***Football Manager***

***Jeu de simulation d’entraineur de l’équipe***

***Rapport préparé pour****: Tewfik Ziadi (tewfik.ziadi@lip6.fr)*

***Date :*** *10 décembre 2019*

***Equipe :*** *Massyl Djemai, Baptiste Pires****,*** *Francis Murray****,*** *Robin Sanvicente, Mohammed Ichou****,*** *Amine Ait Kaci*

TABLE DES MATIERES

[1. Cahier des Charges 2](#_Toc26901069)

[1.1 Présentation : 2](#_Toc26901070)

[1.2 Partie commune : 2](#_Toc26901071)

[1.3 Particularités dans le cas où le sport est le football : 3](#_Toc26901072)

[1.4 Fonctionnalités possibles pour l’utilisateur 6](#_Toc26901073)

[2. Modelisation 7](#_Toc26901074)

[2.1. Diagramme des cas d’utilisation (Use Case Diagram) 7](#_Toc26901075)

[2.2 Fiches (scenarios) 8](#_Toc26901076)

[2.3. Diagrammes des classes d’analyse 22](#_Toc26901077)

[2.4. Diagrammes de séquences 22](#_Toc26901078)

[2.5. Diagramme des classes raffinée 22](#_Toc26901079)

[2.6 Conception + Design Patterns 23](#_Toc26901080)

[3. Codage 24](#_Toc26901081)

[3.1. Génération automatique de code (Java avec Modelio) 24](#_Toc26901082)

[3.2. Compléter le code métier 24](#_Toc26901083)

[3.3. Implémenter IHM simple (menu etc..) 24](#_Toc26901084)

[3.4. Implémenter quelques use cases. 24](#_Toc26901085)

[4. Organisation 25](#_Toc26901086)

[Gestion des versions 25](#_Toc26901087)

[Répartition des Charges 25](#_Toc26901088)

[5. Annexes 27](#_Toc26901089)

[Annexe 1 – Flow Chart 27](#_Toc26901090)

# 1. Cahier des Charges

## 1.1 Présentation :

Sport Manager est une gamme de jeux de simulation de la carrière d’un manager sportif. Il s’inspire fortement des fameux jeux du type Football Manager ou Pro Cycling Manager. Le principe consiste à jouer le rôle d’un manager sportif, en dirigeant une équipe sportive (ou un seul sportif dans le cas de sport individuel) le long de sa carrière, en évoluant dans le temps. L’enjeu est d’obtenir une simulation proche de la réalité en mettant en place les interactions qu’aura le manager avec les entités externes, que ce soit au sein de l’équipe qu’il entraîne mais également les adversaires et concurrents.

La particularité de ce jeu est qu’il dispose d’un noyau dure ne dépendant pas du sport sur lequel il est implémenté, mais également de différentes variantes en héritant et en mettant en place les particularités d’un sport en particulier.

## 1.2 Partie commune :

* Quel que soit le sport, l’utilisateur joue le rôle d’un manager. Il dispose donc d’un personnage présentant des caractéristiques communes (on peut citer : nom, prénom, date de naissance, historique de la carrière sportive), mais également des caractéristiques particulières qui dépendent du sport sur lequel on implémente le jeu.
* Une partie du jeu consiste donc à créer un manager, puis le mettre dans son environnement, en lui associant une équipe (ou un sportif) à gérer, ainsi que les adversaires (par exemple créer un tournoi, les équipes adverses...etc).
* Nous associons également à la partie une date de début (pas forcément la date du jour réelle, juste la date du début de la simulation).
* Nous associons en plus un système de boîte de réception du manager, car durant la partie, le manager aura une boîte de réception dans laquelle il recevra des messages différents, certains messages seront juste des informations qui par exemple résument les résultats sportifs des concurrents, ou les infos importantes du tournoi ou la compétition dans laquelle nous jouons.
* Le manager dispose également d’un tableau de bord lui permettant de gérer son équipe (ou son sportif), en accédant à toutes les informations relatives au calendrier de l’équipe (prochains matchs...etc), à ses performances et résultats sportifs, à son classement, à l’état des physique et moral des joueurs (ou du sportif).
* Le manager est relié avec son équipe (ou le sportif qu’il entraîne) par un contrat, dont il peut à tout moment consulter ses modalités, et demander sa renégociation (durant les périodes autorisées, selon le sport).
* Le déroulement du jeu se passe en faisant avancer le temps, jusqu’à l’arrivée d’une journée contenant un événement à traiter. Les événements peuvent être de diverses natures, et par conséquent nécessiter un traitement totalement différent. Par exemple, un match peut être vu comme étant un événement, et son traitement serait de préparer la composition de l’équipe et de mettre en place la stratégie à utiliser durant le match.
* Ajoutons que les événements ont des importances différentes, c’est à dire que certains doivent être traités obligatoirement dès qu’ils arrivent (on peut citer l’exemple du match), et d’autres peuvent être reportés.

