



IUT Nantes

Pôle Sciences et technologie

Nantes Université

Nantes Université - IUT de Joffre - BUT Informatique

**Rapport de la SAE501
Développement avancé**

R5.Real | Rapport global

sae5-2425-groupe2-medok

Init : OULMAS Thomas, FRÉMONT Tom, MOREAU--THOMAS Nils

Alt : HEQUET Nathan, PRESTI Louis, RAGEAU Nils

3ème année de BUT Informatique - SAE du 1er semestre

Table des matières

Technologies & Organisation.....	3
Évolution.....	5
Projection.....	5
Annexes.....	6
Trello.....	6
Issue Board.....	6
Labels.....	7
Google Sheet.....	7
Statistiques.....	8
Maquettes de l'application.....	15

Technologies & Organisation

Nous avons donc réalisé ce projet en Kotlin, en utilisant Android Studio, car nous étions familier avec cela. Le problème étant que cela restreint l'application aux téléphones Android uniquement ce qui fait que le côté multi-plateforme désiré n'est pas possible dans notre cas.

l'OCR utilisé pour lire les ordonnances est `mlkit` de Google, la raison étant que Tesseract (un autre choix pour l'OCR) est certes open-source, mais il est aussi plus compliqué à manipuler. mais il vaut mieux quelque chose de fonctionnel qu'open-source.

Pour la base de données, nous avons choisi un fonctionnement hybride :

- En local pour les profils et les données de l'application lié à ceux-ci.
- Sur serveur pour les données des médicaments et des pharmaciens

Tout mettre en local serait très coûteux en mémoire à cause de la quantité massive de données, néanmoins notre solution actuelle ne permet pas un fonctionnement hors connexion. L'application fonctionne quand même sans connexion, mais avec des fonctionnalités comme l'annuaire des pharmaciens désactivées. Voici un tableau comparatif pour chaque côté :

Solution	+ Avantages	- Inconvénients
Entièrement sur Serveur	Réduit la place prise par l'application sur l'appareil de l'utilisateur Facilite la mise à jour des parseur de PDF et de données (car nous avons la main dessus)	Non-respect de la confidentialité des profils utilisateurs Doit toujours être actif et donc maintenu par nous Connexion requise
Entièrement en Local	Fonctionne hors connexion Respect de la confidentialité des profils utilisateurs	Complexifie la mise à jour des parseur de PDF et de données (car l'utilisateur doit mettre à jour son application) L'application prendra plus de place sur l'appareil de l'utilisateur
Hybride	Respect de la confidentialité des profils utilisateurs Réduit la place prise par l'application sur l'appareil de l'utilisateur (contient uniquement les profils et leurs données associées) Facilite la mise à jour des parseur de PDF et de données (car nous avons la main dessus)	Le serveur doit toujours être actif et donc maintenu par nous Ne fonctionne pas à 100% hors connexion

Nous avons créé un compte sur notre VM pour le serveur, qui contient diverses données et api, et pour le Jenkins, néanmoins ce dernier n'a pas été utilisé par manque de temps et de tests (ceux-ci sont arrivés tard).

Pour s'organiser tout au long du projet, nous avons commencé par utiliser **Trello***, un outil à base de tableaux Kanban, car certains membres de notre équipe étaient familiers avec celui-ci. Cependant après réflexion nous nous sommes dirigés sur **l'Issue Board*** de Gitlab.

À chaque issue, il y a une branche, les changements la concernant seront donc faits là-bas, ensuite après validation de la merge request sur main l'issue est déclarée comme "Closed" et la merge request effectuée.

Nous avons utilisé les milestones de gitlab afin de savoir l'avancé global du projet et dans quel catégorie se situait telle ou telle issue. Le milestone "Version de base" contient ce qui est demandé par l'application, "Version améliorée" contient ce qui sera ajouté à posteriori.

Les **labels du Gitlab*** ont été utilisés pour déterminer sur une Issue :

- Sa priorité (très haute, haute, moyenne, faible)
- Son statut (En cours, À valider, Abandonné)
- Si c'est un bug
- Sa catégorie (BD, UI/UX, Appli mobile)

Nous avons scindé le projet en 3 repository Gitlab : OCR, server et app.

Le repo OCR est un repo de test qui servait à créer l'OCR, mais ne sachant pas s'il était possible de le faire ou non, un git à part entière a été créé pour ne pas interférer avec le git de l'application mobile.

Le repo app est le repo qui contient l'application mobile, avec la branche main pour la production et les branches issues comme des branches dev.

Le repo server est le repo qui contient le serveur, soit l'API qui permet de récupérer diverses données (sur les pharmaciens par exemple) et aussi un début de parseur du PDF d'interaction médicamenteuse.

Nous avons appliqué la méthode SCRUM où chaque Sprint avait une durée variable selon la charge de travail de la semaine, si il n'y avait pas beaucoup d'avancement global dans la semaine alors celui-ci était étendu d'une unique semaine.

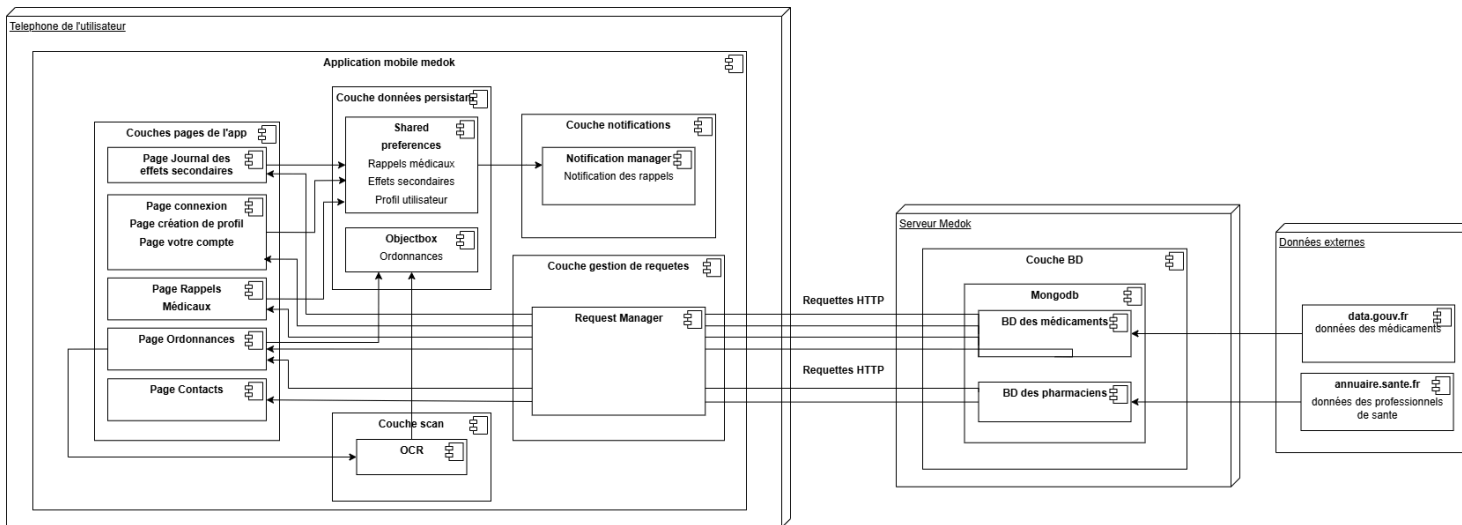
Les sprints durent entre 1 et 2 semaines, les vacances scolaires n'étant pas comprises.

Les réunions avaient lieu chaque fin de semaine ou nous nous réunissions sur Discord pour faire un appel vocal groupé, nous avons choisi Discord car tout le monde dans l'équipe l'utilise.

