**Génie Logiciel**

*Feuilles de marque pour tir à l’arc*

Nom du projet : ***TANGU***

Table des matières

**I-IntroductionI**

**II-****Présentation du projetI**

Objectif du projet1

**III-Organisation du projetII**

Tapez le titre du chapitre (niveau 2)5

Tapez le titre du chapitre (niveau 3)6

**III-** **Planning et exécution du projet  II**

Tapez le titre du chapitre (niveau 2)5

**I-Introduction**

Au sein de l’enseignement Génie logiciel de la L2 informatique, nous sommes emmenés à gérer la création et le déroulement d’un projet au long du semestre.

Alors, le 25 Septembre, Monsieur Thierry Brouard, le « Product Owner », nous a proposé le back log de l’application « Feuilles de marque pour tir à l’arc ». Après quelques recherches et concertations, l’équipe a trouvé ce projet très intéressant. Et même si on ne connaissait pas vraiment le tire à l’arc, on était très intrigués par le fonctionnement de ce sport et son system de gestion d’après nos recherches.

L’équipe était presque déjà formée (On a déjà travaillé l’an dernier) et s’est complété avec l’adhésion de Matthias Brown.

Ainsi, nous nous sommes mis au travail afin de délivrer ‘TANGU’ au client en mettant tout en œuvre afin de respecter le DEADLINE indiqué.

**I-Présentation du projet**

Cette première partie portera sur l’analyse du Back log du projet fourni par le Product Owner\*.

Cette analyse permet de comprendre le noyau de l’application requise.

Cette application sera amenée à enregistrer les performances des entrainements des tirs à l’arc afin d’optimiser sa progression à travers le stockage des scores des entrainements et donc le calcul des statistiques de tirs. On souhaitera en effet obtenir des statistiques des tirs passés et l’évolution de certains variables clés.

L’application est ciblée Smartphone/phablette :

On aura un seul type d’utilisateur : L’archer en question utilisant l’application pendant ses sessions d’entrainements (On doit donc privilégier la portabilité et les entrainements potentiellement mais rarement hors zones de couverture données internet.

Un nouvel archer sera amené à s’enregistrer et se connecter à son compte. A partir de ce point, il pourra commencer par :

Créer un nouvel arc en précisant le nom de l’arc, son poids, taille et d’autres caractéristiques clefs.

Créer un nouveau blason en précisant ses caractéristiques.

Mettre à jour son profil pour indiquer le nom de son club s’il le souhaite par exemple.

Du moment où il a ajouté au moins les informations sur l’arc et le blason, il pourra commencer un nouvel entrainement en les utilisant.

Ce nouvel entrainement est caractérisé par son nom, la distance, adresse, date, équipements utilisés et sa formule de tir (nombre de séries, volées/série et flèches/volée).

La création va déclencher l’entrainement. Il va saisir les points qu’il a eu à chaque tir jusqu’à finir ses séries.

A la fin de chaque volée/série, il pourra consulter les statistiques de ses tirs avec des calculs sur sa moyenne par volée, le pourcentage moyen de flèches en zone 9 et pourcentage de flèches en zone 10. Finalement, on fait construire un graphique représentant ces données.

Finalement il ne restera plus qu’à implémenter le programme en forme d’application Android.

Le back log est donc très précis et clair sur les fonctionnalités attendues.

**III-Organisation du projet**

Comment a-t-on organisé le déroulement et le travail ?

On mettra en lumière dans cette partie :

A -Les choix du cycle de vie (et pourquoi ?).

B -Les ressources utilisés (Matérielle, logicielle).

C -Les rôles et responsabilités de chacun dans le projet.

D -Identification des risques et les mesures prises pour les éviter.

A -Les choix du cycle de vie.

Pour notre cycle de vie, nous avons choisi SCRUM (d’où le vocabulaire utilisé depuis le début) pour son côté libre et flexible par rapport à l’équipe de développement. Nous l’avons trouvé plus adapté au projet dans le sens où on sera amenés à utiliser plusieurs langages et outils qui sont connus pour certains et pas les autres, et donc cela permettrait de faire tourner les taches en faisant profiter le programme des compétences de tous partout sans gêner l’organisation prévu.