## 1.3 Particularités dans le cas où le sport est le football :

* Dans le cas du football, le manager dirige une équipe de footballeurs, sous un contrat qu’il signe avec le club (au début de la partie, on assigne un contrat standard d’une durée d’un an, renouvelable, mais l'entraîneur peut changer de club s’il reçoit d’autres offres durant la période où cela est possible), avec le président du club.
* Un joueur est de football est considéré comme une personne avec un poste de prédilection (gardien de but ou défenseur ou milieu de terrain ou attaque), des notes sur ses capacités physiques et techniques, (note de défense, note de milieu de terrain, note d’attaque) qui progressent au fil des années. Ajoutons à cela 3 notes un peu plus dynamique sur l’état physique, l’état moral et la note moyenne (selon ses performances dans les matchs) du joueur à un instant donné. Notons que l’on mémorise également le nombre de buts marqués, de passes décisives, de cartons jaunes et de cartons rouges.
* Une équipe de football est composée d’une liste de joueurs (au moins 18 joueurs par équipe), d’un stade (sachant qu’un stade est représenté par son nom, le nom de sa ville, ainsi que le nombre de places dont il dispose), d’un président, d’un entraîneur (c’est la place que prend l’utilisateur du jeu). A tout moment, l’équipe dispose d’une stratégie par défaut, d’une masse salariale (donnée en euros/mois), ainsi qu’un budget pour les transferts (donné en euros).
* Une stratégie est associée à une équipe. Elle contient des notes (de 1 à 100) qui fixent le degré de : l’agressivité sur le terrain (qui maximise les chances d’avoir des blessures et des cartons jaunes et rouges), les passes, la nature des passes (de 0 à 33 pour passes longues, de 33 à 66 pour passes moyennes et de 66 à 100 pour passes courtes), ainsi qu’une note sur le taux de tirs de loin. Ajoutons à cela une note de 100 points à partager en 3, une partie pour l’attaque, l’une pour le milieu de terrain et l’autre pour la défense (exemple : 20 pour le milieu, 50 pour l’attaque et 30 pour la défense). De plus elle comporte un schéma tactique (3-5-3, 4-4-2, 5-3-1... Il s’agit d’un triplet d’entiers, le premier représente le nombre de défenseurs, le second le nombre de milieux de terrain, et le 3ème le nombre d’attaquants).
* Un championnat est une compétition qui s’effectue en boucle, qui se déroule le long de l’année. Chaque instance d’un championnat est appelée saison, donc au bout de chaque saison, il y a un classement final, et un gagnant du championnat, ainsi qu’un joueur désigné.
* Le championnat est composé d’une liste d’équipes qui s’affrontent selon un calendrier, ainsi que d’une liste d’arbitres. Un calendrier est organisé en journées, et chaque journée est composée d’une liste de matchs ainsi que d’une date, sachant qu’on considère qu’il y a une journée de championnat par semaine.
* Pour un championnat de n équipes, il y a 2\*(n-1) journées, c’est à dire que chaque équipe dispute 2 matchs avec chacune des équipes du championnat (Un match aller et un match retour), ces journées sont disposées de manière à ce que pour chaque match aller d’une équipe, le match retour se déroule (n-1) journées après (donc les matchs qui se disputent de la journée 0 à la journée n-1 sont identiques dans l’ordre à ceux des journées n à 2 (n-1) ).
* Une journée d’un championnat de n équipes est composée de n/2 matchs, c’est à dire que toutes les équipes doivent participer à chaque journée.
