**UNIVERSITÉ DU QUÉBEC À MONTRÉAL**

**LIVRABLE SPRINT 2**

**PERSONAL ASSISTANT**

**TRAVAIL**

**PRÉSENTÉ**

**À**

**MILI HAFEDH**

**DANS LE CADRE DU COURS**

**PROJET D’ANALYSE ET DE MODÉLISATION**

**INM5151**

**PAR**

**de VIPART Lukas**

**PINEAULT Yannick**

**THIEMONGE Christophe**

**18 NOVEMBRE 2019**

[**1. INTRODUCTION**](#_r9g6q69d332a) **2**

[1.1 Objectifs](#_sfhnnxbzvk7v) 2

[1.2 Vue d’ensemble du produit](#_ysw17qe4wixt) 2

[1.3 Définitions, acronymes et abréviations](#_ft7lpkyfvvux) 2

[1.5 Aperçu du document](#_5un10ld40ui1) 2

[**2. DESCRIPTION GÉNÉRALE DU LOGICIEL**](#_ys224y1isvtq) **3**

[2.1 Vue d'ensemble des fonctions du produit](#_rpo0vl7pqcrd) 3

[2.2 Contenu des sprints](#_2bq8x2qdie1t) 4

[**3. CONTENU DU SPRINT ACTUEL**](#_czfegikgpgzf) **5**

[3.1 Cas d’utilisation implantés dans le sprint](#_w3x5ntdm14r3) 5

[3.2 Modèle de classe](#_w2tjoe92ws6t) 6

[Diagramme sprint 1](#_utwz1m59ib2s) 6

[Diagramme sprint 2](#_c0td2j2da6ps) 7

[**4. Prochain sprint**](#_mrlgraiz8pr2) **8**

[4.1 Revue du sprint](#_8lmma1ahu4xx) 8

[4.2 Plan pour le prochain sprint](#_x32dyc4rjmtn) 9

[**5. Annexes**](#_jclp1imdc3ig) **10**

## **1. INTRODUCTION**

### **1.1 Objectifs**

Ce document vise à expliciter ce qui a été implémenté durant le second sprint de développement de Personal Assistant.

### **1.2 Vue d’ensemble du produit**

Discord est une application permettant aux gamers de discuter tout en jouant. Un chat vocal est proposé, mais aussi textuel avec de nombreuses options d’administration. Les utilisateurs doivent en premier joindre un serveur. Des rôles leur sont attribués permettant de discuter, partager des images, des vidéos, des liens, créer des salons privés, etc…

Afin de pallier à la difficulté d’administration d’un grand nombre d’utilisateurs (notamment sur des discords publiques) mais aussi pour ajouter des fonctionnalitées qui ne sont pas présentes à l’origine (mini jeu sur chat texte ou musique en chat vocal), des bots ont été introduits. Ils sont développés par la communauté et offrent une panoplie de fonctionnalités extrêmement large, allant de la simple attribution de rôles automatiquement à des utilisateurs, à la gestion poussée des messages et lancement de serveurs de jeux.

Personal Assistant sera un bot discord. Nous souhaitons par son développement améliorer la fluidité d’utilisation des serveurs sur lesquels il se trouve (administration) mais aussi permettre plus d’interaction avec les utilisateurs (mini-jeux, discussion, musique…). Il ne sera en outre pas utilisable en dehors de l’application discord.

### **1.3 Définitions, acronymes et abréviations**

Nous nommerons le bot PA pour Personal Assistant.

### **1.5 Aperçu du document**

Ce document contient en premier lieu une description générale du logiciel dans laquelle seront expliqués les objectifs que le projet veut combler tout en organisant de manière générale les sous-objectifs pour chacun des sprints.

Les cas d’utilisation sont ensuite présentés avec un diagramme de séquence plus détaillé pour l’un des cas d’utilisation. L’évolution du diagramme de classe est représentée par l’ancien diagramme ainsi qu’un nouveau représentant l’état actuel du bot.