B -Les ressources utilisés (Matérielle, logicielle)

Ayant choisi Scrum, il était évident qu’on allait privilégier un contact très rapproché dans la Scrum Team. Grace aux moyens mis à dispositions par l’université, on a pu utiliser très souvent des salles libres pour effectuer des taches allant du Daily Scrum jusqu’à la conception des modules et l’organisation des sprints.

Ensuite il était nécessaire d’utiliser des AGL pour la conception-modélisation des bases de données, sprints et conceptions de code.

Pour la gestion du projet, l’essentiel était fait sur des éditeurs de texte et des tableurs (pour les documents illustrant les sprints par exemple).

Nous avons travaillé très souvent ensemble (à l’université comme dit précédemment) mais aussi utilisé des logicielles tel que Discord, Slack.

Pour la gestion des fichiers, on utilise Mega.nz pour soumettre les versions stables et d’autres fichiers de gestion. Et pour le versionning, on manipule GitHub (Parce que c’est pratique en groupe) avec des branches différentes pour chacun permettant de pouvoir aller chercher les dernières versions publiées de chaque fichier ou même retrouvés d’anciens fichiers utiles.

Langages utilisés : HTML, CSS responsive, PHP, MYSQL, JavaScript, Ajax, Java.

Framework et plateformes utilisés : StarUML, PHP Storm, WampServer, Android Studio, Power Designer, Bootstrap, Swiper Javascript et d’autres bibliothèques javascript pour les graphes.

C -Les rôles et responsabilités de chacun dans le projet.

***Scrum Team:***

|  |  |
| --- | --- |
| *Product Owner/Encadrant:* | **Thierry Brouard** |
| *Scrum Master* | **Mahmod Alhabaj** |
| *Development Team:* | **-Tom Belda**  **-Mahmod Alhabaj**  **-Matthias Brown**  **- Mohamad-Ali Dakroub**  **- Gérard Doglobe** |

Il est vrai que chaque membre va toucher au moins une fois à chaque principe (code, organisation, docs, tests). Néanmoins, chacun possède quand même un titre caractérisant son rôle dominant et ses points forts pour savoir à qui avoir recours en présence de problèmes.

Les tâches sont : Analyse, Documentation, Développement, Tests, Design.

L’analyse des différents partis a toujours était fait en groupe. Il n’y a donc pas vraiment de responsable. Néanmoins, voici la liste des responsabilités de chacun :

Mahmod Alhabaj : *Scrum Master, développeur, éditeur de documents*.