Les avancées de chacun étaient écrites dans un **Google Sheet***, celui-ci à évoluer au fil du temps au vu des différents besoins (marquer une tâche comme abandonnée, avoir plus de lisibilité...)

Le SCRUM Master et le Product Owner varient pour chaque sprint afin de ne pas être las et que chacun puisse animer les réunions.

Et voici comment le projet était organisé au niveau des couches : [Diagramme de déploiement de l'application Medok](#)



Évolution

Comme mentionné précédemment, nous sommes passés de Trello à Gitlab, et notre Compte rendu à évolué au fil du temps pour le rendre plus pratique.

Pour ce qui est du Discord et du Gitlab, nous avons des **statistiques*** globales et pour chaque utilisateurs disponibles en annexes.

Nous avons fait une première **maquette de l'application*** cependant celle-ci n'était pas très claire et belle, alors une **deuxième maquette*** a été faite qui clarifie tous les besoins de l'application et avec un meilleur rendu, ci qui a motivé l'équipe à travailler.

Projection

Pour améliorer l'efficacité du projet, nous aurions dû imposer un temps minimum travaillé sur le projet par semaine et communiquer clairement sur les blocages.

Pour améliorer la cohésion de l'équipe, écrire les conventions de nommage et hiérarchiques dès le début du projet aurait été d'une grande aide.

Nous aurions dû aussi utiliser le Time Tracking de gitlab pour avoir une meilleure visibilité sur quelles tâches l'équipe bloquent, ou si la tâche a été sous-estimée.

Pour avoir un fonctionnement garanti, il aurait fallu écrire les tests au fur et à mesure et non vers la fin du projet.

Pour avoir un départ vif, les technologies auraient dû être établies dès le départ et non pendant deux mois entiers.

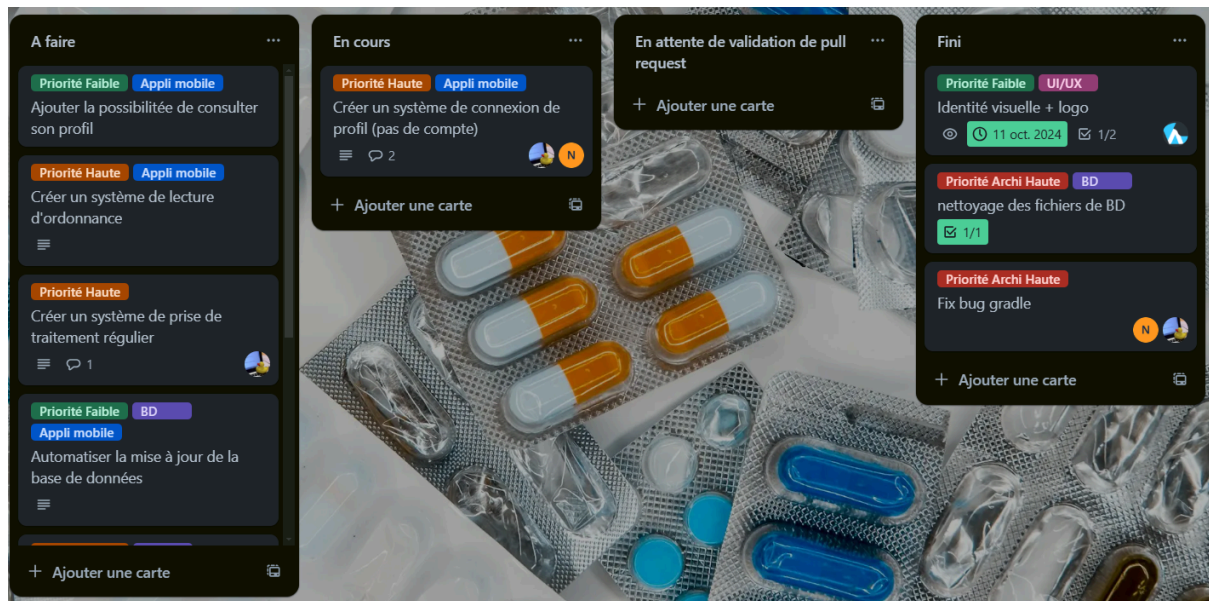
Pour finir, nos objectifs aurait pû être moins exigeant en vue de la contrainte de temps et notre manque de pratique sur certains points.

Dans le futur, l'application aura un Jenkins pour exécuter les tests automatiquement et garantir un bon fonctionnement.

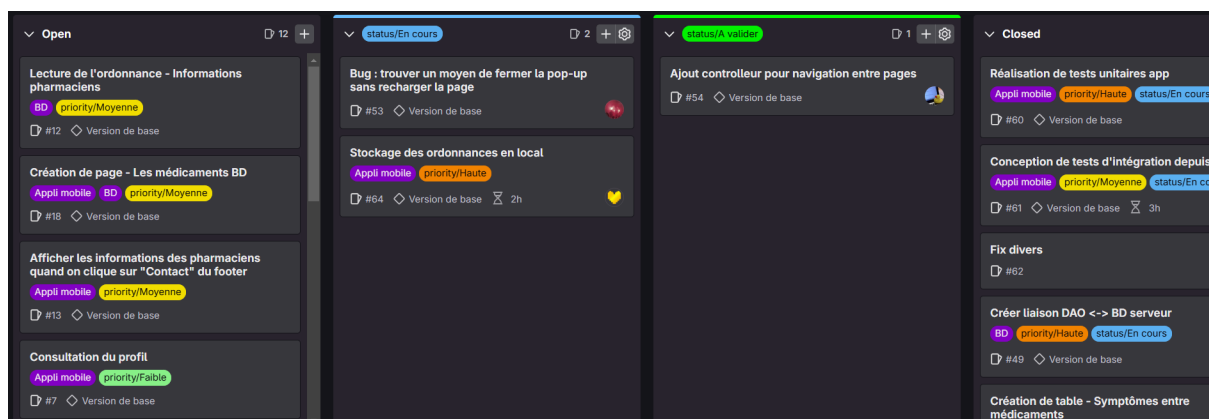
Les interactions médicamenteuses, l'annuaire de contact des pharmaciens et la gestion de profils utilisateurs seront aussi implémentés et l'application sera multi-plateforme.

Annexes

Trello



Issue Board



Labels

Appli mobile sae5-2425-groupe2-medok / app	priority/Faible sae5-2425-groupe2-medok / app	
BD sae5-2425-groupe2-medok / app	priority/Haute sae5-2425-groupe2-medok / app	status/A valider sae5-2425-groupe2-medok / app
Bug sae5-2425-groupe2-medok / app	priority/Moyenne sae5-2425-groupe2-medok / app	status/Abandonné sae5-2425-groupe2-medok / app
UI/UX sae5-2425-groupe2-medok / app	priority/Très haute sae5-2425-groupe2-medok / app	status/En cours sae5-2425-groupe2-medok / app