* Un match est une rencontre entre deux équipes, dans un stade (le stade d’une des deux équipes), arbitré par un arbitre choisi par le championnat. Durant un match, chacune des deux équipes présente une composition de 11 joueurs selon une formation donnée (exemple : 1 gardien, 3 défenseurs, 4 milieux, 5 attaquants), et à la fin du match, on obtient un score, c’est à dire le nombre de buts marqués par chacune des équipes, ainsi que des statistiques générales du match (pour chacune des équipes le pourcentage de possession de balle, le nombre de passes, le nombre de tirs, le nombre de cartons jaunes et le nombre de cartons rouges), mais également des statistiques particulières pour chaque joueur (note moyenne sur 10, nombre de buts, nombre de cartons jaunes, carton rouge ou pas).
* Notons que dans la génération du calendrier du championnat, on fait en sorte que pour deux équipes A et B, le match aller A contre B se jouera dans le stade de A et le match retour dans le stade B.
* Nous modélisons les championnats européens pour simplifier (notamment le championnat de France, d’Espagne, d’Italie, d’Angleterre, et d’Allemagne).  
  Tous les championnats dépendent d’un organisme global, qu’on appelle l’UEFA (Union des associations européennes de football), qui gère tous les championnats européens, ainsi que le tournoi de la ligue des champions, dans lequel participe les 4 leaders de chaque championnat européen (ceci est différent de la réalité, mais nous considérons ça dans notre simulation pour simplifier).
* Une équipe fait partie d’un championnat national (selon le pays de l’équipe dirigée), ainsi qu’à la Ligue des champions si elle a fait partie des 4 premiers dans le classement de son championnat à la fin de la saison précédente.  
  Dans les championnats nationaux, on attribue des points à chaque équipe au fil des journées de la saison : si une équipe gagne un match elle obtient 3 points, si elle fait un match nul (égalité) elle en obtient 1 et si elle perd un match elle ne gagne aucun point.
* Dans la ligue des champions, on change les règles, dès qu’une équipe perd un match elle est éliminée de la compétition. On obtient donc une finale et un seul gagnant qui n’aura perdu aucun match. Les matchs de la Ligue des champions se jouent en terrain aléatoire. Le calendrier de la ligue des champions est désigné par un tirage au sort à chaque tour qui détermine qui joue contre qui.
* Les matchs de championnat se déroulent les dimanches, à chaque semaine. Ceux de ligue des champions les mardis, chaque trois semaines.
* 2 fois par an, il y a une phase de transition que l’on appelle Mercato. Il s’agit d’une période durant laquelle les équipes font des transferts de joueurs et d'entraîneurs. Il y en a une à la fin de la phase aller des championnats européens (c’est à dire au bout de la journée n-1 pour un championnat de n équipes), pour un mois; ainsi qu’une durant l’été (pour 2 mois).
* Durant les périodes de mercato, l’utilisateur à 2 fonctionnalités qu’il peut faire : Soit acheter des joueurs et en vendre, en étudiant des offres qu’on reçoit d’autres clubs, ou bien, changer de club pour lui même dans le cas ou il reçoit des offres d’autres clubs qui voudraient qu’il devienne leur entraîneur.
* Dans tous les cas de transferts entre joueurs ou entraîneurs, un contrat est généré contenant le salaire du signataire et les dates de validité du contrat.
* La difficulté du jeu consiste à atteindre les objectifs que le président du club nous fixe. (Par exemple, il peut exiger que l’équipe se classe au moins 3ème du championnat, et qu’elle ne perde pas plus de 3 matchs consécutifs à domicile... ). Sinon, on risque de se faire licencier par le président et se retrouver sans club, auquel cas on aura perdu la partie.