Equipe de développement

Matthias Brown : *Développeur et graphiste.*

Tom Belda : *Editeur de documents, testeur.*

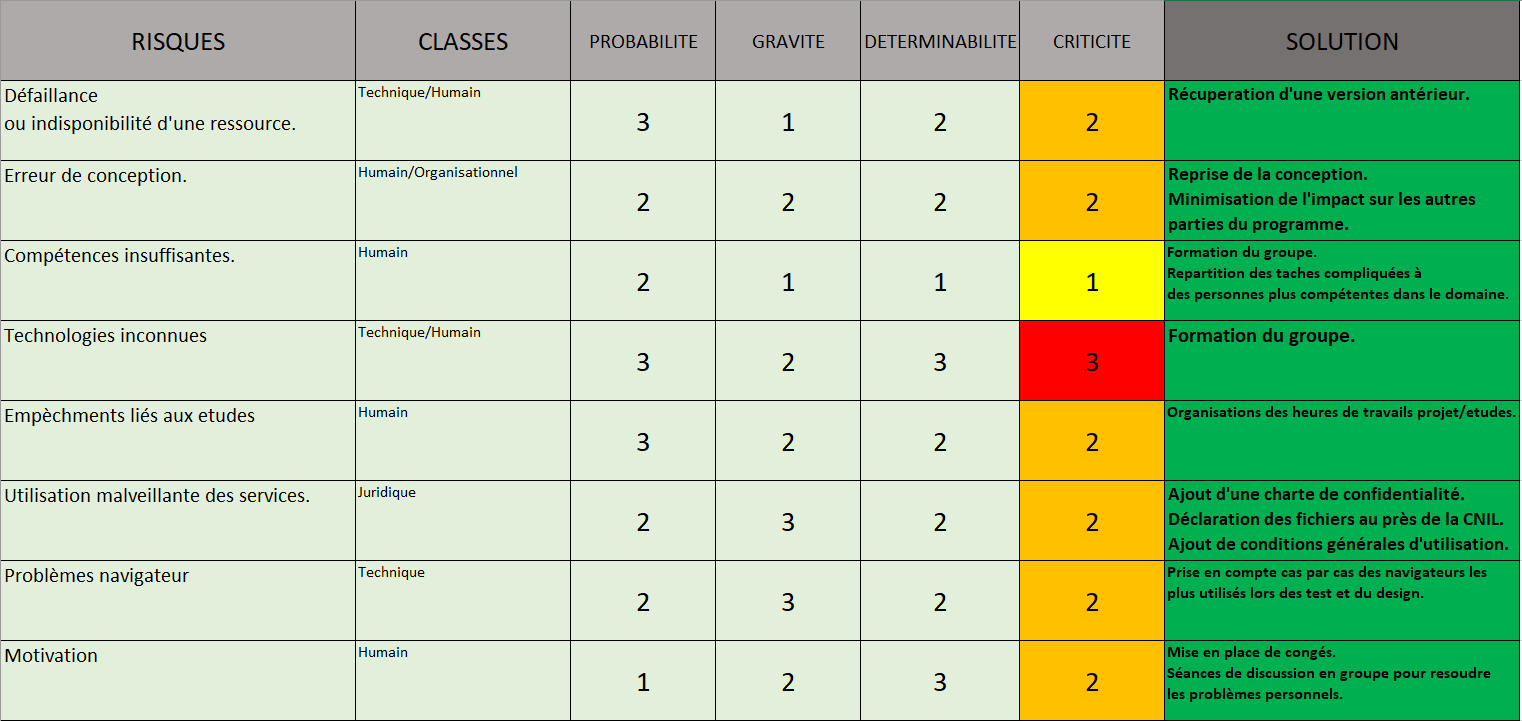
Mohamad-Ali Dakroub : *Graphiste.*

Equipe design graphique

Gérard Doglobe : *Graphiste.*

D -Identification des risques et les mesures prises pour les éviter.

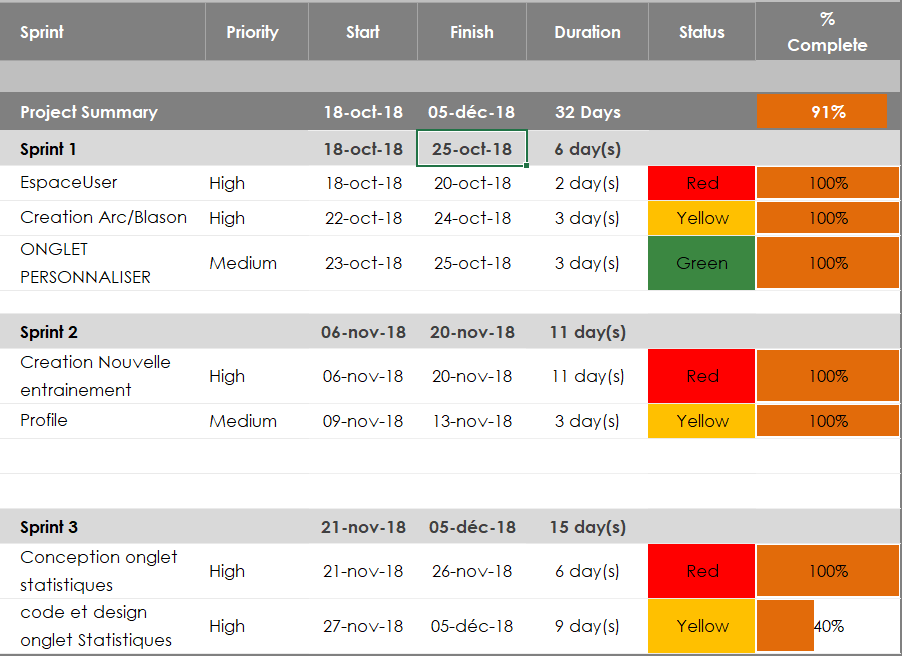
Afin de faciliter la gestion des risques, on a essayé de prévoir ce qu’allait nous poser un problème pendant les différents sprints en assignant un niveau de criticité en fonction de la probabilité-gravité-déterminabilité.