La dernière section présente une revue du second sprint maintenant complété en passant par nos bons coups et nos faiblesses, puis par des solutions à certains problèmes, suivit par notre plan pour le prochains sprint.

## **2. DESCRIPTION GÉNÉRALE DU LOGICIEL**

### **2.1 Vue d'ensemble des fonctions du produit**

Ici sont énumérés les fonctions que notre produit final sera capable d’accomplir.

On divise ses fonctions en deux sections: divertissement qui permet de fournir une expérience discord plus amusante et administration permettant la gestion des utilisateurs.

Les commandes de divertissement :

* + - Lecture de musique par youtube : Les fonctions permettant de demander au bot de lire de la musique youtube dans le canal vocal.
    - Auto apprentissage de réponses automatiques à certains messages (ou de réactions) : Permet au bot d’apprendre une liste de réponses pré faites face à divers messages.
    - Lancement de mini jeux textuels (Pierre-papier-ciseaux/ lancer de pièces) : permet de lancer des mini jeux tel que Pierre-papier-ciseaux ou le lancer de pièces.
    - Recherche d’images google images/reddit : permet de rechercher et d’afficher dans le canal textuel des images tirées de reddit ou de google image.
    - Recherche de définitions: permet de rechercher une définition et de l’afficher dans le canal textuel.
    - Lancement et gestion d'applications sur la machine hôte : permet de lancer et gérer diverses applications sur la machine hôte comme par exemple un serveur minecraft pour jouer entre amis.
    - Demande de la météo pour une zone X : permet d’afficher la météo pour une ville demandée dans le canal textuel.
    - Demande l'heure pour une timezone X : permet d’afficher l’heure pour une ville demandée dans le canal textuel.
    - Ramener le résultat d’une recherche youtube : affiche dans le canal textuel le premier lien résultant d’une recherche youtube.
    - Ramener les statistiques du joueur pour un jeu : affiche dans le canal textuel les statistiques d’un joueur pour une liste de jeux prédéfinie.
    - Ramener des infos sur un film : Affiche dans le canal textuel des informations sur le film demandé.

.

Les fonctions d’administration:

* + - Autosuppression des commandes déjà réalisées : supprime une commande qui a déjà été exécutée par le bot.
    - Arrêt du bot : éteint le bot.
    - Ramener les logs d’un utilisateur : ramène les logs du serveur triés pour uniquement refléter un utilisateur.
    - Bannissement/kick d’un utilisateur : expulse un utilisateur indésirable du serveur.
    - Auto-attribution de rôles par réaction : Attribue automatiquement un rôle à un utilisateur en fonction de sa réaction à un message du bot.
    - Attribution de rôle par message : Attribue un rôle en utilisant un message dans le chat.
    - Attribution automatique d’un rôle de base : Attribue un rôle de base à tous les utilisateurs rejoignant le serveur.
    - Suppression automatique de certains messages par mots clés : Supprime automatiquement les messages contenant certains mots pris dans une liste.
    - Création de channels textuels/vocaux : Permet de créer un nouveau canal de discussion.
    - Envoie de messages privés aux utilisateurs du discord : Envoie un message privé de la part du bot à un utilisateur (message créé par un modérateur par exemple pour une annonce ou pour un avertissement).

### **2.2 Contenu des sprints**

Premier Sprint: Il a fallu en premier créer notre propre serveur discord et y ajouter un bot. Ensuite nous avons implémenté une partie des fonctions essentielles au bon fonctionnement du bot (certaines fonctions n’étaient pas complètes). Ce sont des fonctions de base présentes dans tous les bot discord. En priorité nous avons eu la commande d’attribution des rôles et la lecture de musique via youtube.