## 1.4 Fonctionnalités possibles pour l’utilisateur

* Faire avancer le temps jusqu’au prochain événement.
* Parcourir les championnats (en affichant leurs informations et statistiques dans l’interface). (à considérer comme 3 use cases distincts)
* Parcourir les équipes (en affichant leurs informations et statistiques).
* Parcourir les joueurs (en affichant leurs informations et statistiques).
* Interagir avec les joueurs en leur envoyant des offres de transfert.
* Interagir avec l’administration de notre club pour allouer un budget plus important pour les salaires ou pour les transferts (demande qui peut être acceptée ou rejetée, selon le degré de satisfaction de l’administration envers l'entraîneur).
* Parcourir ses messages reçus (fonctionnalité facultative, faite juste dans le cas ou l’on implémente la partie messagerie de l'entraîneur).
* Pouvoir à tout moment mettre en place la stratégie par défaut de notre équipe
* Pouvoir mettre en place juste avant un match une stratégie différente que celle par défaut.
* L’administration offre à l’équipe une cagnotte pour les festivités, qui augmente à chaque victoire d’un match. Cette cagnotte est utilisée pour organiser des team building afin de remonter le moral aux joueurs. Il y a donc possibilité d’organiser un team building SI le budget est suffisant (le mieux est de le faire quand on sent que notre équipe à besoin d’un coup positif de moral).

# 2. Modelisation

## 2.1. Diagramme des cas d’utilisation (Use Case Diagram)

A close up of a logo

Description automatically generated

## 2.2 Fiches (scenarios)

|  |  |
| --- | --- |
| Nom | Consultation des messages |
| Scénario | L’utilisateur souhaite consulter la messagerie pour voir si il a reçu des messages |
| Acteur | Entraîneur (utilisateur) |
| Objectif dans le contexte | Permettre au joueur de consulter ses messages |
| Pré conditions | 1. Application lancée 2. Partie créée |
| Post Conditions (réussite) | 1. Affichage des messages au centre de l’écran |
| Post Conditions (échec) | 1. Avertir l’utilisateur qu’il y a un problème lors de l’affichage de ses messages et le renvoyer sur la page d’accueil. |
| Règle de gestion | 1. La boîte de réception contient les messages anciens, et les messages non lus avec un titre écrit en gras |
| Description | Action |
| 1.Le joueur clique sur le bouton “Boîte de réception” (ou Messagerie) |
| 2.L’application change d’écran pour afficher l’écran dédié aux messages |
| 3.Une fois sur l’écran qui affiche les messages, elle affiche un chargement |
| 4.   1. Si la requête réussi, alors elle affiche une liste de message ou une phrase pour indiquer qu’il n’y a pas de messages 2. Sinon, elle affiche une fenêtre pop-up à l’utilisateur qui lui indique que sa demande à échouer |
| Priorité | Elevée |

|  |  |
| --- | --- |
| Use Case | Consulter les informations des joueurs. |
| Scénario | Happy Path |
| Objectif dans le contexte | Permet à l’utilisateur de visualiser et gérer l’effectif de joueurs de son équipe, ainsi que consulter les informations de n’importe quel joueur (même en dehors de l’équipe que l’on entraine). |
| Pré conditions | Avoir commencé une partie  Avoir lancé le jeu. |
| Post Conditions (réussite) | Affichage des joueurs et de leur statistiques (physiques, morales, performances durant les derniers matchs). |
| Post Conditions (échec) | Affichage refusé, donc redirection à l'écran principal. |
| Règle de gestion | n/a |
| Description | 1. Depuis le Tableau de bord, il clique sur “joueurs” 2. La liste des joueurs de notre équipe s’affiche. 3. L’utilisateur clique sur le joueur dont il veut consulter les détails. 4. La page du joueur s’affiche, avec sa photo et le détail de ses statistiques. |
| Extensions | L’utilisateur interagit avec un joueur en lui envoyant une offre de transfert (s’il est en dehors de notre équipe), ou renégocier son contrat (s’il est dans notre équipe). |
| Priorité | Très important |