Ce tableau nous permettra ensuite de mieux nous organiser et à prévoir le temps nécessaire pour régler chaque problème lors de la planification.

III -Planning et exécution du projet :

Pour réaliser TANGU, on a opté pour une solution à 3 Sprints.

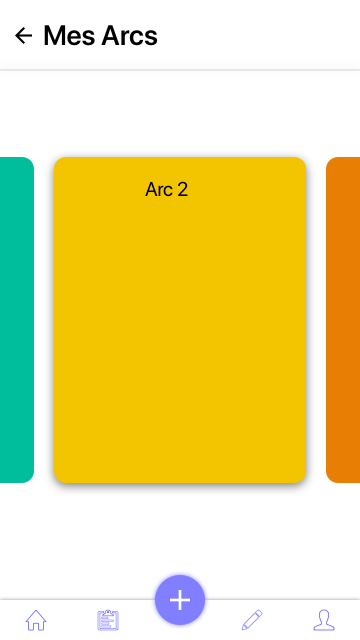
Voici comme indiqué dans le document comment ils vont se dérouler :

Le Sprint tracker explique donc qu’on aura 3 Sprints. Le premier débutera le 18 octobre et va durer 6 jours.

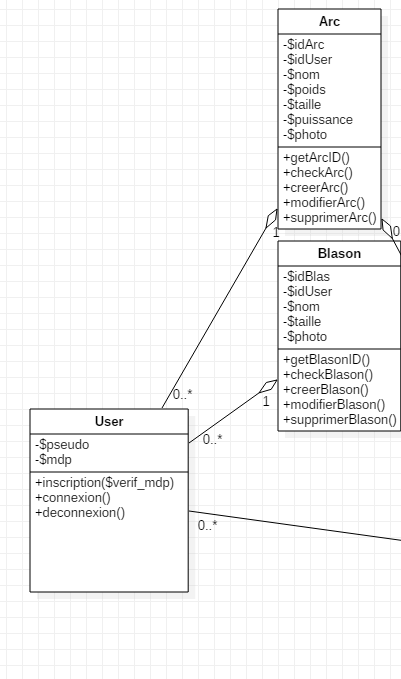
Le sprint 1 comprend :