Google Sheet

1ère version du compte rendu

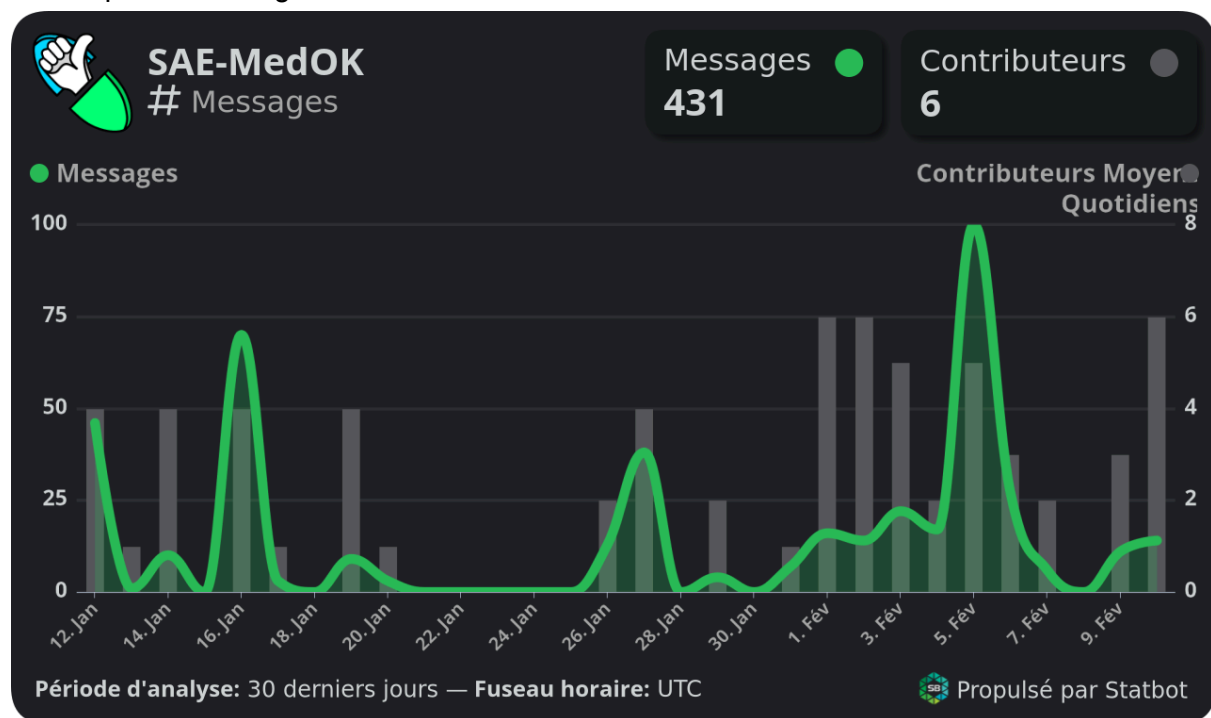
	Compte-Rendu de fin de sprint du 07/10 au 13/10				
	Product Owner :	Louis P.	SCRUM Master :	Nils MT.	
	Version :	0.0.0	Release fonctionnelle :	Oui	pas de bug & projet vide
	Tâches en cours				Tâches terminées
					Retour client
Louis P.	- #15 créer la base de donnée (les relations)				- #6 nettoyage des données fait et mis sur git et au bon encodage (les scripts de nettoyages sont dispo dans le git) - restructure du projet (pour avoir android dans un dossier)
Nathan H.	- #2 création des pages du mdp oublié				
Nils MT.	- #14 dialogue de création et de modification de rappel				- #14 vue d'affichage des rappels - #4 création de l'identité visuelle - aide à la création et l'organisation du figma
Nils R.	- #5 création des vues de connexion et creation de compte				- setup des issues (pour que Thomas puissent les utilisées)
Thomas O.					- transfert des tâches de trello vers les issues de gitlab
Tom F.	- #14 optimisation des pages (réduire le nombre) - #14 réactions médicamenteuses				- #14 journal effet secondaire, - #14 messagerie avec les pro de la santé - #14 pages de connexion - #14 médicament enregistré/catalogue de médoc
	Bug fix en cours				Bug fix terminées
					Retour client
Louis P.					
Nathan H.					- #3 gradle qui marchais pas
Nils MT.					
Nils R.					- #3 gradle qui marchais pas - les gitignore qui étais pas bon avec la restructure du projet
Thomas O.					
Tom F.					
	Notes additionelles				
	Prochain sprint : 1 semaines				
	Décision sur la BD : SQLite (parce que gratuit, customisable et le SQL on connait, et utilisable localement hors-ligne)"				

2ème version du compte rendu

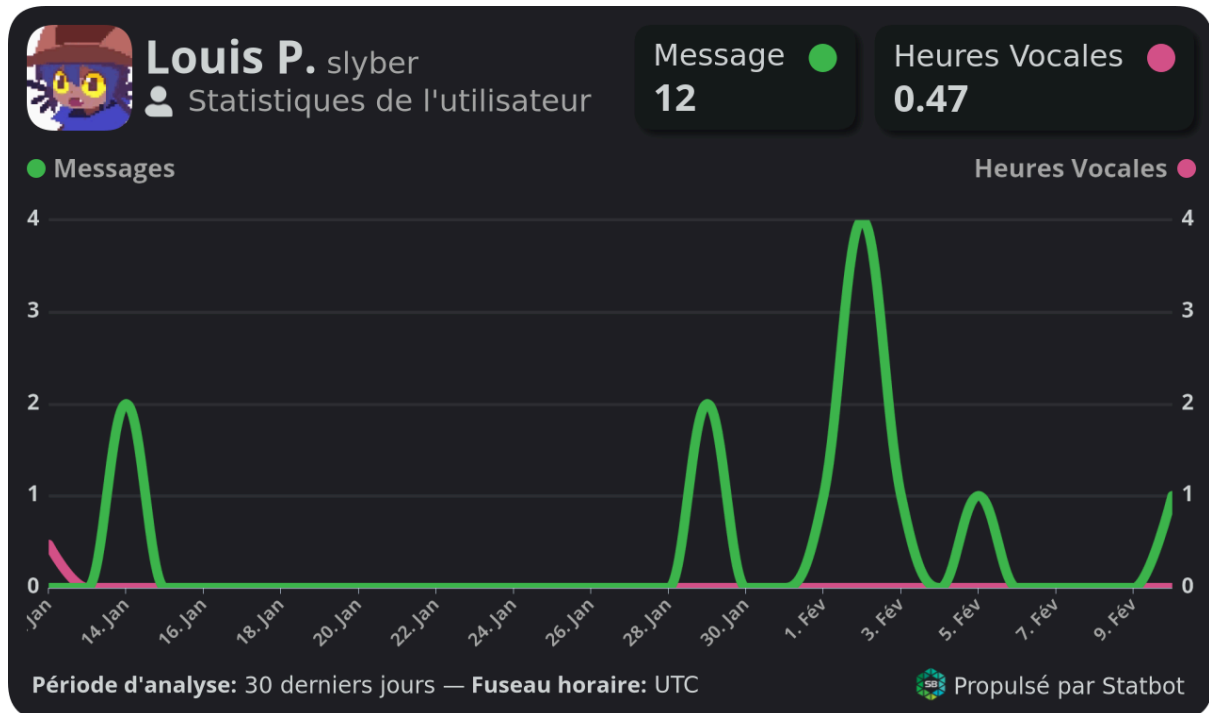
Compte-Rendu de fin de sprint du 03/02 au 09/02					
Product Owner :	Tom F.	SCRUM Master :	Thomas O.	Next Scrum Master :	Tom F.
Version :	0.0.0	Release fonctionnelle :	Oui	Next Product Owner :	Thomas O.
Membre	Type	Status	Coté	Description	Note
Tom F.	Tâche	Terminé	app	Faire un diagramme de déploiement et de composants & connecteurs	
Tom F.	Tâche	Terminé	app	Faire des tests coté app	
Tom F.	Tâche	Étendu sur ce sprint	app	42 - Faire la page des contacts	
Tom F.	Tâche	En cours	app	63 - Test de model Email et PhoneNumber	
Thomas O.	Tâche	Terminé	app	61 - Conception de tests pour vérifier l'intégration de l'API sur l'appli	Des tests avec un serveur mocké + avec le vrai serveur s'il est online
Thomas O.	Tâche	Terminé	app	64 - Faire sauvegarder les ordonnances en local	
Thomas O.	Tâche	Étendu sur ce sprint	app	Faire transvaser le stockage de données à un format de données plus adapté comme objectbox	Délai causé par la conception de tests qui était au final plus importante dans l'immédiat
Nils R.	Tâche	Étendu sur ce sprint	app	Implémentation des alertDialog dans les controllers	Les alternants seront en entreprise
Nils MT.	Bug	Terminé	app	62 - Fix divers et variés	
Nils MT.	Tâche	Terminé	autres	Faire diapo présentation appli	
Nils MT.	Tâche	Terminé	autres	Rapport d'économie	
Nils MT.	Tâche	Étendu sur ce sprint	app	Mise en place de la communication côté application vers la BD sur les DAOs restant	
Nils MT.	Tâche	Étendu sur ce sprint	server	Mettre serveur sur VM	+ adapté l'appli (RequestManager) en conséquence
Nils MT.	Tâche	En cours	autres	Vidéo démo app	
Nils MT.	Tâche	En cours	autres	Rapport de qualité de dev	reste la partie à Thomas
Nils MT.	Tâche	En cours	autres	Rapport de dev avancé	
Nils MT.	Tâche	En cours	autres	Rapport global	
Nathan H.	Tâche	Étendu sur ce sprint	app	(42 - app) - Création d'un utilisateur en format json	Les alternants seront en entreprise
Louis P.	Tâche	Étendu sur ce sprint	server	Rassembler les fonctions des DAO dans le DAO abstract	Les alternants seront en entreprise