|  |  |
| --- | --- |
| Use Case | Interagir avec les joueurs en leur envoyant une offre de transfert vers notre club |
| Scénario | Happy Path |
| Objectif dans le contexte | Offre à l’utilisateur la possibilité d’envoyer une offre de transfert à un joueur pour améliorer et perfectionner notre équipe |
| Pré conditions | Avoir commencé une partie  Avoir lancé le jeu.  Avoir affiché les informations d’un joueur |
| Post Conditions (réussite) | Réception d’un message dans notre boîte de réception indiquant que le joueur en question a reçu notre demande de transfert |
| Post Conditions (échec) | Aucun message de confirmation reçu |
| Règle de gestion | Le budget des transferts qui nous est alloué doit être supérieur au montant du transfert que l’on propose, et la masse salariale suffisante également |
| Description | 1. En affichant un joueur donné, cliquer sur le bouton proposer transfert 2. Fixer les dates de début et de fin du contrat proposé 3. Fixer le salaire offert 4. Fixer le montant du transfert 5. Valider sur le bouton de validation de l’offre pour confirmer l’envoi de l’offre |
| Extensions | Avoir un bouton de redirection vers l’écran de l’administration afin de pouvoir demander une rallonge pour augmenter le budget des transferts |
| Priorité | Intéressant mais pas primordial |

|  |  |
| --- | --- |
| Use Case | Consulter les informations des équipes. |
| Scénario | Happy Path |
| Objectif dans le contexte | Permet à l’utilisateur de visualiser les informations de son équipe, listant ses joueurs, son calendrier et ses résultats. |
| Pré conditions | 1. Avoir commencé une partie 2. Avoir lancé le jeu. |
| Post Conditions (réussite) | Affichage de la liste des joueurs de l’équipe et de leur statistiques, ainsi que le calendrier de l’équipe et ses résultats |
| Post Conditions (échec) | Affichage refusé -> redirection vers l'écran principal. |
| Règle de gestion | n/a |
| Description | 1. L’utilisateur consulter le Tableau de Bord 2. Depuis le Tableau de bord, il clique sur “mon équipe” 3. La liste des joueurs s’affiche. 4. L’utilisateur clique sur le joueur dont il veut consulter les détails. 5. La page du joueur s’affiche, avec sa photo et le détail de ses statistiques. |
| Extensions | L’utilisateur interagit avec un joueur en lui envoyant une offre de transfert. |
| Priorité | Très important |

|  |  |
| --- | --- |
| Use Case | Consulter le tableau de bord |
| Scénario | Happy Path |
| Objectif dans le contexte | Donner à l’utilisateur une vision globale de la partie. |
| Pré conditions | 1. Avoir commencé une partie 2. Avoir lancé le jeu. |
| Post Conditions (réussite) | Affichage d’informations sur l’équipe : son calendrier, ses derniers résultats, les budgets, ainsi que des informations sur le championnat de l’équipe : le calendrier de la prochaine journée, le classement des équipes, le classement des buteurs(10 premiers), et le classement des passeurs décisifs (10 premiers) |
| Post Conditions (échec) | Affichage refusé message d’erreur affiché. |
| Règle de gestion | n/a |
| Description | 1. L’utilisateur clique sur  “Tableau de Bord” |
| Extensions |  |
| Priorité | Très important |