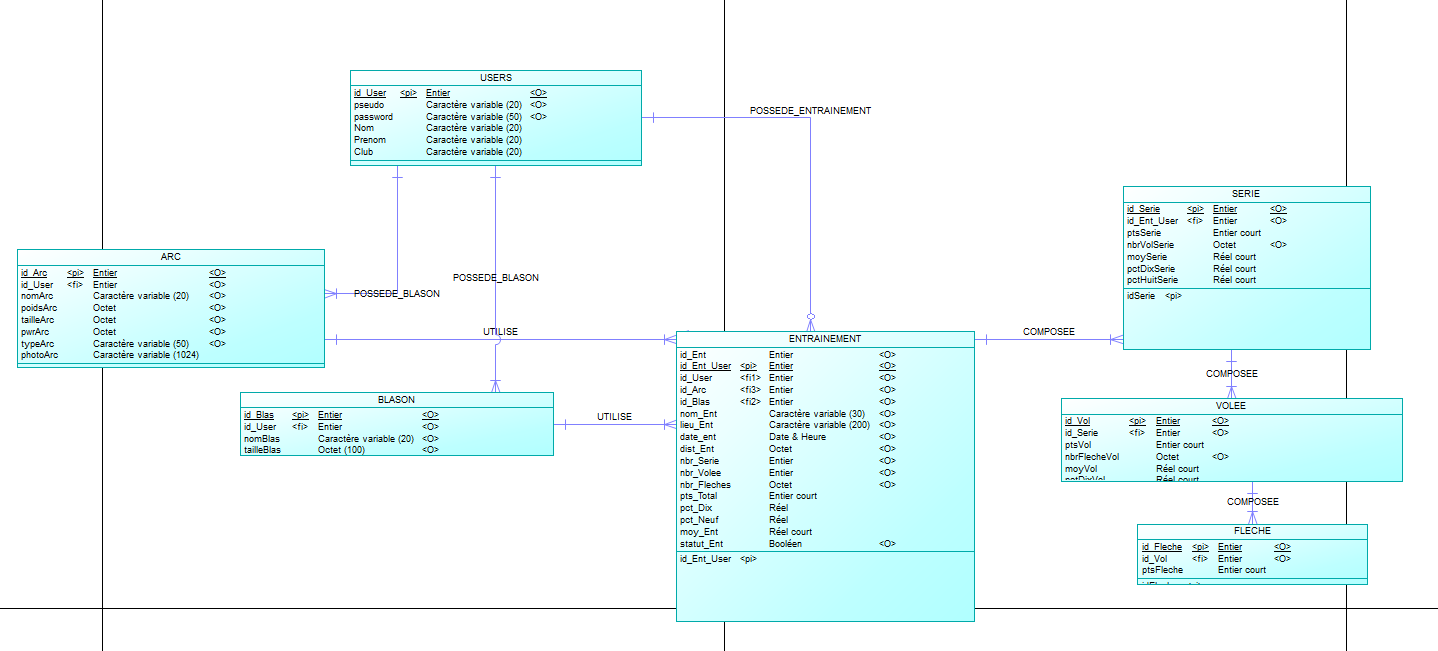
* La conception, programmation, documentation et testes de l’Espace User. C’est-à-dire qu’on sera capable de créer un utilisateur, ce qui implique une base de données bien définie. (Tache à réaliser en 2 jours.
* On pourra créer de nouveaux arcs, Blasons nécessaires à l’entrainement (tache à réaliser en 3 jours)
* -Finalement l’onglet PERSONNALISER regroupant tout ce qui a était fait précédemment (tache réalisée en 3 jours).

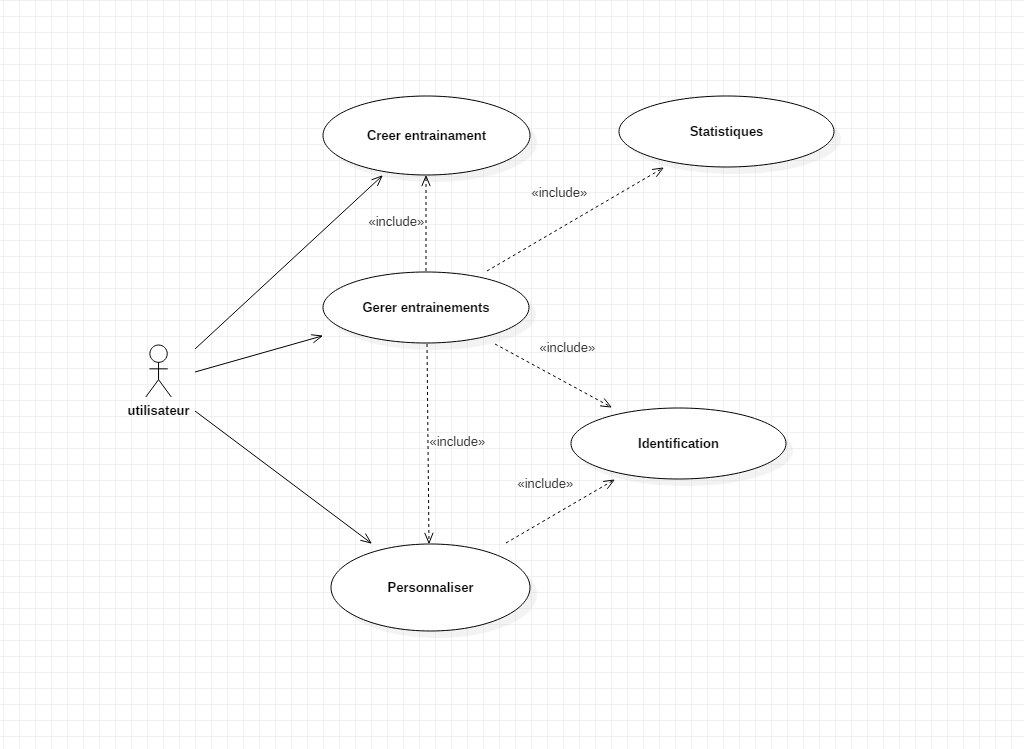
**La conception de ce sprint a donné ces schémas pour la partie Graphique :**



