

Backlog Sprint 1

5 (jours de travail par semaine)*6(heures de travail par semaine)*4(semaines de travail pour sprint 1)=120j (Par membre)
120*6=720 h (pour tous les membres de groupe pour le sprint 1)

ID_F	USER STORY	ID Tâche	Tâche	Estimation (en heure)	Responsable
1	STORY LESS	A	Conception	5h	Toute l'équipe
		B	Implementation BD	2h	Toute l'équipe
2.1	En tant qu'utilisateur, je veux pouvoir m'inscrire et m'authentifier pour accéder aux fonctionnalités de l'application .	2.1.1	Créer le formulaire d'inscription (nom, prénom, email, mot de passe, etc.).	5h	Hbib
		2.1.2	Valider les champs du formulaire (ex : email valide, mot de passe fort, confirmation du mot de passe correspondante).	4h	
		2.1.3	Créer un fichier CSS pour styliser le formulaire.	4h	Hbib
		2.1.4	Implémenter la logique pour enregistrer les utilisateurs dans la base de données locale	5h	Hbib
		2.1.5	Hacher le mot de passe avant de l'enregistrer dans la base de données.	3h	
		2.1.6	Créer le formulaire de connexion (email + mot de passe).	5h	Hbib
		2.1.7	Implémenter la vérification des identifiants .	4h	Hbib
		2.1.8	Ajouter un message d'erreur en cas d'identifiants incorrects et une confirmation de connexion réussie.	3h	Hbib
2.2	En tant qu'admin, je veux pouvoir ajouter les utilisateurs .	2.2.1	Créer un formulaire d'ajout d'utilisateur (par l'admin).	4h	Hbib
		2.2.2	Créer un fichier CSS pour styliser le formulaire	4h	Hbib
		2.2.3	Implémenter la logique pour enregistrer	6h	Hbib

			l'utilisateur dans la base de donnée.		
		2.2.4	Ajouter un message de confirmation après l'ajout.	3h	Hbib
2.3	En tant qu'admin, je veux pouvoir supprimer les utilisateurs.	2.3.1	Créer un bouton "Supprimer" à côté de chaque utilisateur dans la liste.	5h	Hbib
		2.3.2	Implémenter la logique pour supprimer un utilisateur de la base	6h	Hbib
		2.3.3	Ajouter une confirmation avant la suppression.	3h	Hbib
		2.3.4	Mettre à jour l'interface après la suppression.	4h	Hbib
2.4	En tant qu'admin, je veux pouvoir modifier les informations des utilisateurs.	2.4.1	Créer une interface pour afficher la liste des utilisateurs avec un bouton "Modifier" pour chaque utilisateur.	6h	Hbib
		2.4.2	Créer un formulaire de modification pré-rempli avec les informations actuelles de l'utilisateur.	5h	
		2.4.3	Implémenter la logique pour mettre à jour les informations dans la base locale.	6h	Hbib
		2.4.4	Ajouter un bouton "Modifier" pour enregistrer les modifications.	3h	Hbib
		2.4.5	Ajouter une option d'annulation des modifications	4h	Hbib
2.5	En tant qu'admin, je veux pouvoir consulter la liste des utilisateurs ainsi que leurs informations .	2.5.1	Créer une interface affichant la liste des utilisateurs.	4h	Hbib
		2.5.2	Implémenter la logique pour récupérer la liste des utilisateurs depuis la base locale.	4h	Hbib
		2.5.3	Ajouter des options de tri et de recherche.	5h	Hbib
		2.5.4	Tester l'affichage	3h	Hbib
3.1	En tant que joueur, je veux pouvoir planifier un match en	3.1.1	Créer le formulaire de planification de match	3h	Nourhene

	définissant la date, l'heure et le type de jeu.	3.1.2	Créer un fichier css pour la planification du match.	3 h	Nourhene
		3.1.3	Créer le contrôleur "Planifier_match".	5 h	Nourhene
		3.1.4	Implémenter la logique pour sauvegarder la match en base de données	6 h	Nourhene
		3.1.5	Mettre en place un message de confirmation pour l'utilisateur	3h	Nourhene
3.2	En tant que joueur, je veux pouvoir participer à un match.	3.2.1	Créer le bouton "Participer" sur l'interface utilisateur.	3 h	Nourhene
		3.2.2	Créer un fichier CSS pour styliser le bouton "Participer".	3 h	Nourhene
		3.2.3	Créer le contrôleur "Participer_match" pour gérer la participation.	5 h	Nourhene
		3.2.4	Implémenter la logique pour ajouter le joueur en tant que participant dans la base de données.	6 h	Nourhene
		3.2.5	Après la participation, afficher un message confirmant la participation au match a été faite avec succès.	3 h	Nourhene
		3.2.6	Élaborer les tests unitaires pour valider la participation à un match.	5 h	Nourhene
3.3	En tant que joueur, je veux pouvoir consulter la liste des matchs.	3.3.1	Concevoir une interface qui affiche tous les matchs disponibles pour la consultation.	2 h	Nourhene
		3.3.2	Implémenter la logique pour récupérer la liste des matchs depuis la base de données.	3h	Nourhene
		3.3.3	Tester l'affichage de la liste des matchs sur l'interface.	4 h	Nourhene
		3.4.1	Créer l'interface qui permet à l'utilisateur de modifier	3 h	Nourhene