|  |  |
| --- | --- |
| Use Case | Consulter les informations de l’administration |
| Scénario | Happy Path |
| Objectif dans le contexte | Donner à l’utilisateur une interface contenant les informations de l’administration du club, en donnant un aperçu sur les finances du club. |
| Pré conditions | 1. Avoir commencé une partie 2. Avoir lancé le jeu |
| Post Conditions (réussite) | Affichage d’informations sur l’administration du club : nom du président, budget du club pour les transferts, budget du club pour les salaires (masse salariale), budget du club pour les team building, nom du stade, informations générales du club entraîné |
| Post Conditions (échec) | Retour à la page du tableau de bord. |
| Règle de gestion | n/a |
| Description | 1. L’utilisateur clique sur  “Administration” |
| Extensions | Possibilité de demander une augmentation du budget des transferts, des salaires ou des team building.  Possibilité de mesurer le degré de satisfaction de l’administration envers nous |
| Priorité | Peu important |

|  |  |
| --- | --- |
| Use Case | Demander une augmentation du budget des transferts |
| Scénario | Happy Path |
| Objectif dans le contexte | Donner la possibilité à l’utilisateur de demander une rallonge du budget alloué aux transferts. Cela afin d’avoir la possibilité de faire des offres de transferts avec de plus grands montants |
| Pré conditions | 1. Avoir commencé une partie 2. Avoir lancé le jeu |
| Post Conditions (réussite) | Réception d’un message de réponse de l’administration, acceptant ou refusant notre demande |
| Post Conditions (échec) | Non réception de message de réponse de l’administration |
| Règle de gestion | n/a |
| Description | 1. L’utilisateur clique sur  “Administration” 2. L’utilisateur clique sur le bouton demander rallonge du budget des transferts 3. L’utilisateur indique le montant souhaité puis valide sa requête grâce au bouton de validation |
| Extensions |  |
| Priorité | Peu important |

|  |  |
| --- | --- |
| Use Case | Demander une augmentation de la masse salariale |
| Scénario | Happy Path |
| Objectif dans le contexte | Donner la possibilité à l’utilisateur de demander une rallonge du budget alloué aux salaires. Cela afin d’avoir la possibilité de faire des offres de contrats aux joueurs (les nôtres et aussi ceux que nous recrutons) avec de plus grands salaires |
| Pré conditions | 1. Avoir commencé une partie 2. Avoir lancé le jeu |
| Post Conditions (réussite) | Réception d’un message de réponse de l’administration, acceptant ou refusant notre demande |
| Post Conditions (échec) | Non réception de message de réponse de l’administration |
| Règle de gestion | n/a |
| Description | 1. L’utilisateur clique sur  “Administration” 2. L’utilisateur clique sur le bouton demander rallonge de la masse salariale 3. L’utilisateur indique le montant souhaité puis valide sa requête grâce au bouton de validation |
| Extensions |  |
| Priorité | Peu important |

|  |  |
| --- | --- |
| Use case | Organiser un team building à l’équipe |
| Scénario | L’utilisateur choisit une date durant laquelle sera organisé un team building pour remonter le moral aux joueurs, sous réserve de disposer du budget suffisant |
| Objectif | Remonter le moral des joueurs |
| Pré conditions | 0)Jeu lancé  0’)Partie créée  1)Disposer du budget suffisant pour les team building  2)Que la date du team building ne soit pas occupée par un match |
| Post conditions (réussite) | Le moral des joueurs remonte individuellement |
| Post condition (échec) | Le moral des joueurs reste comme il était avant la tenue du team building |
| Règles de gestion | On vérifie que le club dispose du budget suffisant pour organiser un team building, et également si |
| Description | 0)Si le budget alloué au team building est insuffisant, demander une rallonge de budget à l’administration du club  1)L’utilisateur (donc l'entraîneur) choisi une date (date sans match programmé) pour le team building  2)Il valide son choix  3)On avance le temps (et éventuellement on traite des événements au passage)  4)Dès que la date du team building arrive, l’utilisateur reçoit un message dans sa boite de récéption l’informant de la tenue du team building  5)Après ça, le moral des joueurs de l’équipe augmente de quelques points, et le budget alloué au team building diminue |
| Extensions | 0->1)Dans le cas ou le budget est insuffisant, on passe au use case “Demander d'allouer plus de budget pour les team building”. 0->2)Si la demande est acceptée, on peut revenir à 1, sinon le scénario ne peut pas aboutir  1->1)Si l’utilisateur choisi une date durant laquelle un match était prévu, on signale cette erreur, et on demande de choisir une nouvelle date  1->2)Si la nouvelle date choisie est valide, on peut se revenir à 2, sinon on revient à 1->1 |
| Priorité | Niveau de criticité faible |