**Et également côté UML avec le diagramme de classes User, Arc et Blason :**



**On retrouve aussi le MLD de la Base de données** :

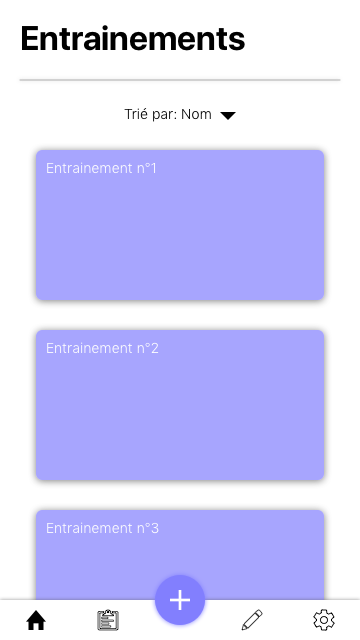
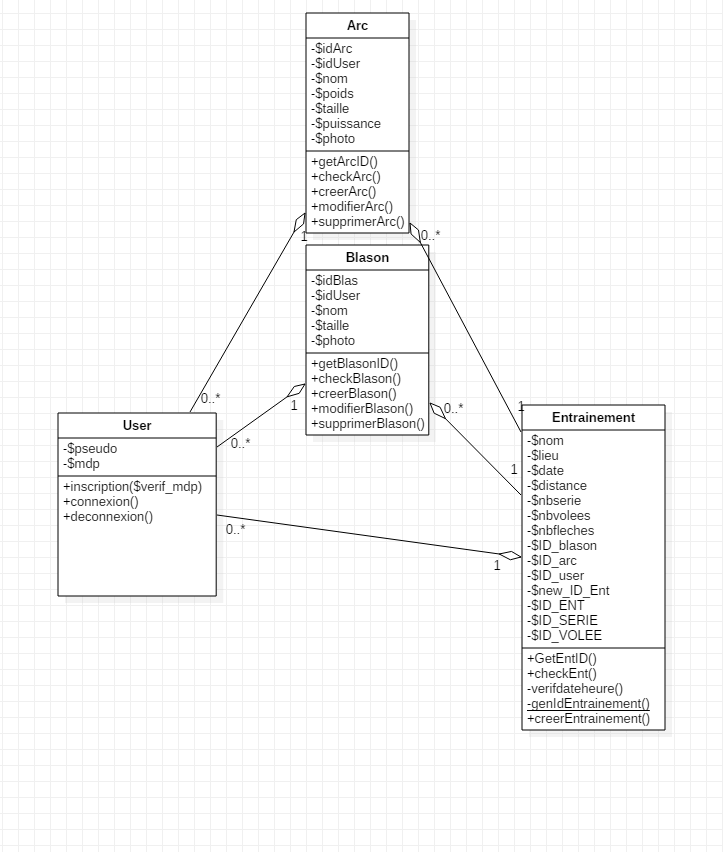
***Diagramme de cas d’utilisation*** :

Bien entendu, ces diagrammes/MLD sont susceptibles de changer ou pas avec le temps en fonction de nos besoins.

Le Sprint 2 était orienté partie création nouvel entrainement et son déroulement.

Cette partie consiste à ajouter un nouvel entrainement avec des caractéristique propres tel que le nom, la date, le matériel utilisé, distance, ainsi que le nombre de séries, volées et flèches.

Egalement, on a accès à une liste avec tous nos entrainements dans le passé. Et en consultant chaque entrainement, on peut voir nos scores/performances.