Statistiques

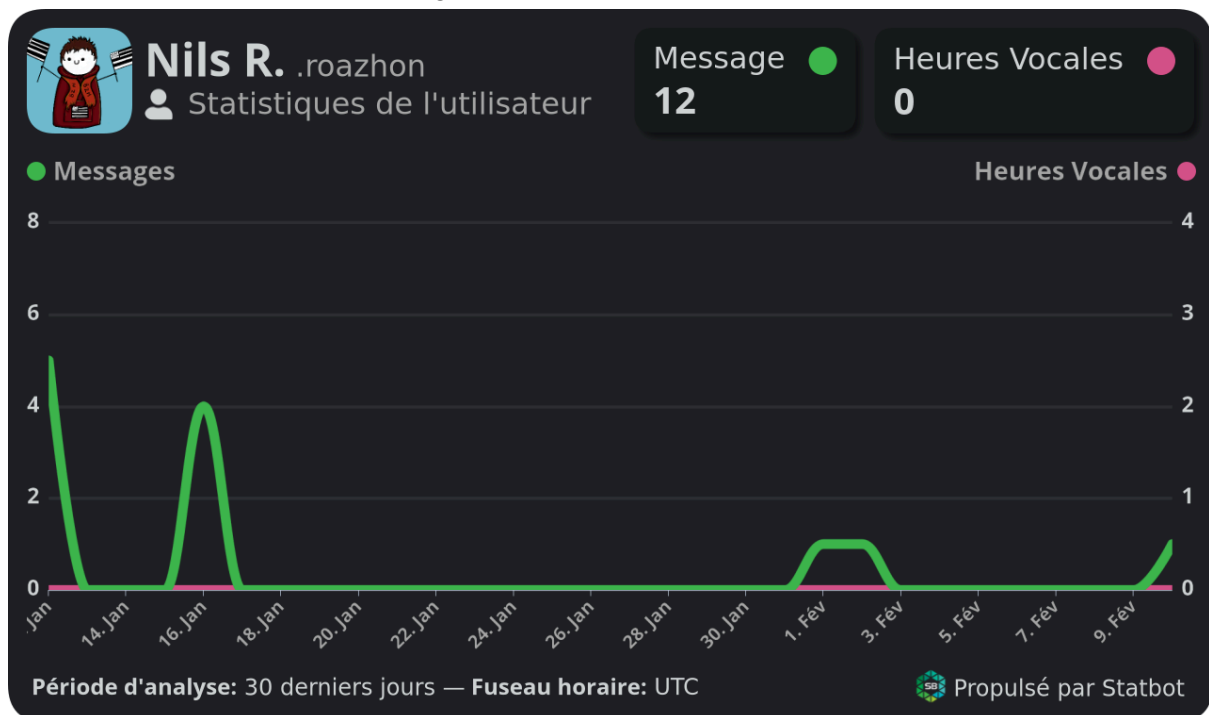
Statistiques Discord globales



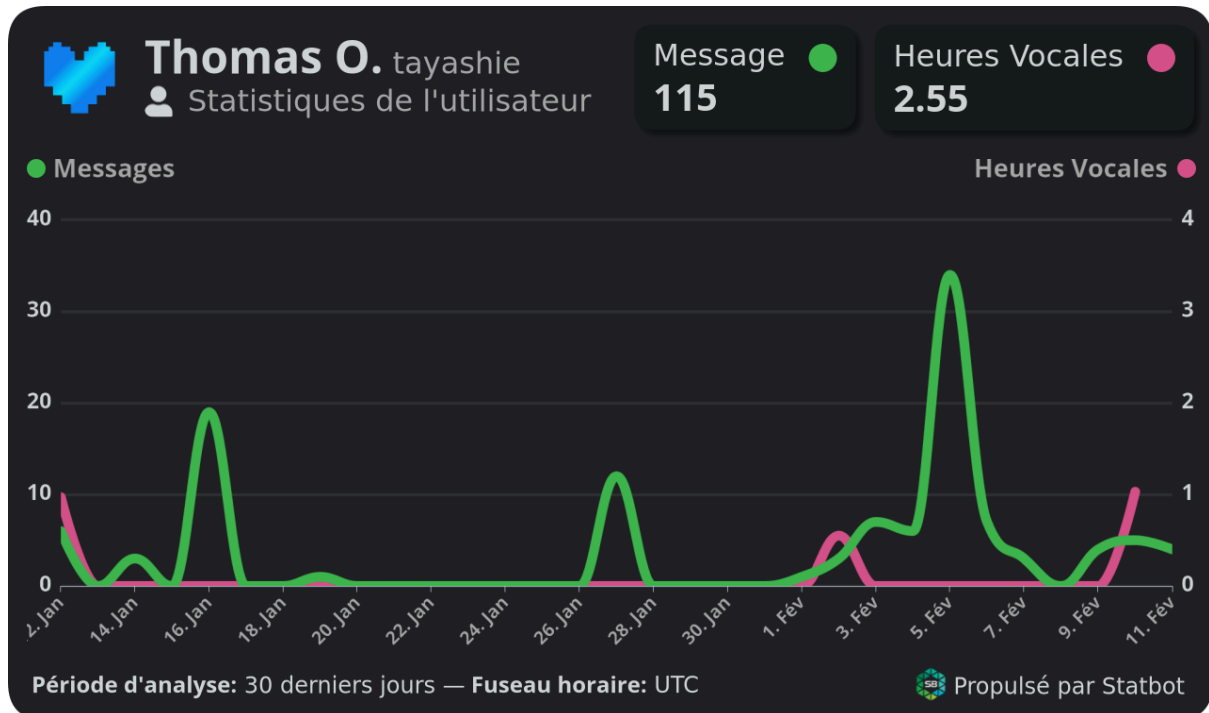
Statistiques Discord pour Louis Presti (sur 30J)