3.4	En tant que joueur, je veux pouvoir modifier les informations d'un match.		les informations d'un match.		
		3.4.2	Implémenter la logique pour mettre à jour les informations d'un match dans la base de données.	4 h	Nourhene
		3.4.3	Créer un bouton "Sauvegarder" de sauvegarde pour permettre à l'utilisateur de valider ses modifications.	2 h	Nourhene
		3.4.4	Implémenter la logique pour valider les changements.	4 h	Nourhene
		3.4.5	Créer une fonctionnalité d'annulation des modifications.	3 h	Nourhene
3.5	En tant que joueur, je veux pouvoir annuler un match.	3.5.1	Ajouter un bouton "Annuler" pour chaque match.	4 h	Nourhene
		3.5.2	Implémenter la logique pour annuler un match.	4 h	Nourhene
		3.5.3	Mettre à jour l'interface après annulation.	5 h	Nourhene
		3.5.4	Ajouter une confirmation avant l'annulation.	3 h	Nourhene
		3.5.5	Créer un test unitaire pour valider l'annulation du match.	4 h	Nourhene
3.6		3.6.1	Ajouter la recherche des matchs par différents critères	4h	Nourhene
		3.6.2	Implémenter le tri des matchs selon divers paramètres	3h	Nourhene
		3.6.3	Mettre en place des filtres avancés pour afficher les résultats.	3h	Nourhene

		3.6.4	Intégrer une API de mailing pour les notifications et rappels.	6h	Nourhene
		3.6.5	Ajouter une API de cartes pour la localisation des événements.	6h	Nourhene
4.1	En tant qu'utilisateur premium, je veux pouvoir demander un régime personnalisé .	4.1.1	Créer un formulaire de demande de régime personnalisé.	4 h	Mariem
		4.1.2	Créer un fichier CSS pour le formulaire	4 h	Mariem
		4.1.3	Implémenter la logique pour enregistrer la demande en base de données.	6 h	Mariem
		4.1.4	Mettre en place un message de confirmation pour l'utilisateur.	3 h	Mariem
4.2	En tant que nutritionniste, je veux pouvoir ajouter des régimes alimentaires aux utilisateurs .	4.2.1	Créer le formulaire d'ajout de régime alimentaire.	5 h	Mariem
		4.2.2	Créer un fichier CSS pour le formulaire.	5 h	Mariem
		4.2.3	Créer UN BOUTON Ajouter Régime pour gérer l'ajout des régimes.	5 h	Mariem
		4.2.4	Implémenter la logique pour enregistrer le régime en base de données.	6 h	Mariem
4.3	En tant que nutritionniste, je veux pouvoir modifier les régimes alimentaires des utilisateurs.	4.3.1	Créer l'interface de modification des régimes.	5 h	Mariem
		4.3.2	Implémenter la logique pour mettre à jour les	6 h	Mariem

			informations dans la base de données.		
		4.3.3	Créer un bouton "Modifier" pour valider les modifications.	2 h	Mariem
		4.3.4	Implémenter la logique de validation des changements.	4 h	Mariem
		4.3.5	Ajouter une option d'annulation des modifications.	5 h	Mariem
4.4	En tant que nutritionniste, je veux pouvoir supprimer les régimes alimentaires des utilisateurs .	4.4.1	Ajouter un bouton "Supprimer" .	3 h	Mariem
		4.4.2	Implémenter la logique pour supprimer un régime de la base de données.	5 h	Mariem
		4.4.3	Ajouter une confirmation avant la suppression.	5 h	Mariem
4.5	En tant que nutritionniste, je veux recevoir une notification lorsqu'un joueur passe en compte premium.	4.5.1	Détecter l'activation du compte premium par le joueur	2 h	Mariem
		4.5.2	Ajouter la notification dans la base de données locale	5 h	Mariem
		4.5.3	Créer une interface pour consulter les notifications	5 h	Mariem
		4.5.4	Ajouter une alerte visuelle ou sonore lors de la réception d'une notification	3 h	Mariem
4.6	En tant que joueur, je veux uploader une photo de mon plat pour que l'IA analyse sa compatibilité avec mon régime alimentaire.	4.6.1	Créer une interface pour que le joueur puisse importer une photo	5 h	Mariem
			Ajouter un système de vérification du format (JPEG, PNG) et de la taille	3 h	Mariem
		4.6.2	Utiliser une bibliothèque ou API (ex: TensorFlow , YOLO , ou OpenCV) pour	5 h	Mariem

			identifier les aliments présents sur la photo		
		4.6.3	Vérifier si le plat est conforme aux recommandations du nutritionniste.	4 h	Mariem
		4.6.4	Afficher le diagnostic de l'IA	6 h	Mariem
		4.6.5	Gérer les cas d'erreurs	2 h	Mariem

