



ATTENDANCEBOT

TOUJOURS PRET A FAIRE L'APPEL

08/03/2021

Projet de gestion de présence dans un cours virtuel Discord

Equipe AttendanceBot

Module Génie Logiciel

L3 MAGE APPRENTISSAGE



UNIVERSITÉ
**PARIS
DESCARTES**



SOMMAIRE

SYNTHÈSE DU PROJET	3
Introduction	3
Choix de projet: Attendance Bot	3
Lien entre la théorie et la pratique	7
LE PROJET	8
Organisation mise en place	8
Problèmes rencontrés, solutions trouvées	9
Extrait du cahier des charges (techniques et besoins) depuis le repot Git	11
Points qui ont bien fonctionnés	13
Points perfectibles, restes à faire, points non solutionnés par rapport au cahier de charges	16
CONCLUSION	17
Retours personnels sur le module	17



SYNTHÈSE DU PROJET

Introduction

Dans le cadre de notre module de Génie Logiciel, nous avons appris des ficelles qui nous permettraient de nous adapter ou/et d'appliquer de bonne pratique dans le monde de l'entreprise.

Le projet suivant est une mise en pratique de quelques notions abordées en cours notamment celles autour de la gestion des projets.

I. Choix de projet: Attendance Bot

Notre choix de projet s'est orienté vers une bot Discord qui gère la liste des présents lors d'un cours.

Cette demande a été proposée par M. BOUTOUR qui trouve que le système d'aujourd'hui est superflu avec des actions redondantes. En effet, les professeurs doivent faire une capture d'écran de la liste des étudiants présents dans le canal audio du cours actuel puis retranscrire dans un fichier la liste des présents.

Cette liste ainsi que la capture d'écran sont ensuite envoyées au secrétariat qui rentre dans une feuille Excel/CSV la liste des présents à tel cours durant telles heures pour l'envoyer au CFA AFIA.

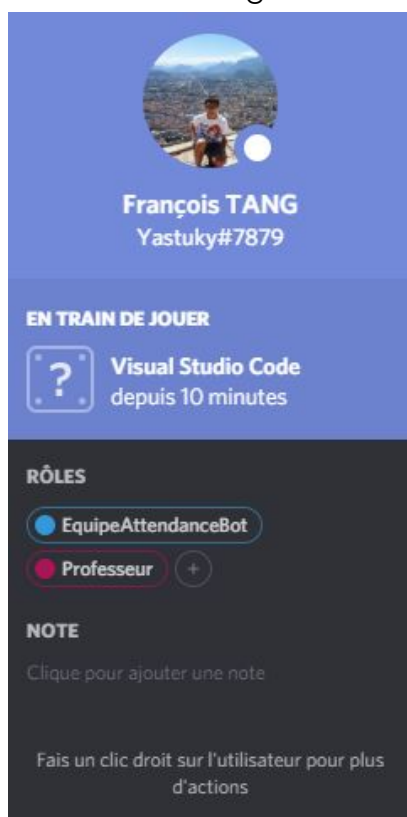
Après avoir écouté et compris le besoin de notre premier interlocuteur M. BOUTOUR, nous nous sommes tournées vers Véronique afin de lui demander si elle avait un besoin spécifique concernant ce système de présence. Après avoir échangé avec elle, nous avons eu part de sa demande qui reposait également vers un allègement des étapes redondantes qu'elle effectuait pour comptabiliser et envoyer les absences. Elle voudrait recevoir de la part de chaque professeur un fichier Excel/CSV avec la liste des étudiants présents et leur temps de connexion ou l'heure à laquelle ils se sont connectés au cours. La capture d'écran étant une demande spécifique du CFA AFIA il n'est pas possible de le supprimer à moins d'en discuter avec eux.

Suite à ces deux interview nous avons décidé de mettre en place un bot d'appel Discord qui permettrait aux professeurs de lancer un appel et de générer une feuille CSV donnant la liste de ceux qui ont réagi.