|  |  |
| --- | --- |
| Use Case | Demander une augmentation du budget des team buildings |
| Scénario | Happy Path |
| Objectif dans le contexte | Donner la possibilité à l’utilisateur de demander une rallonge du budget alloué aux team building. Cela afin d’avoir la possibilité de faire des team building pour améliorer le moral de l’équipe, comme décrit dans le use case correpondant |
| Pré conditions | 1. Avoir commencé une partie 2. Avoir lancé le jeu |
| Post Conditions (réussite) | Réception d’un message de réponse de l’administration, acceptant ou refusant notre demande |
| Post Conditions (échec) | Non réception de message de réponse de l’administration |
| Règle de gestion | n/a |
| Description | 1. L’utilisateur clique sur  “Administration” 2. L’utilisateur clique sur le bouton demander rallonge du budget des team buildings 3. L’utilisateur indique le montant souhaité puis valide sa requête grâce au bouton de validation |
| Extensions |  |
| Priorité | Très peu important |

|  |  |
| --- | --- |
| Use Case | Consulter les informations des championnats |
| Scénario | Happy Path |
| Objectif dans le contexte | Donner à l’utilisateur un aperçu des résultats des autres championnats que le nôtre (et le nôtre également). |
| Pré conditions | 1. Avoir commencé une partie 2. Avoir lancé le jeu. |
| Post Conditions (réussite) | Affichage d’informations sur le championnat sélectionné : le classement des équipes, le classement des meilleurs buteurs, le classement des meilleurs passeurs, le classement des cartons jaunes et des cartons rouges, par équipes et par joueurs, calendrier de la prochaine journée (avec éventuellement possibilité de consulter les journées suivantes) |
| Post Conditions (échec) | Retour au tableau de bord. |
| Règle de gestion | n/a |
| Description | 1. L’utilisateur clique sur  une liste déroulante en haut de l’écran, appelée “championnats” 2. L’utilisateur choisit le championnat qu’il veut consulter et clique dessus 3. Il obtient l’affichage décrit dans la ligne Post condition “réussite” |
| Extensions |  |
| Priorité | Pas important |

|  |  |
| --- | --- |
| Use case | Visualiser la stratégie actuelle de l’équipe |
| Scénario | L’utilisateur clique sur un bouton “stratégie” qui le redirige vers une page lui permettant de voir la stratégie de son équipe en détail |
| Objectif | Pouvoir voir la stratégie et la formation utilisée par notre équipe, pour pouvoir la modifier (dans un autre use case) |
| Pré conditions | 0)Jeu lancé  0’)Partie créée  1)Avoir appuyé sur le bouton stratégie |
| Post conditions (réussite) | La stratégie de l’équipe avec sa composition dans un terrain de football à droite de l’écran est affichée |
| Post condition (échec) | On n’affiche pas la stratégie de l’équipe et sa composition |
| Règles de gestion |  |
| Description | 0)L’utilisateur clique sur le bouton stratégie |
| Extensions | Pouvoir modifier la composition de l’équipe et la stratégie utilisée par l’équipe |
| Priorité | Important mais pas primordial |