Deuxième Sprint: Nous souhaitions maintenir le bon fonctionnement du bot tout en implémentant les fonctions souhaitables. Nous considérons ces fonctions comme souhaitable car elles aident à faciliter la vie des utilisateurs sur le serveur discord (en supprimant les messages grossier dans le chat par exemple). Compléter ce qui n'avait pas été finis dans le sprint 1 (permissions selon le grade pour ban/kick/attribution et message d’au revoir pour la commande !bye). Ensuite la suppression automatique des messages par mot(s) clé(s) et les résultats de recherche youtube. Rendre le code plus propre en ayant des classes pour chacunes des fonctions de notre bot.

Troisième Sprint: Implémenter ce que l’on a pas pu dans les Sprints précédents (mise en pause d’un vidéo youtube par exemple). Si le temps le permet des fonctions optionnelles. Les fonctions optionnelles sont celles qui permettent le plus au bot de se distinguer des autres. Ce sont des ajouts personnels qui pourraient exister dans aucun autre bot. Les recherches de définitions et d’images auront la priorité dans le cas d’implémentation de fonctions optionnelles. En dernier, un mini-jeu sera développé si toutes les autres fonctions sont implémentées.

## **3. CONTENU DU SPRINT ACTUEL**

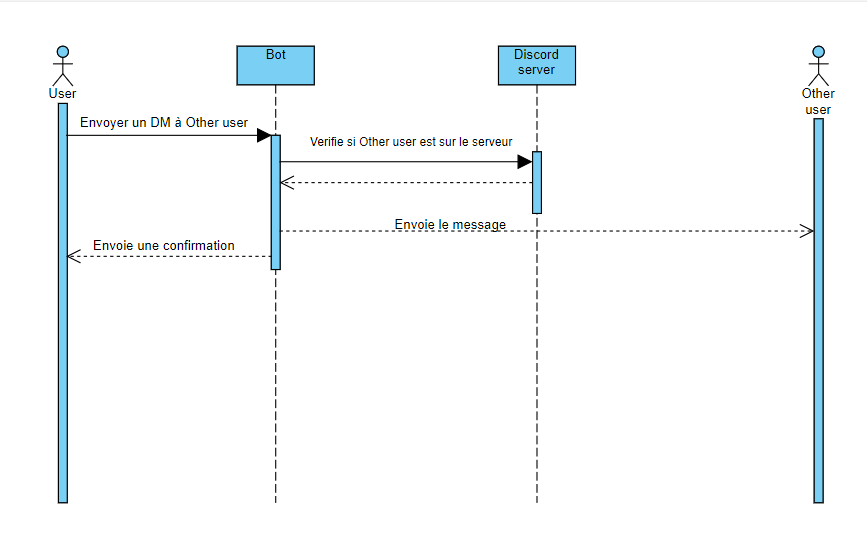
### **3.1 Cas d’utilisation implantés dans le sprint**

* Auto suppression des commandes déjà réalisées (uniquement pour certaines commandes) :

1. L'utilisateur rentre une commande
2. Le bot reconnaît et exécute la commande
3. En fonction de la commande le bot supprime le message initial de l’utilisateur

* Envoie de messages privés aux utilisateurs discord :

1. L’utilisateur rentre la commande pour le message privé
2. Le bot reconnaît la commande et récupère le message à envoyer
3. Le bot vérifie que l’utilisateur désigné fait partie du serveur
4. Le bot envoie le message à l’utilisateur désigné et à l'émetteur du message.