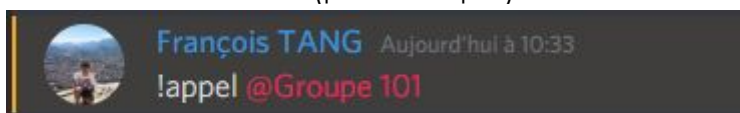


À l'heure actuelle le bot fonction de la manière suivante :

Étape 1 : Le professeur doit avoir obligatoirement le rôle "Professeur"

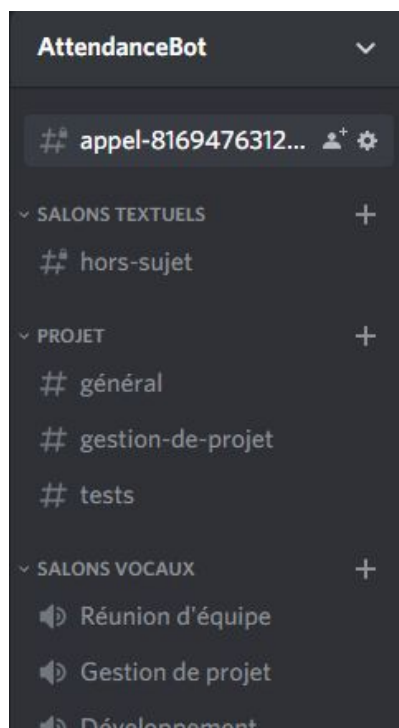
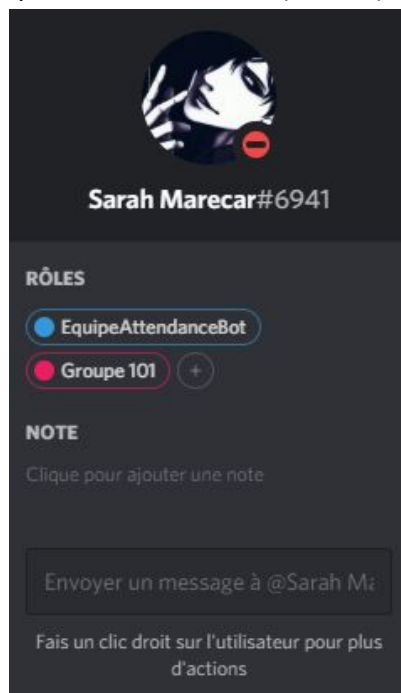


Étape 2 : le professeur lance un appel avec la commande "**!appel @Groupe 201**" (par exemple)



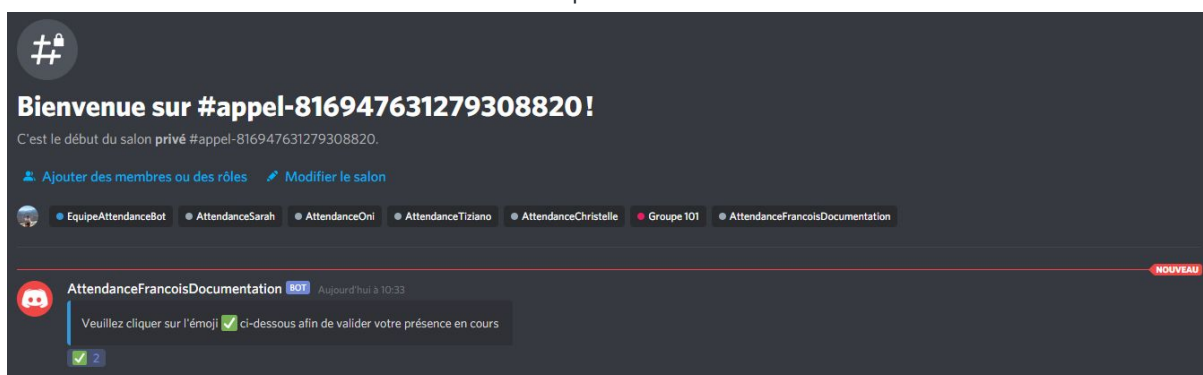


Étape 3 : les étudiants ayant le rôle concerné par l'appel (et le professeur lui-même) verront un salon spécifique se crée