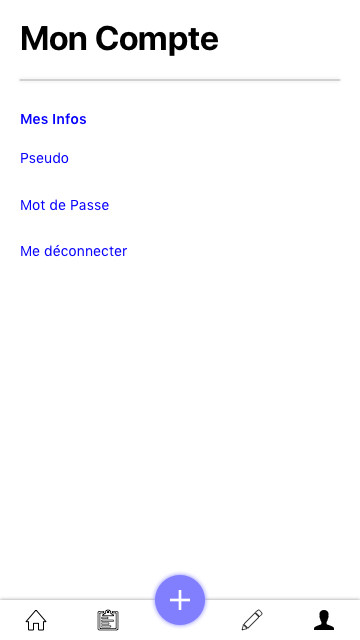


*Mise à jour du diagramme de classes de TANGU en ajoutant Entrainement.*

*Premier prototype de la liste des entrainements*

Egalement, Il y’a la partie profile avec la possibilité de changer son nom d’utilisateur et son mot de passe. Et aussi rajouter quelques détails optionnels comme le nom de son club.

Pour effectuer ces actions, il suffit d’utiliser les méthodes déjà crées dans la classe User par exemple.



Finalement, le 3ème Sprint est celui qu’on considère le plus long à faire.

En effet, 15 jours sont requis pour qu’on arrive à apporter une solution présentable au client, permettant de calculer et surtout modéliser ses performances.

La difficulté dans ce sprint réside dans comment manipuler des graphs avec pleins de variables et de conditions différents. En effet, nous n’avons pas eu de cours sur JavaScript, et surtout pour arriver à modéliser des fonctions.

Nous avons donc opté pour un Framework JavaScript en ligne nous permettant de créer des graphs avec des valeurs et paramètres données.

C’est d’ailleurs la raison pour laquelle on a prévue dans les risques ce genre de problèmes.

En prenant donc le temps d’analyser le fonctionnement de ce Framework et ses interactions avec notre Template et base de données, on a décidé de l’intégrer. Et comme prévoit la solution au manque de connaissances, nous avons pris quelques jours pour nous former, ce qui était bien évidemment inclus et prévue déjà dans la phase de planification

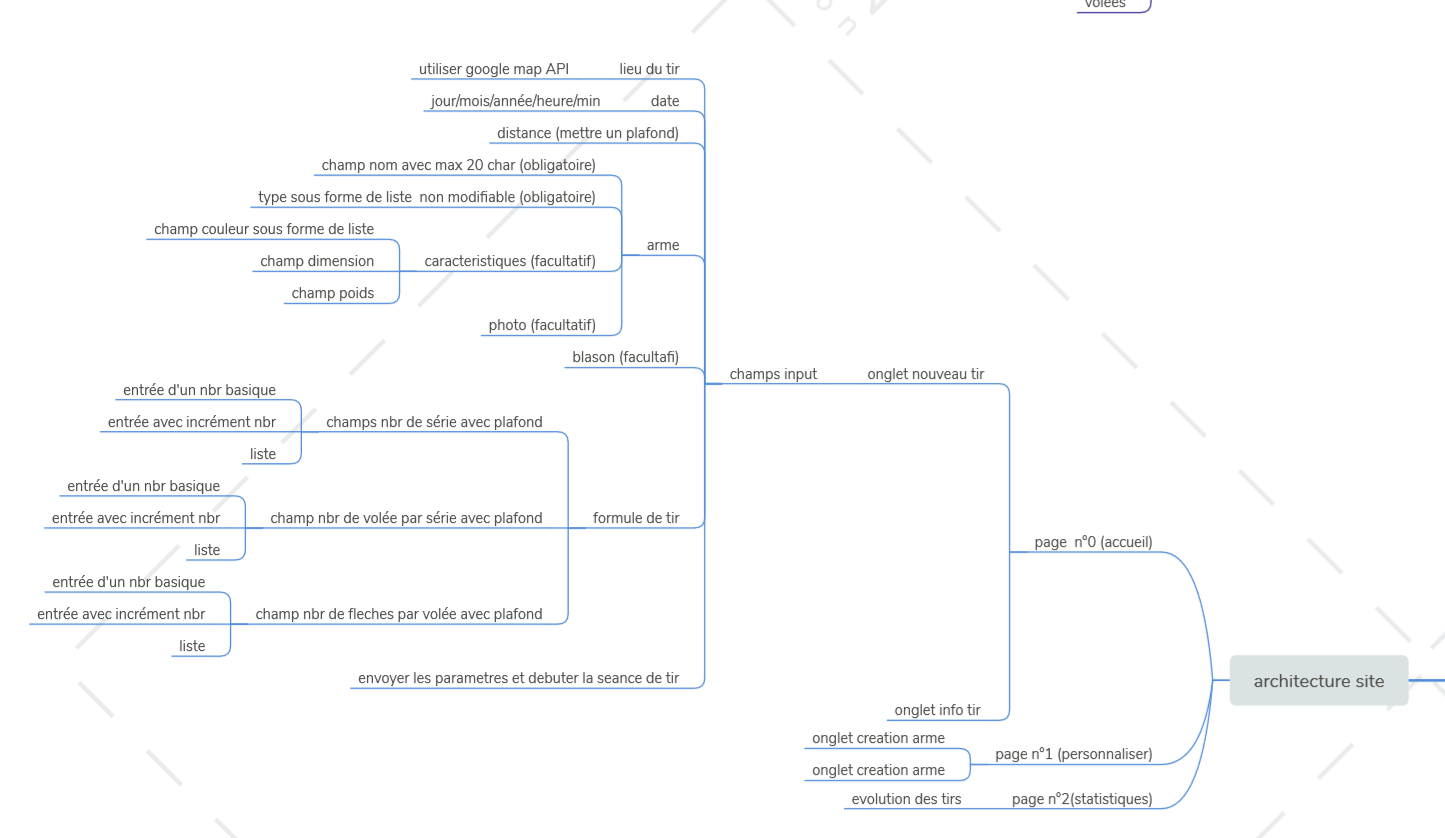
Voici donc quelques schémas montrant notre vision des graphs :