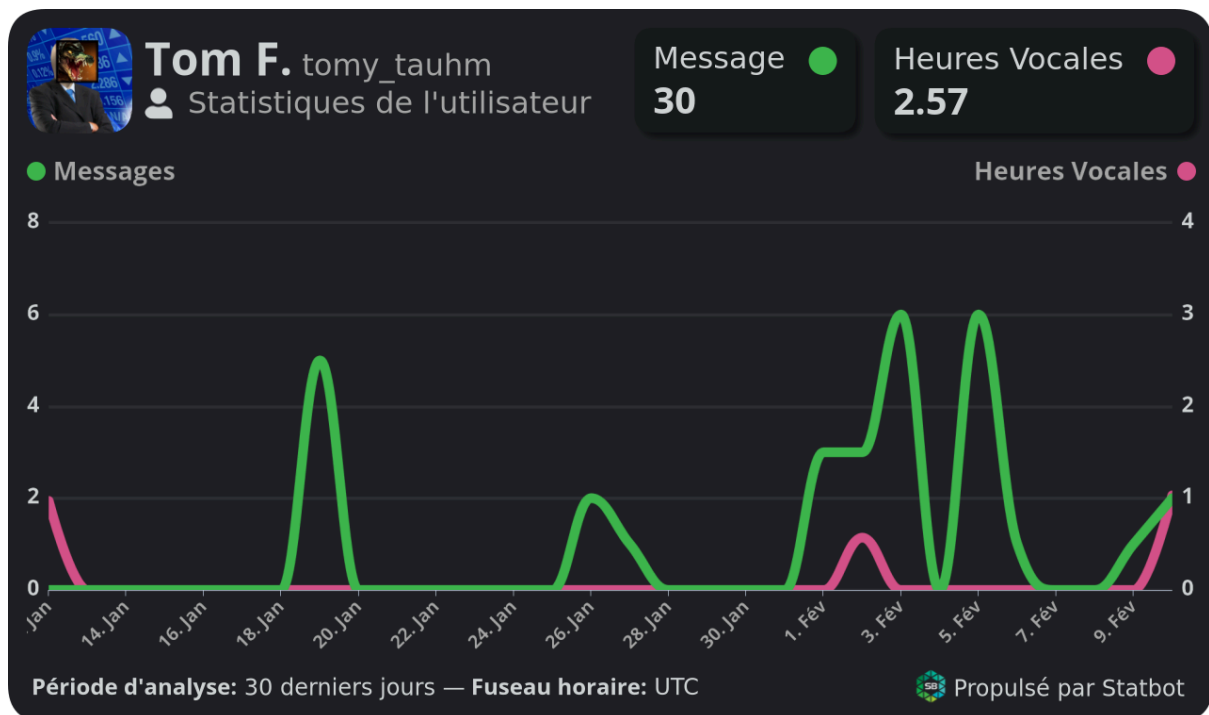
Statistiques Discord pour Nils Rageau (sur 30J)



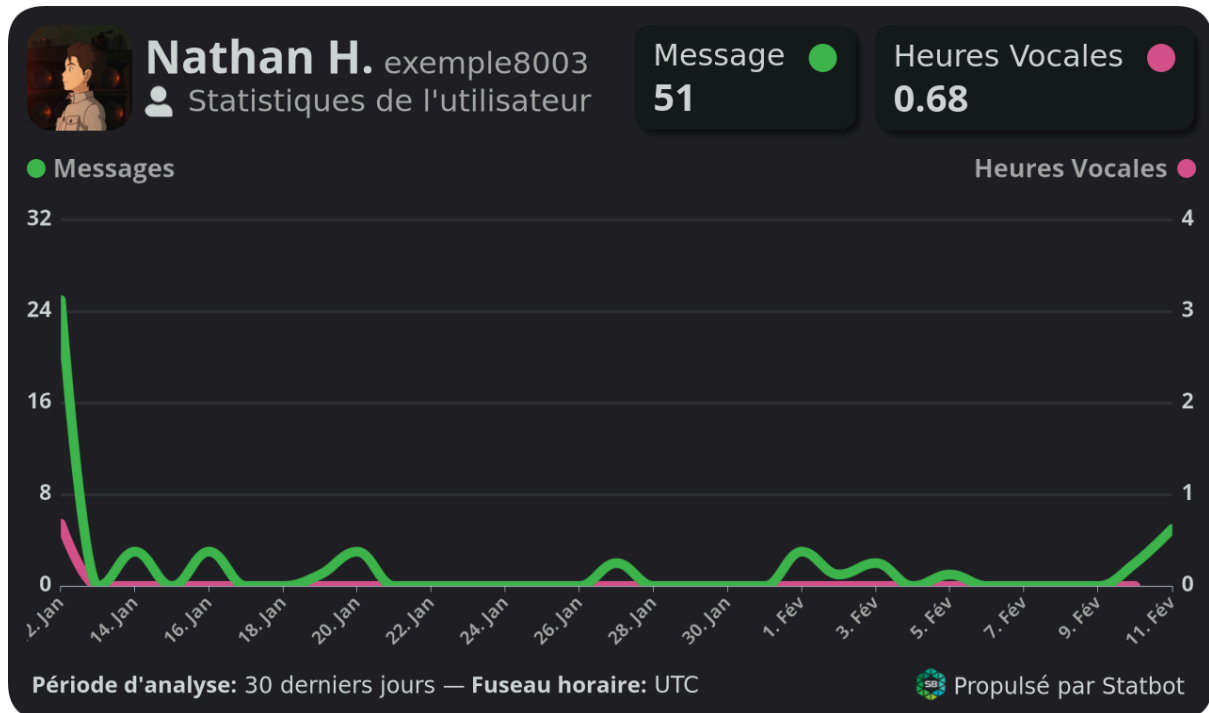
Statistiques Discord pour Thomas Oulmas (sur 30J)



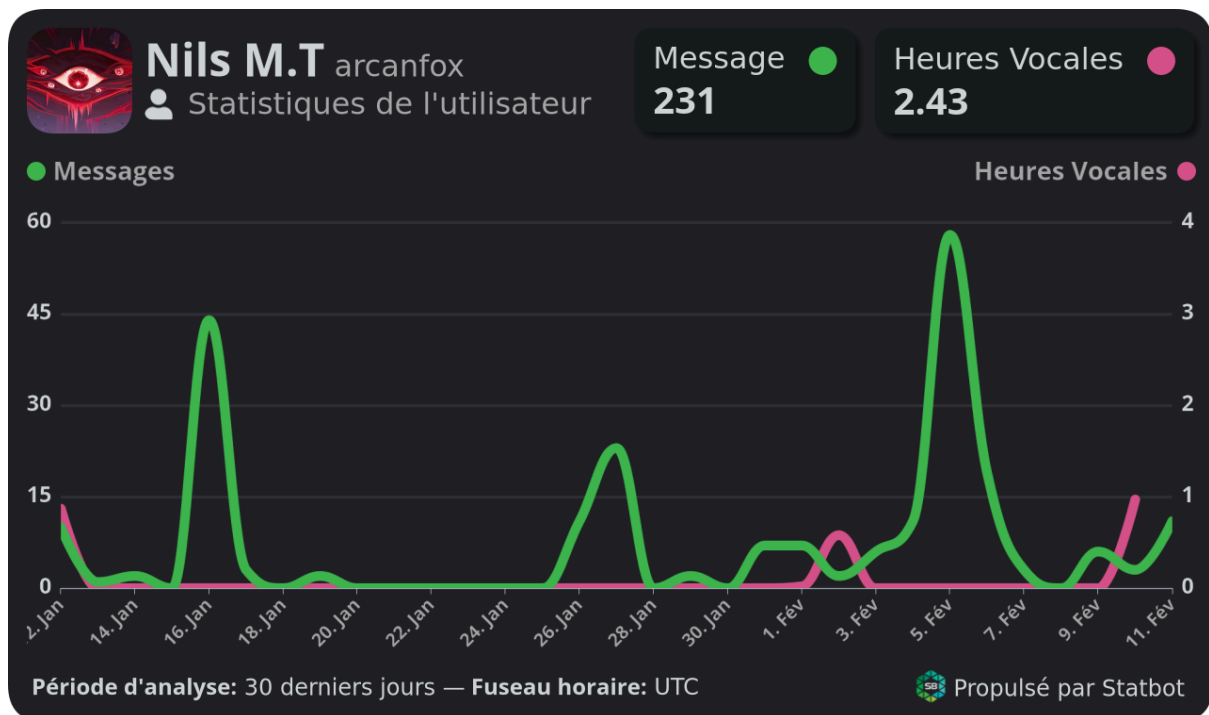
Statistiques Discord pour Tom Fremont (sur 30J)