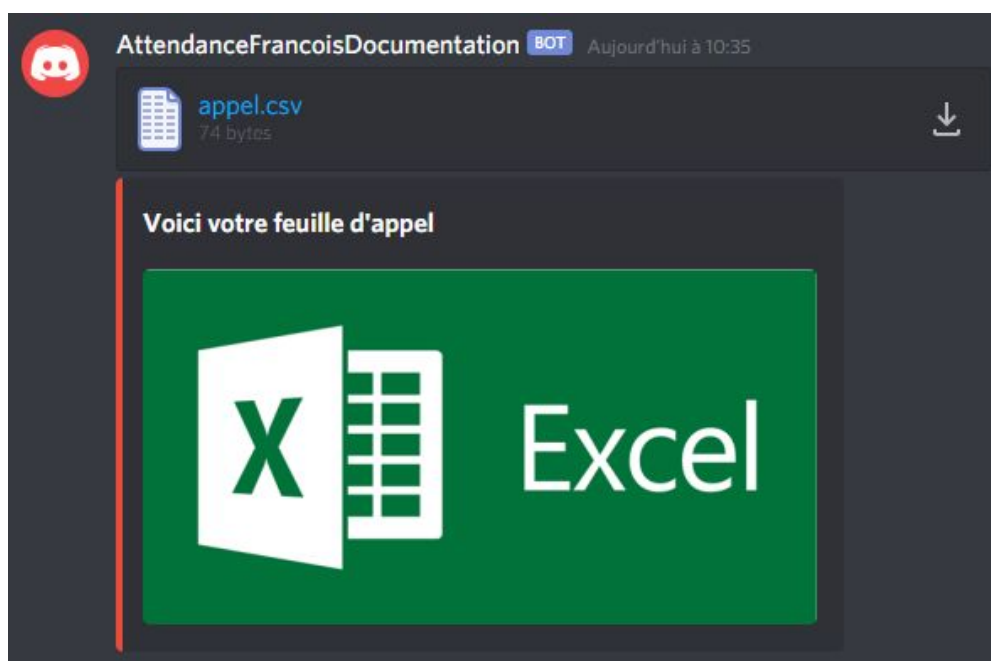
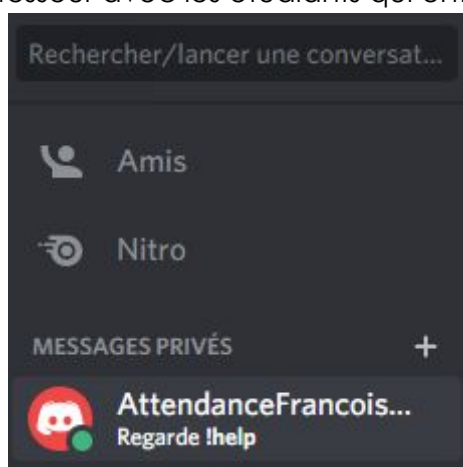




Étape 4 : ils verront un message les demandant de cliquer sur ✓ afin de valider leur présence



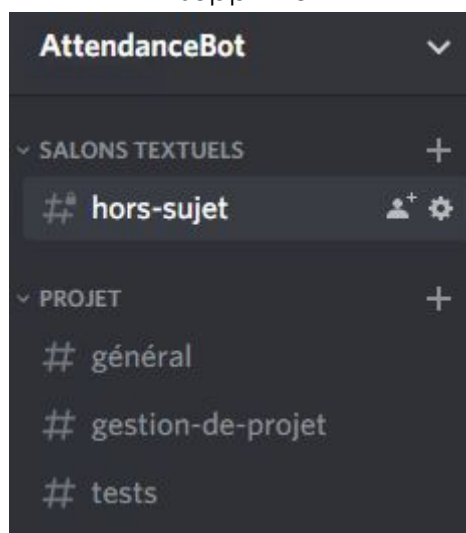
Étape 5 : 2 minutes après le début de l'appel, la feuille de présence est envoyée au professeur avec les étudiants qui ont cliqué sur l'icône





```
appel - Bloc-notes
Fichier Edition Format Affichage Aide
Id,Nom,Présence en cours visio,Date
1,Sarah Marecar,OK,07/03/2021 10:35
```

Étape 6 : 10 secondes après l'envoi de la feuille d'émargement, le salon est supprimé



II. Lien entre la théorie et la pratique

Le lien entre la théorie et la pratique fut clairement notre projet AttendanceBot. En effet, nous avons mis en place un ensemble d'outils présentés en cours (git, etc.) qui nous a en effet bien servi. Et ceux qui avaient déjà des compétences dans tel ou tel domaine aider les autres via des "sessions de formation" afin qu'on se mette à jour. Nous avons également respecté les consignes demandées, sur les méthodes de fonctionnement (agile, etc.) de développement, les différents tests à faire (intégration continue, tests unitaires, etc.), mais aussi les architectures de projet à respecter (l'injection de dépendances, l'inversion de contrôle, etc.). Tout cela nous permet de dire que le lien entre la théorie (cours) et la pratique (le projet) est bien réel. Mais il n'est parfois pas évident : nos tests ne sont pas tous finis par exemple et cela part d'un manque de temps.



