINALE : Simple FINALE : Double

ANNEXE 2 – GUIDE METHODOLOGIQUE

- L'utilisation d'un AGL est obligatoire pour la génération automatique de code et la rétro conception (reverse engineering), et l'environnement NetBeans est fortement conseillé pour la programmation Java (et le développement web !). On utilisera par exemple le nom Tennis_UML comme nom de projet PowerAMC, Tennis_JAVA pour le projet NetBeans et Tennis_BD pour le répertoire des tables relationnelles.
- Pour les données persistantes, vous choisirez une implémentation par BD avec l'API JDBC pour ORACLE, MySQL ou autre BD.
- Le module a une durée de 44h. En mode traditionnel, l'analyse requiert environ 4 séances (16h), la programmation 6 séances (22h). En mode Agile, vous définissez la durée de sprint, et les éléments ciblés pour chaque itération. Les dernières 6h 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.
- Comme le temps risque d'être court, il est important de **réaliser d'abord tout ce qui concerne le « métier »** (processus métiers décrits dans l'énoncé : planifier les matchs des jours de qualifications, par ex.) AVANT toute « gestion » (gestion des joueurs, des matchs, des arbitres, etc.).
- Pour éviter toute erreur de débutants (écrasement de fichiers) et faciliter le développement à plusieurs, vous utiliserez un **système de gestion des versions** type CVS (*Concurrent Versions System*), GIT ou Mercurial.
- Pour le développement WEB, l'utilisation de **frameworks** est autorisée.
- **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 l'étape programmation, avec durée et affectation aux membres du binôme. Outre les tâches de codage des différentes parties, vous penserez aux tâches annexes comme la formation à un nouvel outil (ex. CVS), la création des tables, les tests, la préparation de la démonstration, etc.
Vous noterez ensuite le temps passé à réaliser chacune des activités pour pouvoir, en fin d'implémentation, **construire un planning réel** et comparer avec le prévisionnel.

Pour la conduite par en **mode agile**, il faudra donner précisément la durée des itérations choisie, mais surtout la trace de tous les *backlogs de sprint* et des tâches précises envisagées en début de chaque itération, afin que l'on puisse évaluer votre façon de gérer le développement agile. Il faudra choisir le *Product Owner* parmi l'un d'entre vous (ou bien demander à l'enseignant s'il est OK).

Les encadrants sont là en tant que « **client** » et pour vous aider avec UML et Java, pas pour diriger votre travail. Il y a plusieurs façons de voir et de procéder pour l'application proposée, chaque binôme construira son système selon les modules choisis et son point de vue qu'il présentera et argumentera 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** (surtout ne pas reprendre l'énoncé fourni tel quel !). Vous vous aiderez pour cela de la génération automatique de rapport d'analyse à partir de l'AGL (à paramétrer : tout rapport de plus de 50 pages ne sera pas lu).

Il comportera au minimum :

- 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". Même si le fonctionnement ne semble pas le meilleur, c'est celui que l'on veut voir modélisé;
- Les **Packages** que vous avez identifiés et leurs relations de dépendance;

Puis par package :

- le **Diagramme des Cas d'utilisation** avec les principaux acteurs et les rôles que ces acteurs vont jouer en interaction avec le système;
- Puis, par **cas d'utilisation** jugé intéressant :
 - o un **Diagramme de Séquence Système** pour chaque cas, avec les scénarios envisagés ;
 - o Un **Diagramme d'Activités** ou **d'interactions (Diagr. de Séquence)** décrivant les cas complexes ;
- Le **diagramme de classes du package**, avec explicitation des attributs et opérations (arguments, types, donner éventuellement les diagrammes d'activités des opérations complexes).

Le dossier incorpore aussi une **partie IHM** : vous fournirez les dessins d'écran et les schémas d'enchaînement des fenêtres, pour illustrer le fonctionnement choisi.

Enfin pour la **partie conception**, on donnera :

- les Diagrammes de Classes de Conception (enrichis des éléments de conception) ;
- le schéma de la BD et le dictionnaire de données ;
- le script de création des tables ;
- les squelettes des classes Java/MySQL issues de la génération automatique de code.

TRES IMPORTANT : les diagrammes 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.

3.1. 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 pour le stockage des données).
- **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 inconvénients (détails envisagés en analyse qui s'avèrent inutiles ou irréalisables lors du codage, contraintes de codage qui conditionnent l'analyse, modification de l'analyse en cours de codage et justifications...).
- **Points d'amélioration / extensions** : vous énumérez ici les extensions que l'on pourrait envisager pour cette application, compte tenu des idées (ou des limites) qui sont apparues en la développant.
- On pourra aussi faire une **rétro-conception** du code développé et comparer avec le modèle d'analyse qui avait été proposé.

3.2. Éléments de gestion de projet

- Donner les **2 plannings des tâches** (prévisionnel et réel) et effectuer la comparaison. Ce travail devrait vous permettre de mieux chiffrer ensuite les durées des tâches ;
- Préciser aussi le **partage des tâches** au sein du binôme (cela peut être mentionné dans le planning réel) ;
- On rendra compte de **l'utilisation d'un système de gestion des versions** du projet ;
- Expliquer le **protocole de tests effectués pour valider l'application**, avec l'utilisation par ex. de JUnit sous NetBeans (démarche, jeux de tests, tests unitaires, tests d'intégration).
- Donner un **bilan personnel** de ce module : ce que vous avez appris, les difficultés rencontrées, etc.