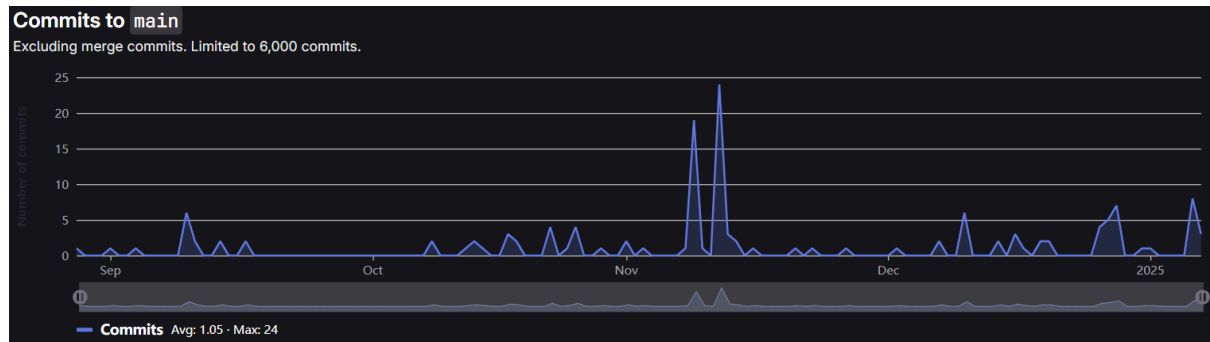
Statistiques Discord pour Nathan Hequet (sur 30J)



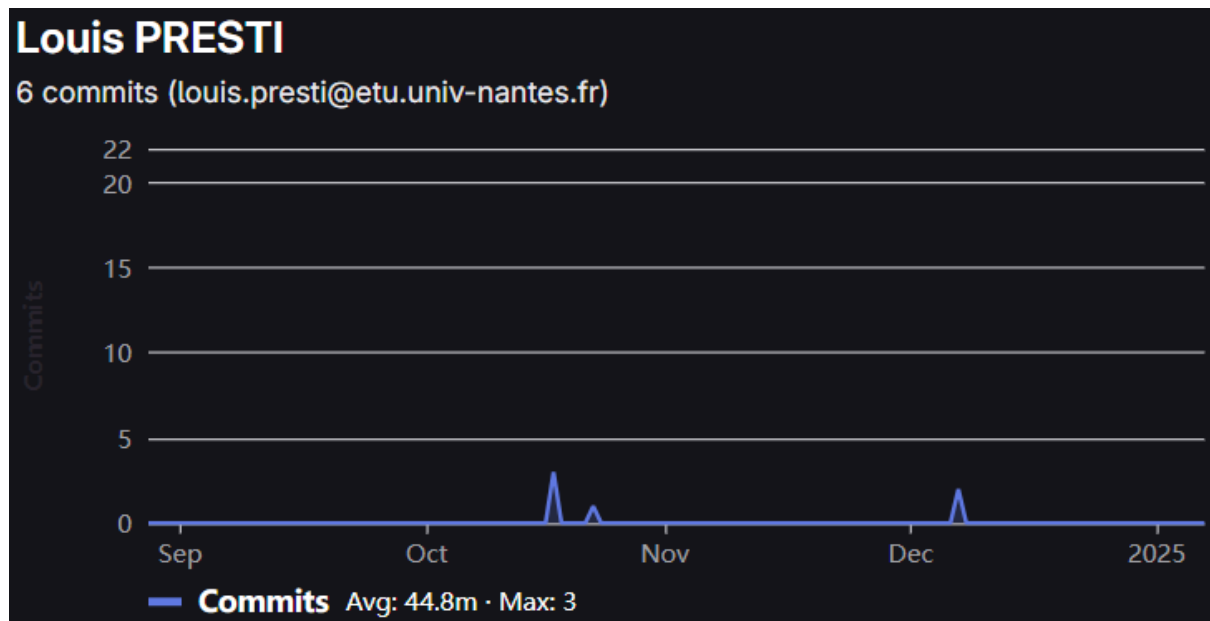
Statistiques Discord pour Nils Moreau-Thomas (sur 30J)