LE PROJET

III. Organisation mise en place

1. Méthode de fonctionnement (Agile)

- a. Au début du projet (en groupe)
 - Écoute du besoin du client (Véronique/M. BOUTOUR)
 - Interview afin de mettre en place le cahier des charges
 - Ecriture du scénario global pour un "utilisateur" lambda
- b. Milieu du projet (développement-pair)
 - Session de développement en binôme/trinôme
 - Entraide entre les sous-groupes
 - Gestion de la ressource en scrumban
- c. Fin de projet (en groupe)
 - Ecriture du rapport + diaporama

2. Outils

- a. Au début du projet (en groupe)
 - Réflexion sur les langages/outils de mise en place du projet
 - Mise en place du kanban
 - Mise en place d'un serveur Discord pour nos tests et communication
 - Apprentissage du fonctionnement de git et sa mise en place : <https://github.com/AttendanceBOT/AttendanceBOT>
- b. Milieu du projet (développement-pair)
 - Utilisation de Visual Studio Code pour le TypeScript
 - Utilisation de TeamViewer pour les phases de développement
- c. Fin de projet (en groupe)
 - Utilisation de Google Drive

3. Mise en place de rôles

- a. **GHISOTTI Tiziano** - *Lead développeur*
- b. **MAHOLIDA Christelle** - *Scrum Master*
- c. **MARECAR Sarah** - *Développeuse*
- d. **RAKOTONIRINA Onitiana** - *Lead développeur*
- e. **TANG François** - *Gestion de projet*

Les rôles sont répartis de cette manière même si nous avons un peu tous la casquette de l'autre.



IV. Problèmes rencontrés, solutions trouvées

1. Slappey

Slappey est un outil CLI (Command Line Interface) qui permet de générer un projet Discord.JS et Discord.TS en quelques secondes. Nous avons dans un premier temps mis en place un projet sans architecture. Par la suite nous avons décidé d'utiliser le framework Slappey car il nous permettait d'avoir une architecture qui nous faisant gagner du temps en production, ce qui nous facilitait le travail mais ne respectait pas la contrainte imposée du projet (l'inversion of control, test unitaire etc.) .

Solution trouvée :

Ne répondant pas à nos critères, la dernière architecture vers laquelle nous nous sommes tournée est celle du livrable actuel et repose sur une architecture créée par nous-même avec TypeScript tout en respectant le design pattern Singleton.

2. Problème de gestion de temps

Durant la période du début du module à sa fin, beaucoup de projets se sont enchaînés avec des dates butoirs plus courtes que celles de Génie Logiciel. Nous avons alors fonctionné par priorité de rendu de projet ce qui a diminué le temps d'investissement accordé à celui-ci. L'autre problème qu'on a rencontré a été au niveau de nos emplois du temps. Ils étaient tous différents, dans la semaine, trouver du temps aux mêmes moments était rare et il fallait s'adapter en fonction de ceux qui étaient disponibles et non à chaque fois.

Solution trouvée :

La solution qu'on a mise en place a été de prioriser les fonctionnalités principales et voir par la suite si on pouvait faire plus. Pour mieux se raccorder on a également travaillé en petits groupes qui s'auto-organisaient entre eux.

3. Documentation Typescript

Typescript est un langage de programmation open source fait par Microsoft. C'est plus précisément un surensemble de Javascript. Elle a été conçue dans le but de faire des projets à grandes échelles. C'est



un langage qui reste toutefois jeune et qui a connu une popularisation croissante depuis 2017 ce qui peut poser certains problèmes.

Solution trouvée :

En effet, cette croissance rend sa documentation moins riche que d'autres langages comme Javascript par exemple, ce qui nous a obligé à nous auto-former sur de nouvelles pratiques d'utilisation.

4. Problème au niveau des rôles pour un appel

Nous avons une crainte qu'un étudiant d'un groupe A puisse valider sa présence pour un appel qui ne le concerne pas (d'un groupe B). Après le lancement d'un appel, on a pensé qu'il fallait récupérer leur rôle afin de leur donner l'autorisation de cliquer sur l'icône ✓ pour valider leur présence. Nous nous sommes rendu compte par la suite que cette solution était impossible avec l'API Discord.

Solution trouvée :

Pour pallier ce problème nous avons alors changé de stratégie d'autorisation d'émargement en cherchant ce qui était possible de faire avec l'API Discord. On considère qu'il est important de ne pas donner accès à n'importe qui le droit d'émarger dans un groupe dont il n'est pas affilié.

