

Léo Lustgarten - Idée de Projet tutoré

Bot Discord : organisation de cagnotte

Description :

Le but de ce projet est de créer un bot discord permettant aux utilisateurs d'un serveur de simplifier les comptes d'une ou plusieurs cagnottes, et d'avoir facilement accès aux informations sur celle-ci. L'utilisation se fera à l'aide de commandes à taper dans un canal de messages. Plusieurs cagnottes peuvent être en cours sur un même serveur. Pas de moyen de paiement intégré, le bot sert à automatiser les calculs pour que les utilisateurs se remboursent facilement à la fin

Actions :

Diverses commandes permettront de : // Cette liste est susceptible de s'agrandir

- Créer une cagnotte (exemple : /createCagnotte nomCagnotte, échéanceParticipation [...])
- Donner un montant de participation (/participe [montant])
- Ajouter un objet à la liste des objectifs de cagnotte (/add nomObjet, prix, [...])
- Payer un objet de la liste
- Afficher les noms des cagnottes en cours
- Afficher les informations d'une cagnotte
- Ajouter un moyen de transaction préféré pour les participants (*Paypal, virement, liquide...*)

Certaines commandes seront accessibles que par certaines personnes

Créateur de la cagnotte :

- Ajouter/retirer un admin

Créateur et Admin :

- Retirer/Modifier une participation
- Retirer/Modifier un paiement
- Ajouter une participation d'un participant extérieur
- Renommer la cagnotte
- Calculer les dettes
- Envoyer en privé les infos des dettes

/ exemple un utilisateur recevra un message qui récapitule ce qu'il doit aux autres utilisateurs (ou ce qu'on lui doit), et quels sont les moyens de paiement que préfèrent ces autres utilisateurs */*

- Fermer la cagnotte

Technos :

D'après de rapides recherches il est possible d'utiliser du js ou du python comme langage de programmation. Je n'ai pas de préférences particulières entre les deux.

Le bot devra fonctionner avec une base de données. L'occasion parfaite pour mettre en pratique les cours de BDD de ce semestre (Les triggers appris lors de ces premiers cours seront par exemple utiles pour mettre à jour les infos d'une cagnotte à l'insertion ou la modification d'une participation. Ou encore pour annuler l'insertion d'un paiement si les fonds restants sont insuffisants)

Difficultés :

La partie programmation ne me semble pas comporter de complexité particulière.

La difficulté sera plutôt dans la nouveauté qu'est la programmation d'un bot Discord et dans la recherche de son fonctionnement (Comment s'y prendre, comment détecter les commandes et récupérer leur contenu, comment afficher les informations dans Discord, etc..) Ainsi que dans la mise en place d'une base de données propre, incluant les bonnes pratiques des cours de ce semestre au fur et à mesure.

PHASES DU PROJET

Phase 1 : Un utilisateur gère la cagnotte seul

Un seul utilisateur gère la cagnotte seul :

- Ajouter le bot sur un serveur
- Créer une cagnotte
- Ajouter des participations (créateur) //gardé par la suite pour l'ajout de participation extérieur)

Exemple : /participe [nom] [montant]

- Ajouter des dépenses (créateur)

Exemple : /achat [nom] [montant]

- Afficher les informations d'une cagnotte

Exemple : /afficher [nomCagnotte]

Le total des participations est de : [montant]

Vous avez dépensé : [montant]

Il reste donc : [montant]

- Afficher les noms des cagnottes en cours
- Calculer les remboursement à faire (pour qu'il y est le moins de transaction possible)
- Fermer la cagnotte

[Table Cagnotte, Participation, Achat](#)

Phase 2 : Participant et Achat précis

- Voir à quelles cagnottes on participe (éphémère)

Exemple : /moi

Anniversaire Gus

Surprise Leonore

(synthèse de combien on a donné comme participation (l'argent qu'on doit potentiellement) / combien on a payé (et donc l'argent qu'on nous doit))

- Chacun peu mettre sois même sa participation

Exemple : /participe [montant] // Récupération auto du pseudo de l'utilisateur

- Ajouter un objet à la liste des objectifs de cagnotte (/add nomObjet, prix, [...])

- Acheter un objet de la liste

Exemple : /achat [nomObjet]

- Ajouter un moyen de transaction préféré pour les participants (Paypal, virement, liquide...)

[Table Participant, Achat](#)

Phase 3 : Droits et modifs (ou que faire si les utilisateurs font n'importe quoi)

3 niveaux : Créateur, Admin, Utilisateur (modifier en conséquences les droits des actions précédentes)

- Ajouter/retirer un admin (créateur)
- Retirer/Modifier une participation (Admin)
- Retirer/Modifier un paiement (Admin)
- Renommer la cagnotte (Admin)

Phase 4 : Améliorations

- Envoyer en privé les infos des dettes

/ exemple un utilisateur recevra un message qui récapitule ce qu'il doit aux autre utilisateurs (ou ce qu'on lui doit), et quels sont les moyens de paiement que préfères ces autres utilisateurs */*

- Ajouter des dates d'échéances et des rappels
- Ajouter des auto complétions (les noms des cagnottes, les pseudos)

Liens utiles et documentation

Discord

<https://discord.com/developers/docs/intro>

<https://discord.com/developers/docs/interactions/application-commands>

<https://discord.com/developers/docs/interactions/receiving-and-responding>

Python

<https://docs.python.org/3/>

Python et discord: discord.py

<https://discordpy.readthedocs.io/en/latest/intro.html>

<https://discordpy.readthedocs.io/en/stable/ext/commands/commands.html> // commandes

<https://discordpy.readthedocs.io/en/stable/api.html?highlight=embed#discord.Embed>

//Pour block d'affichage

Python et SQLite: aiosqlite

<https://pypi.org/project/aiosqlite/>

SQLite

<https://www.sqlite.org/index.html>

Bot Free Hosting ?

[Daki | Free Bot Hosting](#)

Projet Tutoré

Bot Cagnotte

Kitty	
PK	ID : integer
unique	Name : text
>= 0	Funds : real DEFAULT 0
	CreatorName : varchar (35)
unique	ChannelName : varchar (100)

Share	
PK/FK	IdKitty : integer
PK	Pseudo : varchar (35)
>= 0	Amount : real

Purchase	
PK	IdPurchase : integer
FK	IdKitty : integer
FK	Pseudo : varchar (35)
>= 0	Amount : real
	Object : text