|  |  |
| --- | --- |
| Use case | Modifier la stratégie actuelle de l’équipe |
| Scénario | L’utilisateur, après s’être placé sur l’écran de visualisation de la stratégie à la possibilité en modifiant l’emplacement des joueurs de la liste à gauche de l’écran de modifier la composition de l’équipe et de voir ce changement au niveau du terrain de football dessiné à droite de l’écran. De plus il peut modifier la formation (c’est à dire le placement des joueurs, en le visualisant sur l’écran de droite). Il a aussi la possibilité de régler quelques paramètres de stratégie (attaque, défense, agressivité... )(soit des booléens ou des valeurs entre 1 et 100) |
| Objectif | Pouvoir personnaliser la stratégie et la formation utilisée par notre équipe, pour pouvoir l’appliquer dans les matchs de notre équipe |
| Pré conditions | 0)Jeu lancé  0’)Partie créée  1)Avoir appuyé sur le bouton stratégie |
| Post conditions (réussite) | La stratégie de l’équipe avec sa composition dans un terrain de football à droite de l’écran est bien modifiée selon notre volonté |
| Post condition (échec) | La stratégie n’a pas été visiblement modifiée selon l’affichage |
| Règles de gestion | On n’autorise pas plus de 18 joueurs au total dans l’effectif, pas plus de 11 joueurs titulaires, et pas plus de 7 remplaçants |
| Description | 0)L’utilisateur clique sur le bouton stratégie  1)L’utilisateur sélectionne la formation (4-4-2, ou 4-3-3… un triplet de la forme nombre de défenseurs-nombre de milieux de terrain-nombre d’attaquants)  2)L’utilisateur sélectionne les joueurs titulaires en choisissant leur position (en cliquant sur une liste déroulante à gauche de leur nom sur la liste des joueurs, cette liste déroulante propose tous les postes encore disponibles, c’est à dire non affectés)  3)L’utilisateur fixe des paramètres de la stratégie, à savoir l’attaque, la défense, l’agressivité, ceux là sous forme de nombre entre 1 et 100  4)Fixer d’autres paramètres de la stratégie : choisir entre passes courtes ou longues, choisir entre contre attaquer et construire progressivement, choisir entre défense haute ou basse, choisir entre jeu sur les ailes ou axial  5)Cliquer sur sauvegarder la stratégie pour appliquer les changements |
| Extensions | Pouvoir sauvegarder plusieurs stratégies et choisir l’une d’entre elles |
| Priorité | Important mais pas primordial |

|  |  |
| --- | --- |
| Use case | Faire avancer le temps jusqu’au prochain événement |
| Scénario | L’utilisateur fait avancer le temps jusqu’au prochain événement, c’est à dire le prochain moment ou il peut faire une action, comme un match, ou la réponse d’une offre de transfert qu’on a fait |
| Objectif | Faire avancer la partie |
| Pré conditions | 0)Jeu lancé  0’)Partie créée  1)Avoir traité tout événement qui est à traiter obligatoirement  2)Appuyer sur le bouton avancer |
| Post conditions (réussite) | La date affichée avance |
| Post condition (échec) | La date affichée ne change pas |
| Règles de gestion | On vérifie que l'entraîneur a déjà traité tous les événements obligatoires |
| Description | 0)L’utilisateur clique sur le bouton avancer |
| Extensions |  |
| Priorité | Essentiel |

## 2.3. Diagrammes des classes d’analyse

A close up of a map

Description automatically generated

## 2.4. Diagrammes de séquences

🡪 voir documentation Modelio

## 2.5. Diagramme des classes raffinée

🡪 voir documentation Modelio

## 2.6 Conception + Design Patterns

Singleton :

Factory :

Observer :

# 3. Codage

## 3.1. Génération automatique de code (Java avec Modelio)

## 3.2. Compléter le code métier

## 3.3. Implémenter IHM simple (menu etc..)

## 3.4. Implémenter quelques use cases.

# 4. Organisation

## Gestion des versions

**Utilisation de Git :**

On gère les conflits manuellement au niveau du code

**Adaptation à Modelio pour permettre le round trip**

🡪 release

## Répartition des Charges

**Massyl Djemai**

**Baptiste Pires**

**Francis Murray**

**Robin Sanvicente**

**Mohammed Ichou**

**Amine Ait Kaci**

# 5. Annexes

## Annexe 1 – Flow Chart