Désormais, lorsqu'un professeur lance un appel, il utilise le mot clé **!appel** suivi d'une alias qui permet de prendre un groupe d'élèves (ex: **@Groupe 202**). Au lieu de récupérer leur rôle, on va créer un canal visible et accessible uniquement aux étudiants faisant partie de ce groupe. Ils pourront ensuite cliquer sur l'icône qui permet d'émarger pendant 2 minutes. Dix secondes plus tard, après la fin de l'appel, ce channel d'appel disparaît.



V. Extrait du cahier des charges (techniques et besoins) depuis le repot Git

Contraintes du client (secrétariat)

Le bot

- Pouvoir réaliser les présences dans un cours virtuel sur Discord.
- Le professeur peut ajouter ou supprimer un étudiant dans la liste de présence après l'appel effectué.
- L'appel doit durer 2 minutes max (les étudiants ont 2 minutes pour manifester leur présence).

La feuille de présence

- La feuille de présence générée doit ressembler à l'actuel : le modèle Excel/CSV de feuille d'émargement (voire PDF).
- C'est le professeur qui est garant de la feuille de présence : il doit pouvoir exporter la fiche de présence d'un cours.
- Le mail d'envoi de la feuille de présence part manuellement du prof.
- Le professeur télécharge manuellement la feuille de présence, vérifie puis envoie au secrétariat.

Le rapport Discord

Le rapport discord doit contenir :

- L'heure d'arrivée et départ de l'étudiant.
- Le nom & prénom (surnom) de l'étudiant.
- Le classe/groupe (nom du rôle) de l'étudiant.

Les devoirs du client (secrétariat)

Les rôles Discord

- Les rôles Discord des étudiants doivent être explicites en fonction des groupes et déjà assignés à chaque étudiant.

Exemple : le rôle "Groupe 201" est assigné à l'étudiant "Jean-Marie".

- Le rôle Discord des professeurs doit être assigné.

Exemple : le rôle "Professeur" est assigné au professeur "M. DUPONT".



Les surnoms

- Chaque surnom d'un étudiant doit avoir été modifié sous la forme "Prénom NOM". Exemple : "*Toto DUPONT*".

Les channels

- Les channels doivent être organisés en module avec des titres explicites.

Exemple : le(s) channel(s) textuel(s) et vocal(aux) d'un module sont dans un groupe nommé "GÉNIE LOGICIEL 2 (V. BOUTOUR)".

Contraintes du responsable (M. BOUTOUR)

Sur le plan du projet

- Utilisation obligatoire d'un git pour le code.
- Expression du besoin via un document "User Story" avec les critères d'acceptation
- Réaliser une documentation complète en markdown.

Sur le code technique

- Le code doit être testé unitairement.
- Utilisation d'un langage comme Java, JavaScript, Go, Python, etc.
- Respecter l'Inversion of Control.
- Utilisation d'un outil de gestion de dépendance.

Fonctionnalités supplémentaires éventuelles

- Réalisation d'un message qui se déclenche à une heure précise (par exemple 9h) ou sur demande par un professeur (via une commande) chaque jour qui indique la fête du jour. Le mode de déclenchement est au choix.

Exemple : "Aujourd'hui on est le 01 janvier 2021 et c'est la fête du Saint Nicolas".



VI. Points qui ont bien fonctionnés

1. Besoins utilisateurs

a. Lancer un appel

Les personnes qui possèdent le rôle "professeur" peuvent lancer un appel à partir du mot clé : !appel. Celui-ci durera 2 minutes.

b. Émarger

Les personnes qui sont appelées par un alias (ex: @Groupe 201) peuvent émarger en cliquant sur l'icône ✓ dans le salon dédié.

c. Récupérer la feuille d'appel

Un fichier .CVS récapitulant les personnes qui ont émargé est envoyé par message privé directement au professeur qui a lancé l'appel.

