

## Présentation des UE

### PM2 - Projet SAE-3 : Bowling Scorer

*D'ALMEIDA David, DELANIER Lucas, KARTAL Emre, PARANT Louison, VALIN Arthur*

#### UE 1 - Partir des exigences et aller jusqu'à une application complète

Nous allons développer une application mobile qui se nomme "Bowl'n". Elle prendra en compte le besoin client qui est de pouvoir enregistrer les scores des parties de bowling.

Nous allons permettre aux joueurs d'inviter leurs amis afin qu'ils puissent retrouver leurs parties ensemble ainsi que les résultats de ces dernières. L'utilisateur aura accès à ses statistiques qu'elles soient globales ou pour une partie précise.

Ce dernier sera libre de choisir si ses données seront partagées ou s'il les conservera localement. Il pourra ainsi envoyer ses données en différé quand et surtout s'il le souhaite. Cela fonctionne également quand l'utilisateur n'est pas connecté suite à un choix ou une perte.

Notre application essaiera de rester le plus ergonomique possible afin de privilégier l'expérience utilisateur.

#### UE 2 - Sélectionner les algorithmes adéquats pour répondre à un problème donné

Le besoin auquel nous devons répondre est de développer une application mobile, qui devra être disponible à la fois sur Android et IOS.

Nous avons donc décidé de choisir Flutter en tant que framework car il est open-source et est utilisé pour développer des applications cross-plateformes. De plus, il propose un grand nombre de bibliothèques et de widgets intégrés qui facilitent la création d'interfaces utilisateur modernes et personnalisées. Flutter est aussi une des technologies les plus performantes en terme d'application multiplateforme. Ce framework est également en pleine croissance grâce à une communauté et une utilisation grandissantes mais également une documentation de plus en plus fournie.

Nous avons aussi décidé de réaliser et d'intégrer une API qui devrait nous permettre d'effectuer les requêtes directement à la base de données. Ainsi, nous donnerions la responsabilité du côté serveur et non du côté client. Pour la base de données, nous avons choisi l'utilisation PostgreSQL car elle est compatible avec notre API et que nous l'avons beaucoup pratiqué en cours. Nous avons aussi pris l'initiative de développer plusieurs API afin de les comparer avec un benchmark pour identifier la plus performante.

Aussi, la documentation ayant une part importante dans la qualité de travail durant le projet, nous avons donc pensé à utiliser Docsaurus qui est un framework open-source pour la création de documentation statique. Il permet de générer des pages de documentation à partir de fichiers Markdown et de les rendre accessibles via une interface utilisateur simple. Il inclut des fonctionnalités telles que la recherche, l'accès à la table des matières, la navigation par onglets, et la possibilité de personnaliser l'apparence de la documentation. Docsaurus peut, entre autres, être utilisé pour créer des pages de documentation pour des projets de logiciels, des bibliothèques, des API, des guides de démarrage rapide et des manuels d'utilisation.

Lors de la réalisation d'une nouvelle tâche par un membre de l'équipe, les tests fonctionnels et unitaires seront réalisés à chaque push afin de vérifier que l'application reste toujours fonctionnelle que ce soit du côté client mais aussi du côté serveur. Ces tests se verront automatisés grâce à drone qui est un outil de CI/CD (intégration et déploiement continus) open-source. Ainsi, nous améliorerons la qualité du code en détectant les erreurs plus tôt dans le processus de développement.

### **UE 3 - Déployer des services dans une architecture réseau**

Nous allons développer une API respectant l'architecture RestFul afin de séparer notre logique métier du client. Cette API se basera sur le protocole réseau HTTPS pour optimiser la sécurité suivant le Elliptic Curve Cryptography (ECC). Toujours dans cette optique, l'accès à notre API nécessitera une clef de chiffrement.

Nous utiliserons aussi le protocole TCP à travers l'API de socket IO, une librairie nous permettant la création de "rooms" accessibles par chacun de nos utilisateurs.

Notre API effectuera la communication entre notre client et notre BDD PostGreSQL à travers un ORM afin de faciliter la liaison avec celle-ci. Aussi nous prendrons en compte l'architecture de notre API afin qu'elle puisse permettre l'extension à d'autres méthodes de persistance.

La base de données et l'API seront déployées dans un conteneur dans Docker afin que tout le monde puisse y avoir accès.

Afin d'avoir une Application toujours plus qualitative et maintenable, nous documenterons celle-ci avec Open Api via Swagger, qui nous permettra d'exposer les points d'entrée de notre API ainsi que leur fonctionnement.

Du côté client, nous gérerons également l'état de connexion de nos utilisateurs afin, d'une part, proposer une meilleure expérience mais aussi, d'autre part, assurer une cohérence des données.

### **UE 4 - Optimiser une base de données, interagir avec une application et mettre en oeuvre la sécurité**

Nous allons mettre en place une sauvegarde en base de données des résultats des parties pour chaque utilisateur. Nous allons utiliser des triggers pour mettre à jour automatiquement les statistiques de l'utilisateur après chaque nouvelle partie, comme son nombre de victoires, son nombre de parties jouées, son ratio, ...

Les informations de l'utilisateur qui seront stockés seront : son pseudo, son adresse mail et son mot de passe qui sera bien évidemment hacher (probablement avec Bcrypt).

Nous avons aussi anticipé l'utilisation d'index afin d'optimiser certaines requêtes qui peuvent être longues en cas de traitement de beaucoup de données.

### **UE 5 - Appliquer une démarche de suivi de projet en fonction des besoins métiers des clients et des utilisateurs**

Nous allons appliquer la méthode SCRUM qui est une méthode de gestion de projet qui permet de livrer rapidement des résultats concrets en s'adaptant constamment aux changements. Cela nous permettra de respecter les délais et de livrer les rapport de projet avec succès.

Nous allons utiliser des outils de gestion de projet tel qu'un Kanban, des tickets mais aussi un backlog des réunions récurrentes entre les membres de l'équipe et notre tuteur M.Chevaldonné dans le but de connaître l'avis de notre client et d'adapter notre application à ses besoins tout au long du projet. Nous allons aussi utiliser l'outil "Project" sur des machines de l'IUT afin de réaliser les différents diagrammes de suivi tel que le GANTT ou les réseaux PERT.

Nous allons aussi tenir à jour un wiki quotidiennement que nous découperons en semaines afin de renseigner l'implication de chacun ainsi que le temps de travail pour chaque tâche.

#### **UE 6 - Situer son rôle et ses missions au sein d'une équipe informatique**

Nous essayerons d'attribuer les tâches à chaque membre de façon à ce que cela corresponde aux compétences de chacun afin qu'ils puissent utiliser leurs connaissances et expériences pour contribuer de manière efficace au projet.

Nous réaliserons également des réunions et essayerons d'impliquer tout le groupe lors des décisions, pour donner à chaque membre la possibilité d'exprimer leur avis et de faire des suggestions.

Grâce au Kanban et à Codefirst, nous pourrons aussi observer directement l'avancée de chaque personne mais également voir la progression générale du projet. Grâce à cela, chacun pourra être attribué à une tâche rapidement et pourra travailler de manière plus efficace.