Bien que ce schéma soit minimaliste, il nous donne une idée très claire et précise sur comment va se dérouler ces calculs.

D’autres Documents annexes seront mis à disposition dès qu’on aura avancé sur ce sujet.

Finalement, on aura obtenu un site qui valide approximativement la maquette réalisé au début pendant la phase de planification et de préparation.

P.S : Certains options on étaient changées au cours de la phase de développement car ils ont un impact négatif sur la fluidité du site sur smartphone.



Vous retrouverez la suite de cette maquette dans les documents annexes.

IV- Documentation-Tests-Conventions de codage

1. Documentation :

La documentation Technique s’est faite à travers l’utilisation de PHP Storm.

Pour le moment, en raison de manque de temps, nous avons choisi de documenter la classe Arc et la classe blason.

On a commencé d’abord par la description des variables puis les méthodes avec leurs entrées/sorties respectifs. Puis finalement généré la PHP Doc en utilisant Dioxygène.

Pour la documentation utilisateur, on est parti pour faire un mode d’emploi et un guide rapide d’utilisation. Nous avons jugé que les autres documents ne sont pas nécessaires à TANGU. Il est assez intuitif et facile à manipuler.

Finalement, Une charte de confidentialité et un accord d’utilisation sont nécessaires pour répondre, comme signalé précédemment dans les risques, aux utilisations malveillantes de nos services.

*(On rappelle que tous les documents mentionnés sont disponibles dans les dossier annexes.)*

1. Tests :

Nous avons commencé par rédiger un diagramme de flot de la vérification des entrées. A partir de ce diagramme, nous avons rédigé les tests pour arc et blason en essaye toutes les possibilités de valeurs en entrée.

Nous avons néanmoins eu quelques problèmes avec l’installation de phpUnit. Une fois surmonté, les tests on étaient programmés sans autres problèmes majeurs.

1. Conventions de codage :

Nous avons appliqué les conventions de codage sur les classes que nous avons commenté (Arc et Blason).

On a appliqué les conventions de codage vus en génie logiciel :

* Nom de variables en anglais
* Ecriture droMaDaire ( 😉 )
* Pas de commentaires dans les méthodes
* Pas de méthodes de plus de 40 lignes.
* Pas de code en commentaire