2. Besoins techniques

a. Respect de l'inversion de contrôle

Dans le respect inverse de contrôle, on doit respecter l'injection de dépendance. Le framework prend en charge les blocs de code sans relation entre-eux. C'est le framework qui appelle les blocs de code (=l'application) et non l'inverse.

```
AttendanceBOT > ts inversify.config.ts > ...
9 import { CronSaintMessage } from "../src/services/cron-message-saint";
10 import { SaintMessage } from "../src/services/message.saint";
11 import { BotPresence } from "../src/services/bot-presence";
12 import { Help } from "../src/services/help";
13 import { HelpRes } from "../src/services/help-res";
14
15 let container = new Container();
16
17 container.bind<Bot>(TYPES.Bot).to(Bot).inSingletonScope();
18 container.bind<Client>(TYPES.Client).toConstantValue(new Client());
19 container.bind<string>(TYPES.Token).toConstantValue(process.env.TOKEN);
20 container.bind<string>(TYPES.Key).toConstantValue(process.env.KEY);
21 container.bind<string>(TYPES.Prefix).toConstantValue(process.env.PREFIX);
22
23 container.bind<EmbedRoll>(TYPES.EmbedRoll).to(EmbedRoll).inSingletonScope();
24 container.bind<PingFinder>(TYPES.PingFinder).to(PingFinder).inSingletonScope();
25 container.bind<DateFormat>(TYPES.DateFormat).to(DateFormat).inSingletonScope();
26 container.bind<CronSaintMessage>(TYPES.CronSaintMessage).to(CronSaintMessage).inSingletonScope();
27 container.bind<SaintMessage>(TYPES.SaintMessage).to(SaintMessage).inSingletonScope();
28 container.bind<Help>(TYPES.Help).to(Help).inSingletonScope();
29 container.bind<HelpRes>(TYPES.HelpRes).to(HelpRes).inSingletonScope();
30
31 container.bind<BotPresence>(TYPES.ActivityGame).to(BotPresence).inSingletonScope();
32
33 export default container;
34
```

b. Mise en place de l'intégration continue

On a mis en place une intégration continue dans notre GitHub. À chaque "push" ou "pull request", notre projet GitHub lance une série de tests qui doit passer sinon on reçoit un message d'erreur par mail. Cela permet de voir si le futur code ne "casse" pas l'ancien.



● Ajout de l'activité (#28)	Qualification	21 hours ago	...
Continuous Integration #45: Commit a531f3 pushed by Tizianogh			
● Ajout de l'activité	ftang/feature/activity-bot	21 hours ago	...
Continuous Integration #44: Pull request #28 synchronize by Tizianogh			
● Merge branch 'Qualification' into ftang/feature/activity...	ftang/feature/activity-bot	21 hours ago	...
Continuous Integration #43: Commit 2d9c214 pushed by Tizianogh			
● Correction gitignore	ftang/feature/activity-bot	22 hours ago	...
Continuous Integration #42: Commit db42c6 pushed by Yastuky			
● Update bots	ftang/feature/activity-bot	22 hours ago	...
Continuous Integration #41: Commit e41ea3 pushed by Tizianogh			
● N'aurait pas dû être dans le commit	ftang/feature/activity-bot	22 hours ago	...
Continuous Integration #40: Commit 5fbedb2 pushed by Yastuky			
● fix qualification (#29)	Qualification	22 hours ago	...
Continuous Integration #39: Commit ea0fda3 pushed by Tizianogh			
● fix qualification	tghisotti/fix/qualificati...	22 hours ago	...
Continuous Integration #38: Pull request #29 opened by Tizianogh			
● fix qualification	tghisotti/fix/qualificati...	22 hours ago	...
Continuous Integration #37: Commit 789b9fc pushed by Tizianogh			
● Ajout de l'activité	ftang/feature/activity-bot	22 hours ago	...
Continuous Integration #36: Commit 23e9e7e pushed by Yastuky			
● [tghisotti] (feature) : optional feature with cron-node a...	Qualification	23 hours ago	...
Continuous Integration #35: Commit 6ebd76 pushed by Tizianogh			

c. Utilisation de gestion d'un outil de dépendance : npm

Le gestionnaire de paquets officiel de Node.js est npm (Node Package Manager). Il permet de télécharger et d'installer des paquets (aussi appelés modules) pour les utiliser dans un projet ou au contraire pour partager des paquets pour que d'autres utilisateurs puissent les utiliser.

d. Tests unitaires

Des tests ont été réalisés sur certaines fonctions de notre code, notamment sur les tests concernant les date-fns, et help-res.ts. Tout le code n'est pas testé unitairement mais une partie a été faite.