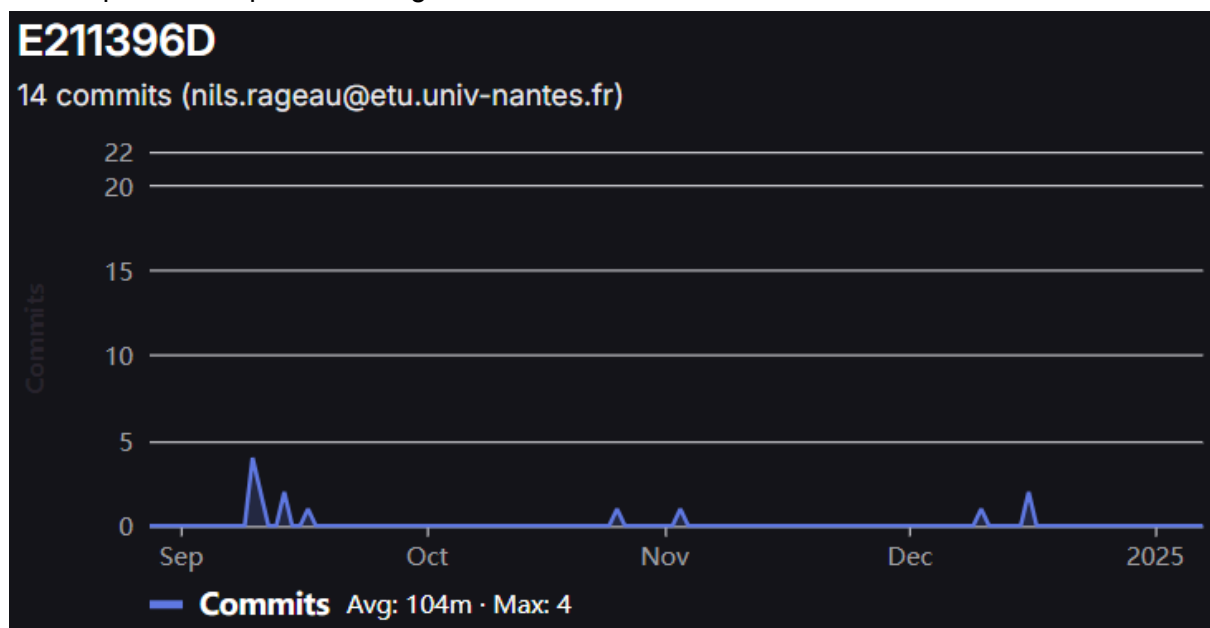
Statistiques Gitlab globales



Statistiques Gitlab pour Louis Presti



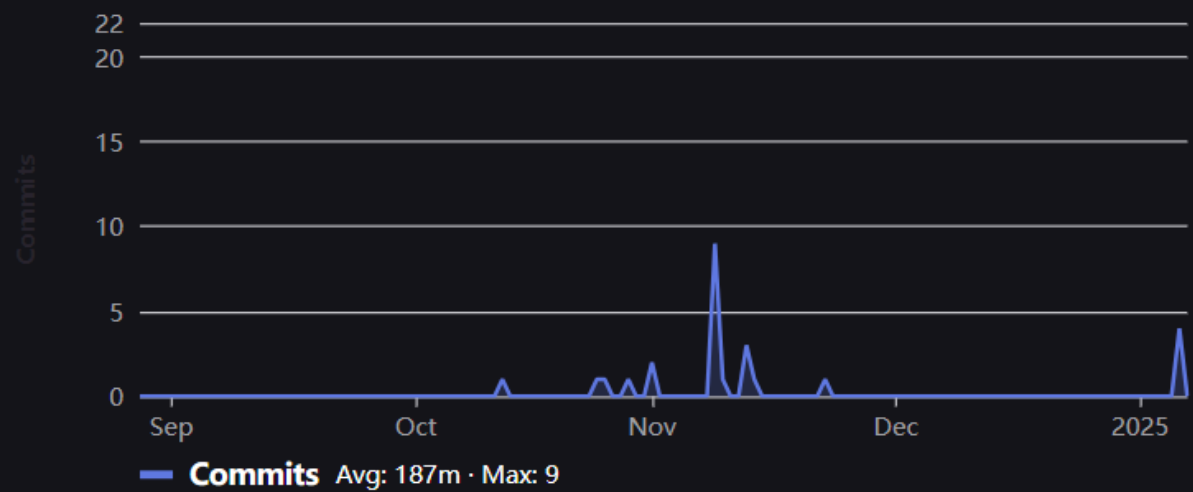
Statistiques Gitlab pour Nils Rageau



Statistiques Gitlab pour Thomas Oulmas

E225413P

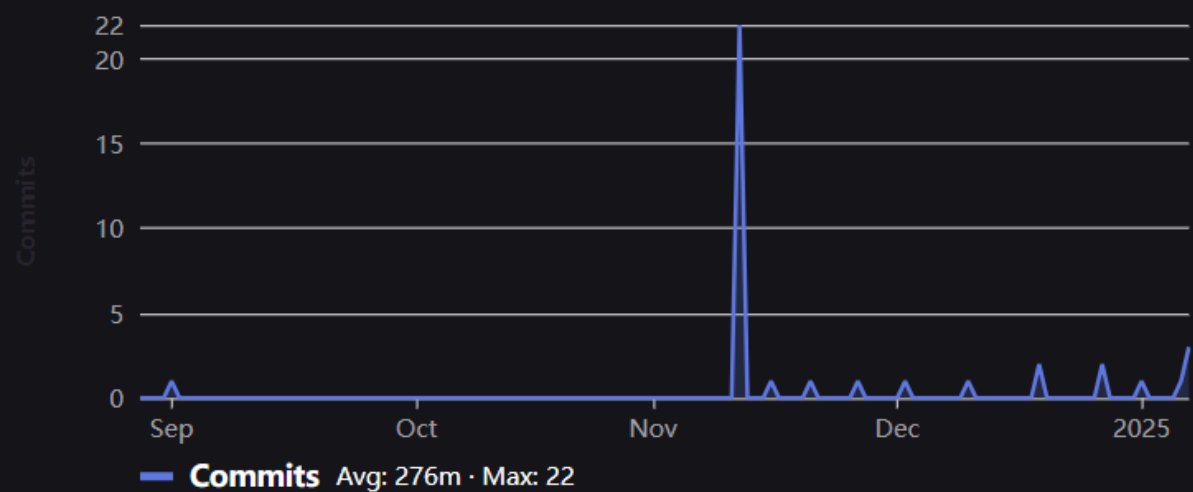
25 commits (thomas.oulmas@etu.univ-nantes.fr)



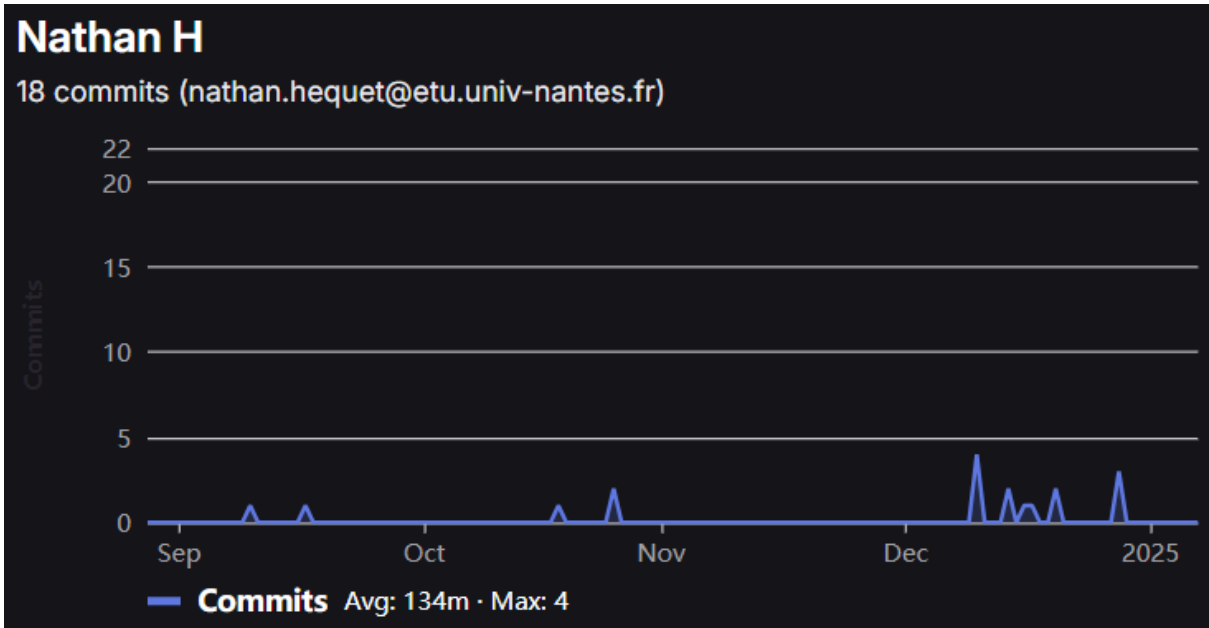
Statistiques Gitlab pour Tom Fremont

E226954P

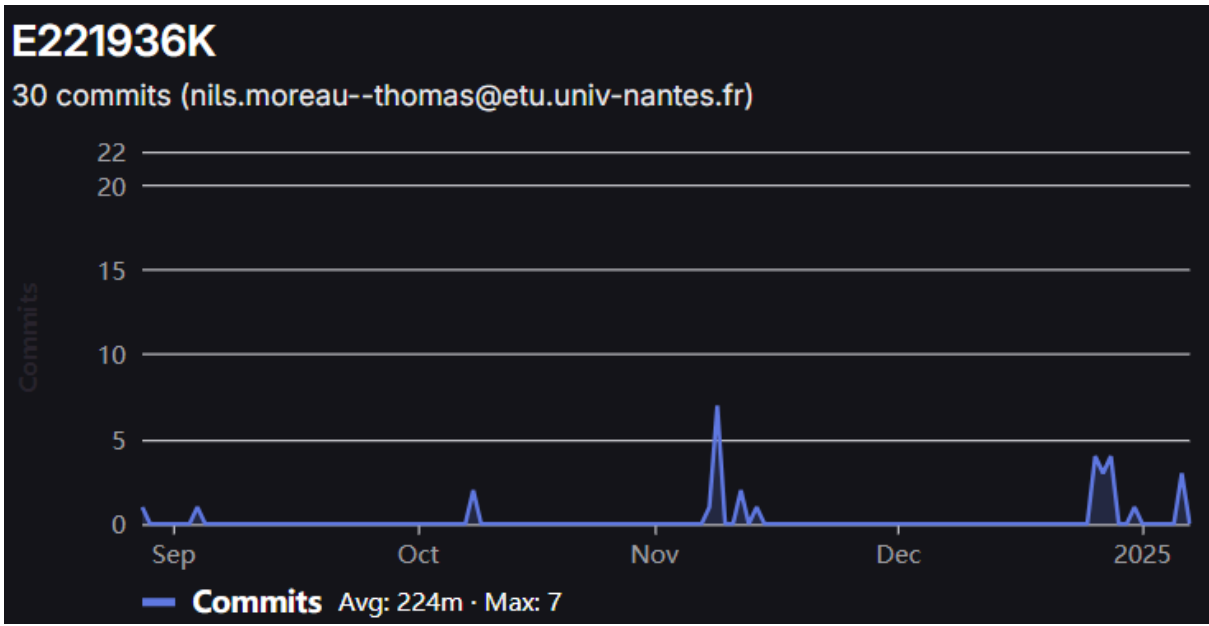
37 commits (tom.fremont@etu.univ-nantes.fr)