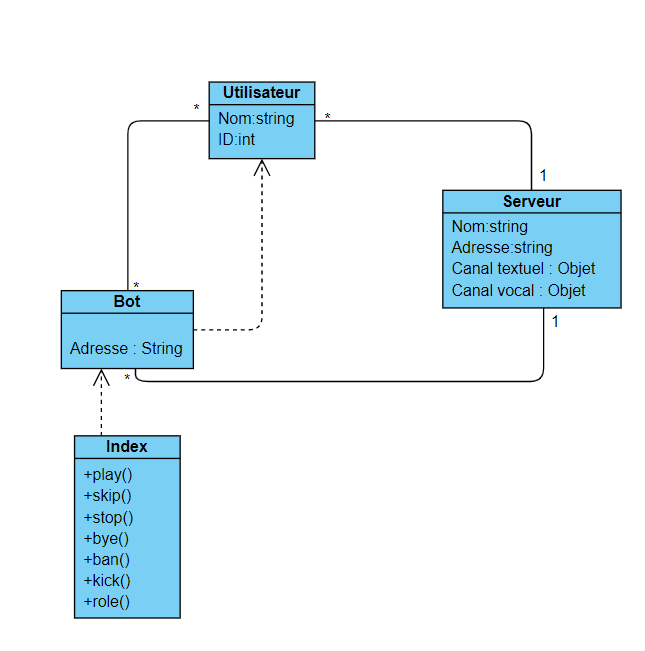


* Suppression automatique de certains messages par mots clé :

1. Un utilisateur entre un message
2. Le bot lit le message
3. Un mot compris dans la liste des mots interdit est détecté
4. Le bot supprime le message
5. Le bot écris un message avisant l’utilisateur de faire attention à son langage