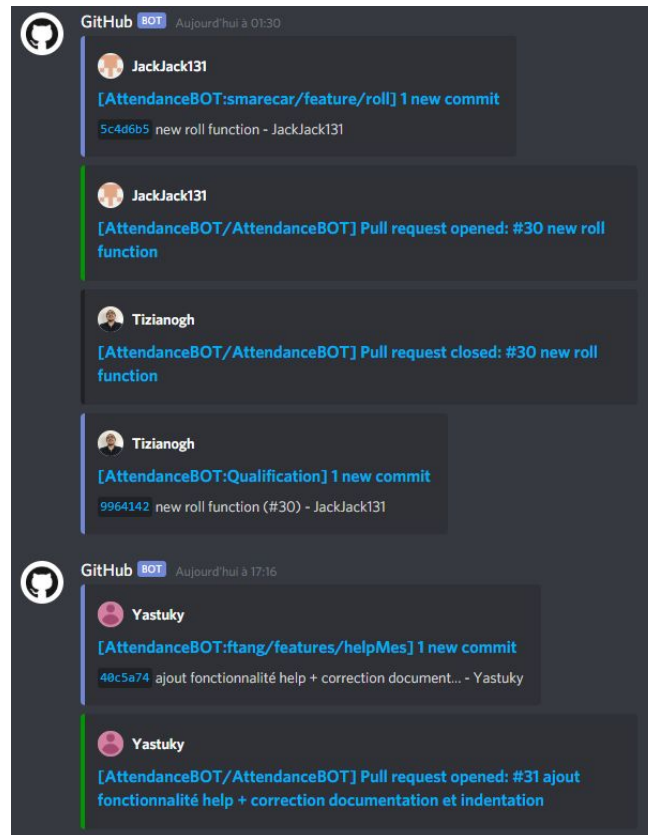
e. Gestion de la concurrence

La concurrence est bien gérée par le bot. Si un professeur lance un appel et un autre professeur lance un appel, le bot gère bien ces 2 derniers en parallèle.

f. Suivi du repot Git

Nous avons beaucoup fonctionné par pull request avec validation à chaque fois par un autre membre afin de ne pas écraser notre branche principale. Cela est une bonne pratique que nous avons mis en place et qui a bien fonctionné.

Nous avons également intégré un bot **GitHub** via un webhook sur notre serveur Discord dans le channel textuel *#developpement* qui est relié à notre repot git. Ce bot nous prévenait à chaque fois qu'un membre effectuait une action sur notre report (pull request, commit, push, review, etc.).



g. Réalisation du BONUS :

La fonctionnalité bonus a été développée et est le déclenchement d'un message chaque jour par le bot qui indique la fête du jour. La fonctionnalité est pour notre part déclenchée chaque jour à 9h ou via la commande **!saint**.



VII. Points perfectibles, restes à faire, points non solutionnés par rapport au cahier de charges

1. Fonctionnalités optionnelles qu'on aurait aimées développer (la liste de ce qui a été développée ou non se trouve dans le User Story)

- a. Action 4 : En tant que professeur, quand un appel est en cours, je peux annuler l'appel
- b. Action 5 : En tant que professeur, quand l'appel est terminé, je peux ajouter/supprimer un étudiant
- c. Action 6 : En tant que professeur, quand un appel est terminé, je peux supprimer le dernier appel
- d. Action 7 : En tant que professeur, quand un appel est terminé, je peux afficher la liste des présences en cours

Cela restent des options d'amélioration de l'application.

Le cœur de l'application en elle-même est présent mais pour la perfectionner un peu plus ces fonctionnalités auraient été bien.

2. Améliorer les fonctionnalités actuelles

- a. Lorsqu'un appel est lancé, permettre au bot de ping les élèves concernés dans le message qui demandent aux étudiants de réagir.
- b. Actuellement les appels marchent pour les rôles (!appel @Groupe 201).
 - i. Si l'argument derrière "!appel" est de type texte (!appel abcdefg) : ça fait rien.
 - ii. Si l'argument derrière "!appel" est utilisateur (!appel @ViBiOh#9339) : l'appel va se lancer mais uniquement pour l'utilisateur concerné.

Cela aurait été bien d'améliorer ces fonctionnalités.

3. Faire tests unitaires

Cela rejoint le point au-dessus mais il faudrait ajouter plus de tests unitaires ce qui permettrait de valider encore plus notre code afin que l'ensemble de nos classes et fonctionnalités soient testées.



CONCLUSION

VIII. Retours personnels sur le module

1. Tiziano GHISOTTI

Le cours de Génie Logiciel a été selon moi le cours se rapprochant le plus de la réalité en entreprise. J'ai vraiment apprécié tous les retours de vos différentes expériences et les astuces qui pourront nous aider plus tard. J'ai découvert des outils que je ne connaissais pas l'existence ou alors que je n'avais pas eu le temps de m'y former. Globalement, les notions évoquées dans votre module étaient très intéressantes.