Statistiques Gitlab pour Nathan Hequet



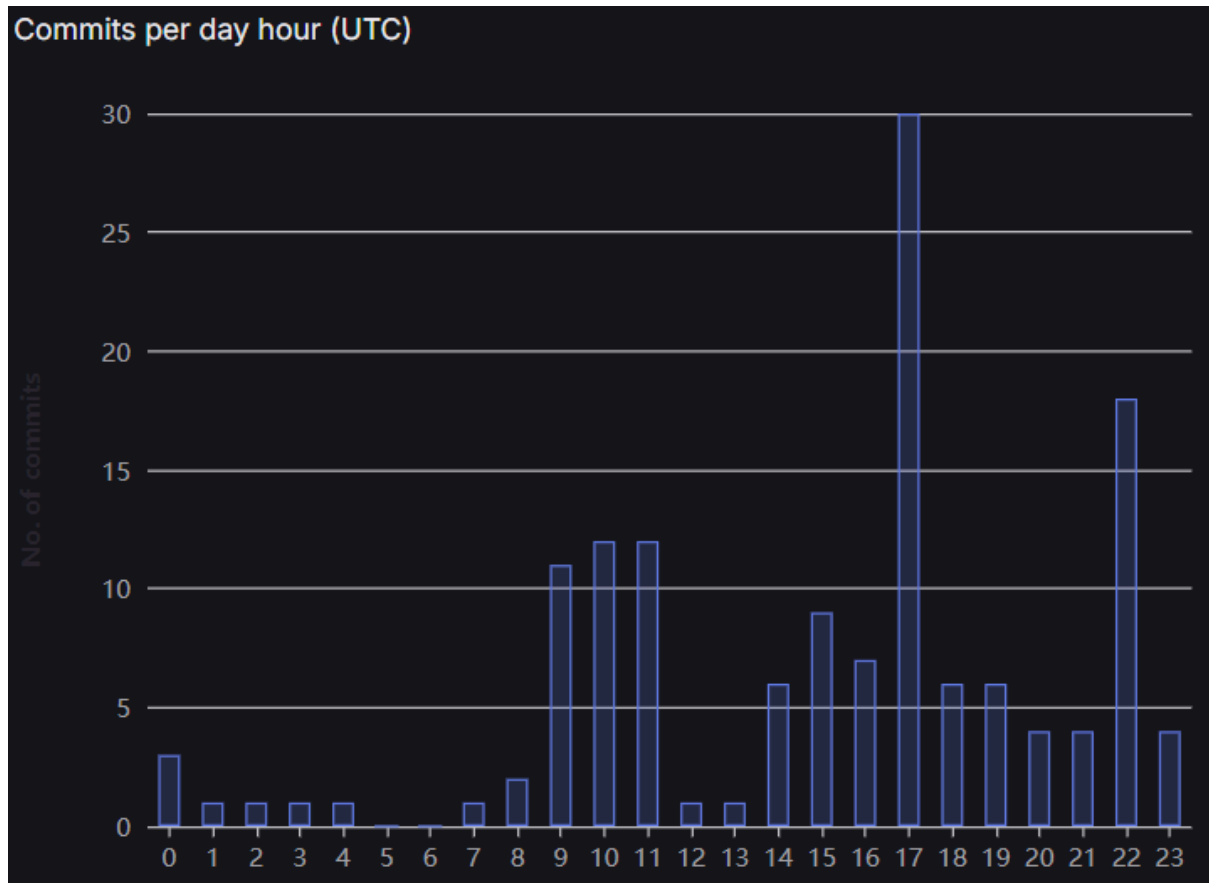
Statistiques Gitlab pour Nils Moreau–Thomas



Statistiques Milestone Gitlab

Version amélioré	sae5-2425-groupe2-medok / app	Open	0/0 complete	<div></div> 0%	
Version de base	sae5-2425-groupe2-medok / app	Open	44/59 complete	<div></div> 74%	

Statistiques Commits par heures Gitlab



Maquettes de l'application

1ère version

The wireframes show the following components for each screen:

- Création de profil:** Identifiant, Nom, Prénom, Date de naissance (DD/MM/AAAA), Mot de passe, Confirmation mot de passe, Taille (cm), Poids (kg), Allergies (Eau, Allergie 1, Allergie 2, Allergie 3), and a 'Créer profil' button.
- Connexion:** Identifiant, Mot de passe, and buttons for 'Créer un profil' and 'Se connecter'.
- Création de nouveau profil:** Identifiant, Nom, Prénom, Date de naissance (DD/MM/AAAA), Mot de passe, Confirmation mot de passe, Taille (cm), Poids (kg), Allergies (Eau, Allergie 1, Allergie 2, Allergie 3), and buttons for 'Retour' and 'Créer profil'.
- Mot de passe oublié:** Code de sauvegarde, and buttons for 'Retour' and 'Changer mot de passe'.
- Changement de mot de passe:** Nouveau mot de passe, Confirmation mot de passe, and buttons for 'Retour' and 'Changer mot de passe'.
- Code de sauvegarde:** Votre code de sauvegarde (XXX-XXX-XXX), a note about saving the code, a checkbox for confirmation, and a 'Suivant' button.

2ème version

The diagram illustrates the user interface design for a medical application, showing various screens and their components:

- Connexion:** Features a logo with a hand holding a green shield and the text "MedOK". Below is a "Connexion" section with input fields for "Identifiant" and "Mot de passe", and buttons for "Connexion" and "Oublier votre mot de passe".
- Échec de la connexion:** A small screen displaying "Échec de la connexion" and "Mot de passe / Identifiant invalide" with a "OK" button.
- Création profil (aucun...):** A screen for creating a profile with fields for "Identifiant", "Mot de passe", "Confirmation mot de passe", "Nom", "Prénom", "Date de naissance", "Sexe (m/f)", "Allergies", and "Vos données". It includes a "Créer le profil" button.
- Création profil (des pr...):** A screen for creating a profile with fields for "Identifiant", "Mot de passe", "Confirmation mot de passe", "Nom", "Prénom", "Date de naissance", "Sexe (m/f)", "Allergies", and "Vos données". It includes a "Retour" button and a "Créer le profil" button.
- Mon profil:** A screen for viewing the profile with fields for "Identifiant", "Nouveau mot de passe", "Confirmation du nouveau mot de passe", "Nom", "Prénom", "Date de naissance", "Sexe (m/f)", "Allergies", and "Vos données". It includes a "Retour" button and a "Modifier mon compte" button.
- Dialog code de s...:** A dialog box for "Mot de passe oublié" with fields for "Code de sauvegarde", "Nouveau mot de passe", and "Confirmer le nouveau mot de passe". It includes a "Retour" button and a "OK" button.
- Dialog code sauv...:** A dialog box for "Votre code de sauvegarde" with a "Vos données" section and a "Confirmer votre sauvegarde" button.
- Suppression de profil:** A dialog box for "Suppression de profil" with a "Oui" button and a "Non" button.
- Votre code de sauvegarde:** A screen displaying "Votre code de sauvegarde est: 1234567890" and a "Confirmer votre sauvegarde" button.