### **3.2 Modèle de classe**

#### Diagramme sprint 1



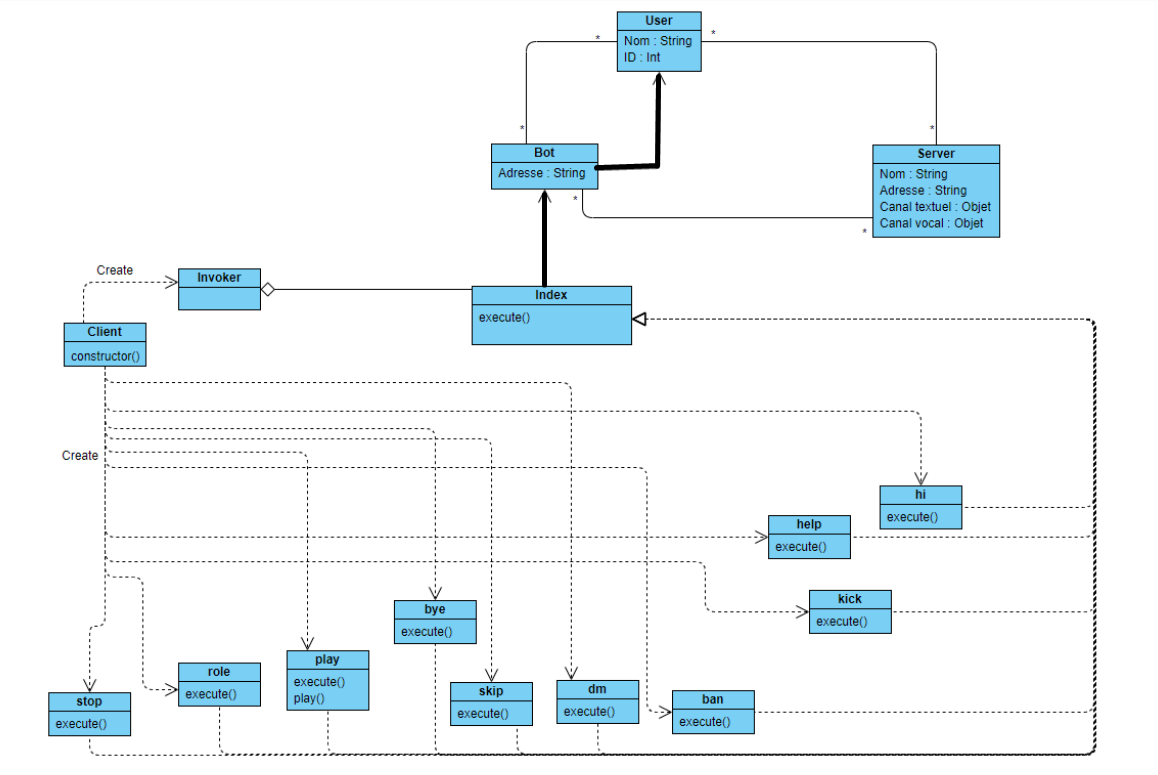
#### 

#### 

#### 

#### 

#### Diagramme sprint 2



## 

## 

## **4. Prochain sprint**

### 4.1 Revue du sprint

*Revue technique*

Nous avons réussi à compléter la majorité des fonctionnalités que nous avions classées comme étant *souhaitables* comme l’était prévue notre organisation des sprints. Toutefois, quelques détails non terminés du sprint 1 n’ont pas été complétés telle que la possibilité de faire pause sur la lecture de musique youtube. De plus, l’une des fonctionnalités que nous avions prévu d’implémenter durant ce sprint est présentement manquante, soit la lecture et la recherche d’une musique youtube via un titre de chanson (présentement via un URL).

De manière plus exacte, nos fonctionnalités souhaitables complétées comprennent ainsi:

* auto suppression des commandes déjà réalisées: Permet de supprimer automatiquement de la discussion les messages contenant des commandes pour PA pour diminuer la pollution (uniquement pour certaines commandes pour garder la cohérence du chat).
* envoie de messages privés aux utilisateurs du discord: Permet de communiquer de manière privée avec un autre utilisateur.
* suppression automatique de certains messages par mots clés: Filtre les messages pour s’assurer que certains mots interdits ne sont pas affichés dans le chat (censure).

Par ailleurs, nous avons aussi rajouté les vérifications d’autorisations sur les commandes de kick et de ban, afin de s’assurer que n’importe qui n’expulse pas des membres du serveurs, ainsi que pour la commande d'ajout de rôle, toujours pour éviter qu’un utilisateur lambda ne demande au bot de le nommer administrateur par exemple. La commande d’extinction du bot a elle aussi été corrigée pour qu’il mette un message de feedback lorsque l’on demande l’arrêt.

Nous avons également séparé le code en plusieurs fichiers selon le patron commande pour respecter améliorer la cohérence et pour rendre le tout plus lisible. Il est maintenant beaucoup plus facile de maintenir le projet et d’ajouter des fonctionnalités.

*Revue gestion*

À travers le premier sprint, notre principal problème était la mauvaise répartition du travail puisqu’un coéquipier avait travaillé plus que les deux autres. Toutefois, nous nous sommes ajustés et l’un d’entre nous a plus aidé dans la partie programmation ainsi que pour le rapport. Notre organisation s’est également amélioré avec la création d’un dépôt commun (via Google Drive) où tous nos documents sont présents.  
Toutefois la communication entre coéquipier a été moins bonne qu’au premier sprint, notamment puisque nous n’étions pas tous présents aux cours du vendredi soir.à cause de divers empêchements.

Ainsi, si nous avons encore des empêchements durant le sprint 3 durant les heures de cours, nous devrons tenter de nous organiser quelques heures, ou du moins, une rencontre en dehors du cours lui-même. Nous habitons tous loin les uns des autres, donc nous aurons l’option de se présenter à l’université ou de faire une réunion en ligne (via Discord par exemple).

Nous nous sommes encore une fois pris dans les derniers jours avant la remise ce qui a condensé le projet sous peu de temps. Il serait donc bon de se faire un calendrier relativement simple qui divise certaines tâches à faire avant *x* date durant le troisième sprint.

### 

### 4.2 Plan pour le prochain sprint

La recherche youtube par mot clé aura la priorité puisque cette fonction aurait dû être complétée durant le sprint 2.

Les recherches de définitions et d’images suivront dans le cas d’implémentation de fonctions optionnelles. Il sera alors possible via une simple commande et un mot clé donné par l’utilisateur d’avoir le ou les résultats les plus populaires sur Google en terme d’image et de définition, le tout agissant comme une sorte de dictionnaire.

Un mini-jeu au format texte devrait en outre être proposé à la fin du sprint.

## 

## **5. Annexes**