Si j'ai un reproche à faire à votre cours, c'est peut-être le manque de pratique. La théorie c'est bien mais la pratique reste quand même mieux pour éviter de nous faire perdre le fil. Je suis aussi attristé de ne pas recevoir nos canards... :(

2. Onitiana RAKOTONIRINA

Le cours de Génie Logiciel est l'un des cours informatiques les plus complets que j'ai eu depuis le début de ma formation en MIAE. Il m'a permis de voir de manière générale et parfois détaillée les technologies, les architectures et la gestion de projet afin de mener à bien une création d'application en partant de zéro.

Dans les axes d'améliorations, j'aurais aimé avoir plus de détails sur la sécurisation des systèmes d'informations en générale ainsi que les codes sources outre que l'injection SQL par exemple.



3. François TANG

Pour être honnête, je dirais que ce module de GL était vraiment très intéressant. Je ne suis pas particulièrement "fan" du développement. Mais les cours dispensés par M. BOUTOUR ont été suffisamment intéressants pour que je l'écoute sérieusement. J'ai appris beaucoup d'outils utiles et le projet a permis de mettre en place une grande partie des bonnes pratiques que vous nous avez enseigné (Git, etc.). Voulant m'orienter dans un futur proche dans la gestion de projets, cela me permettra de mieux comprendre l'environnement des développeurs. J'ai également pu développer des fonctionnalités dans notre projet *AttendanceBot* et cela me confirme qu'une personne ne gère pas que le projet mais peut assister les développeurs si le besoin en est.

Sur les cours en eux-mêmes, je trouvais que c'était bien organisé (par thématique). Je pense qu'il manquerait un peu plus de démonstration sur ce que vous nous avez présenté. Il y en a eu beaucoup mais c'était très rapide et on n'a pas forcément pu tout assimiler (après ce n'est évidemment pas facile avec les cours à distance...).

Sinon, RAS : 9.5/10 ! :)

4. Sarah MARECAR

Honnêtement, le module m'a permis d'avoir une idée beaucoup plus organisée sur la programmation, sur l'architecture. Je ne connaissais que les bases, et j'entendais par-ci par-là des termes comme par exemple "Docker", "inversion de dépendance" sans vraiment comprendre ce que c'était, je ne savais pas faire la différence entre un test unitaire, test fonctionnel, et maintenant j'en ressors avec des connaissances beaucoup plus large et plus précise : franchement, c'était un cours plutôt plaisant de part son côté concret. Il y a certaines fois c'était très compliqué à suivre (je pense à la séance où on avait parlé de beaucoup de choses en l'occurrence des Patterns et des dépendances, c'était un cours assez condensé, je pense que mettre un peu de pratique entre temps aurait été peut-être idéal, je pense ici aux injections de dépendances et à git). En apprenant tout ça, cela nous a permis de vous montrer qu'on pouvait réaliser ce que vous nous avez appris lors des projets. En tout cas, keep it up ! C'est cool d'avoir eu un cours où on se sent libre avec vous !



5. Christelle MAHOLIDA

Ce module a été, pour moi, l'un des plus intéressants en termes de contenu étant donné qu'elle reposait majoritairement sur les bonnes pratiques applicables en entreprise. Pour être honnête j'ai commencé à travailler depuis le début février et c'est après cela que j'ai réellement donné sens à certaines notions abordées (CQQCOQP, la méthode du Canard, la communication en entreprise en prenant le café, le fait de poser des questions etc.). D'un point de vue personnel, j'aime toucher à tout dans l'informatique même si j'ai plus d'affinités vers tout ce qui est conceptuel (UML, Merise), analyse de données, et j'apprécie interagir avec les autres. Je porte tout de même un certain point d'honneur sur la programmation parce que je souhaite pouvoir comprendre les personnes avec qui je travaille et être légitime au poste auquel je serais plus tard même si je ne sais pas encore vers où m'orienter.

Durant ce projet, je me suis retrouvée à plus m'investir dans tout ce qui était rédactionnel. Mais j'ai trouvé l'exercice de rédaction plutôt plaisant. Ce que je regrette a été de ne pas avoir assez maîtriser TypeScript car c'était l'occasion d'apprendre un nouveau langage.