5.1	En tant qu'admin, je veux pouvoir ajouter des événements sportifs .	5.1.1	Créer une interface FXML avec un formulaire pour ajouter un événement	3h	Eya
		5.1.2	Créer un fichier CSS pour styliser le formulaire et rendre l'interface intuitive.	3h	Eya
		5.1.3	Implémenter la logique pour sauvegarder l'événement en base de données	5h	Eya
		5.1.4	Ajouter les contrôles de saisie (ex : date valide, nom obligatoire).	4h	Eya
		5.1.5	Tests unitaires pour valider l'ajout d'événements avec des données correctes.	4h	Eya
5.2	En tant qu'admin, je veux pouvoir modifier des événements sportifs .	5.2.1	Ajouter un bouton "Modifier" pour chaque événement	2h	Eya
		5.2.2	Charger les données de l'événement sélectionné dans le formulaire.	5h	Eya
		5.2.3	Implémenter la mise à jour des informations en base de données.	4h	Eya
		5.2.4	Ajouter des messages d'erreur explicites en cas de saisies incorrectes ou données manquantes.	3h	Eya

		5.2.5	Implémenter des tests unitaires pour valider la mise à jour des événements et la gestion des erreurs.	4h	Eya
5.3	En tant qu'admin, je veux supprimer des événements sportifs .	5.3.1	Ajouter un bouton "Supprimer" pour chaque événement avec confirmation avant suppression.	4h	Eya
		5.3.2	Créer une boîte de dialogue modale de confirmation pour éviter les suppressions accidentelles.	4h	Eya
		5.3.3	Implémenter la logique de suppression dans le contrôleur et vérifier les contraintes liées à la base de données.	5h	Eya
		5.3.4	Ajouter des messages de succès ou d'erreur en fonction du résultat de la suppression	3h	Eya
		5.3.5	Élaborer des tests unitaires pour valider la suppression des événements sportifs et la gestion des erreurs.	3h	Eya
5.4	En tant qu'admin, je veux pouvoir consulter la liste des événements sportifs .	5.4.1	Créer une interface utilisateur qui affiche la liste des événements sportifs.	4h	Eya
		5.4.2	Implémenter la récupération des événements depuis la base de données et les afficher dans une liste.	5h	Eya
		5.4.3	Ajouter des filtres (par date, nom d'événement, etc.) pour améliorer la recherche dans la liste.	5h	Eya
		5.4.4	Ajouter des indicateurs visuels pour chaque événement (ex. : complet,...).	6h	Eya

		5.4.5	Élaborer des tests unitaires pour valider la bonne récupération et affichage des événements.	4h	Eya
5.5	En tant que joueur, je veux pouvoir m'inscrire à un événement .	5.5.1	Créer une interface utilisateur avec un bouton "S'inscrire" pour chaque événement disponible.	2h	Eya
		5.5.2	Implémenter la logique d'inscription : lorsque le joueur clique sur "S'inscrire", enregistrer l'inscription dans la base de données.	5h	Eya
		5.5.3	Créer des messages de validation pour l'inscription : succès, échec si l'événement est complet ou déjà inscrit.	2h	Eya
		5.5.4	Élaborer des tests unitaires et tests d'interface pour vérifier la logique d'inscription et de disponibilité des places.	3h	Eya
5.6	En tant que jouer je veux chercher ,Trier ,Filtrer et consulter map	5.5.5	Ajouter une barre de recherche pour filtrer les événements par date	2h	Eya
		5.5.6	Implémenter des options de tri (par date, popularité, etc.).	4h	Eya
		5.5.7	Ajouter des filtres avancés (par catégorie, statut, disponibilité, etc.).	4h	Eya

		5.5.8	Intégrer une API de cartographie (Google Maps, Leaflet...) pour afficher la localisation des événements	6h	Eya
6.1	En tant qu'admin, je veux pouvoir ajouter des produits à la boutique .	6.1.1	Créer le formulaire d'ajout de produit(fxml)	4h	Sabee
		6.1.2	Développer le contrôleur pour ajouter un produit	4h	Sabee
		6.1.3	Ajouter les règles de validation des données (contrôle de saisie)	3h	Sabee
		6.1.4	Tester l'ajout d'un produit	3h	Sabee
6.2	En tant qu'admin, je veux pouvoir modifier des produits à la boutique .	6.2.1	Créer le formulaire de modification	4h	Sabee
		6.2.2	Développer le contrôleur pour modifier un produit	4h	Sabee
		6.2.3	Gérer le contrôle de saisie pour éviter des erreurs	3h	Sabee
		6.2.4	Tester la modification d'un produit	3h	Sabee
6.3	En tant qu'admin, je veux pouvoir consulter la liste des produits à la boutique .	6.3.1	Créer la page de catalogue produit	4h	Sabee
		6.3.2	Développer la récupération et l'affichage des produits avec leurs images	4h	Sabee
		6.3.3	Tester la liste des produits	3h	Sabee
6.4	En tant qu'admin , je veux pouvoir voir la statistique des produits les plus commandes	6.4.1	Créer une interface pour afficher les statistiques	5h	Sabee
		6.4.2	Implémenter la logique pour récupérer les données dans le backend	6h	Sabee
		6.4.3	Intégrer un graphique (piechart) pour visualiser les produits les plus commandés	4h	Sabee
		6.4.4	Tester la récupération des données depuis la base de données.	2h	Sabee

		6.4.5	Tester l'affichage des statistiques	2h	Sabee
6.5	En tant qu'admin, je veux pouvoir supprimer des produits à la boutique .	6.5.1	Ajouter le bouton de suppression sur l'interface de la liste	3h	Sabee
		6.5.2	Développer la logique de suppression dans le contrôleur	4h	Sabee
6.6	En tant qu'utilisateur, je veux pouvoir consulter les produits disponibles .	6.6.1	Créer la page de catalogue produit	4h	Sabee
		6.6.2	Développer la récupération et l'affichage des produits , si disponible affichage en stock sinon rupture de stock	4h	Sabee
6.7	En tant qu'utilisateur, je veux pouvoir rechercher des produits à la boutique .	6.7.1	Conception d'une barre de recherche dans l'interface utilisateur	3h	Sabee
		6.7.2	Implémentation de la fonctionnalité de recherche dans le controller (recherche par nom	4h	Sabee
6.8	En tant qu'utilisateur, je peux commander des produits en ligne.	6.8.1	Créer l'interface de commande	4h	Sabee
		6.8.2	Ajouter les règles de validation pour la commande (contrôle de saisie)	3h	Sabee
		6.8.3	Affichage d'un message de confirmation ou d'erreur après la commande	3h	Sabee
7.1	En tant qu'utilisateur, je peux ajouter une réclamation.	7.1.1	Créer un formulaire permettant à l'utilisateur de soumettre une réclamation avec des champs nécessaires (description, type, etc.).	3h	amine
		7.1.2	Implémenter la logique pour enregistrer la réclamation dans la base de données après soumission.	3h	amine
		7.1.3	Implémenter des messages d'erreur pour les champs manquants ou incorrects.	4h	amine

7.2	En tant qu'utilisateur, je peux modifier les informations d'une réclamation.	7.2.1	Créer une interface permettant à l'utilisateur de visualiser et de modifier ses réclamations existantes.	6h	amine
		7.2.2	Implémenter la logique pour récupérer les informations de la réclamation et les afficher dans le formulaire de modification.	6h	amine
		7.2.3	Enregistrer les modifications apportées dans la base de données.	5h	amine
		7.2.4	Tester la modification d'une réclamation.	5h	amine
7.3	En tant qu'utilisateur, je peux supprimer une réclamation.	7.3.1	Ajouter un bouton "Supprimer" à côté de chaque réclamation dans l'interface utilisateur.	5h	amine
		7.3.2	Gérer le contrôleur pour supprimer une réclamation.	5h	amine
		7.3.3	Tester la suppression d'une réclamation.	5h	amine
7.4	En tant qu'administrateur, je peux traiter les réclamations.	7.4.1	Créer une interface d'administration pour afficher la liste des réclamations à traiter.	4h	amine
		7.4.2	Gérer le contrôleur pour traiter (approuver ou refuser) une réclamation.	4h	amine
		7.4.3	Ajouter un champ pour que l'administrateur puisse ajouter des commentaires ou des actions spécifiques liées à chaque réclamation.	6h	amine
		7.4.4	tester le traitement des réclamations.	6h	amine

8	STORY LESS	f	Diagramme de Classe d'analyse	5h	Tout l'équipe
		J	Diagramme de Séquence objet	5h	Tout l'équipe
		H	Elaboration de Backlog Sprint 1	5h	Tout l'équipe
		K	Integration tout les taches	6h	Tout l'équipe

