Architecture et Plan du back-end

Soit user l'utilisateur de la plateforme

Plan de la database

Table	Champs	Type	Description
Users	UserId	INT AUTO_INCREMENT PRIMARY KEY	L'id unique de l'utilisateur, clé primaire
	Name	VARCHAR(255) NOT NULL	Nom complet de l'utilisateur
	Password	VARCHAR(255) NOT NULL	Mot de passe de l'utilisateur
	Email	VARCHAR(255) NOT NULL UNIQUE	Mail de l'utilisateur (de préférence celui professionnel)
	ProfileImg	??????	Image de profile de user
	BannerImg	??????	Image de banière de user
	Created_at	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	La date d'enregistrement de l'utilisateur sur la plateforme
Groups	GroupId	INT AUTO_INCREMENT PRIMARY KEY	L'id unique du groupe. C'est la clé primaire
	AdminId	INT [FOREIGN KEY('AdminId') REFERENCES 'Users' ('UserId')]	Une référence de l'id de l'utilisateur à l'origine du groupe dans la table Users. C'est une clé secondaire
	GroupName	VARCHAR(255) NOT NULL	Le nom du groupe
	ProfileImg	??????	L'image de profile du groupe
	Created_at	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	La date de création du groupe
Members	MemberId	INT AUTO_INCREMENT PRIMARY KEY	L'id unique d'un membre d'un groupe donné. C'est la clé primaire
	GroupId	INT [FOREIGN KEY('GroupId') REFERENCES 'Groups'	Une référence de l'id du groupe dont user est membre dans la table

		('GroupId')]	Groups. C'est une clé secondaire
	UserId	INT [FOREIGN KEY('UserId') REFERENCES 'Users' ('UserId')]	Une référence à l'utilisateur en question dans la table Users. C'est une clé secondaire
	Created_at	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	Il s'agit de la date de l'adhésion de user à un groupe donné en tant que membre.
Sessions	SessionId	INT AUTO_INCREMENT PRIMARY KEY	L'Id unique de la Session. C'est une clé primaire
	SessionName	VARCHAR(255) NOT NULL	Le nom de la session
	SessionDate	TIMESTAMP NOT NULL	La date prévue pour la présence
	PreviewTime	TIMESTAMP	L'heure prévue pour la présence au cas où il s'agirait d'une présence planifiée
	StartingTime	TIMESTAMP NOT NULL	L'heure à laquelle la présence a été lancée par l'admin. C'est le fameux timer
	GroupId	INT [FOREIGN KEY('GroupId') REFERENCES 'Groups' ('GroupId')]	Une référence au groupe dans lequel la session de présence va se dérouler
	Status	ENUM('Pending', 'Forgot', 'Done') NOT NULL,	Indique le status de la session
	Gateway	BOOL NOT NULL	Indique le moyen de présence choisi par l'admin
	Created_at	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	La date de creation de la cession
sRepports	RepportId	INT AUTO_INCREMENT PRIMARY KEY	L'Id unique du rapport de présence. C'est une clé primaire.
	SessionId	INT [FOREIGN KEY('SessionId') REFERENCES 'Sessions' ('SessionId')]	Une référence à l'Id de la session dont on veux stocker le rapport. C'est une clé secondaire
	List	?????	La liste des membres obtenue à la fin d'une

			session de présence donnée de meme que leur status ('Présent' ou 'Absent')
AddingRequest	ARId	INT AUTO_INCREMENT PRIMARY KEY	Identifiant unique de la requête d'adhésion. C'est une clée primaire
	SenderId	INT [FOREIGN KEY('SenderId') REFERENCES 'Users' ('UserId')]	Une référence à L'Id de L'émetteur de la requête dans la table Users
	ReceiverId	INT [FOREIGN KEY('ReceiverId') REFERENCES 'Users' ('UserId')]	Une référence à L'Id du Récepteur de la requête dans la table Users
	Status	ENUM ('Pending', 'Accepted', 'Refused') NOT NULL	Indique si la demande a été aceptée ou refusée
	Type	ENUM ('IVT', DA) NOT NULL	Permet de différencier les demandes d'adhésions (DR) des Invitations (IVT)
	Created_at	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	La date et l'heure auquelles la demande a été émise
QrCode	QRCodeId	INT AUTO_INCREMENT PRIMARY KEY	Identifiant Unique du Qrcode. Clé primaire
	QRCodeSerie	????	La série de code générée pour une session donnée
	SessionId	INT [FOREIGN KEY('SessionId') REFERENCES 'Sessions' ('SessionId')]	Une référence à la session nécéssitant les codes Qr dans la table Session. C'est une clé secondaire
	Created_at	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	La date et l'heure auquelles le code fait référence
Notifications	NotificationId	INT AUTO_INCREMENT PRIMARY KEY	La contenance de data dépend en vrai du type car on est sensé y mettre toutes les informations nécéssaire. Les explications sont disponibles dans la section notification des
	UserId	INT [FOREIGN KEY('SenderId') REFERENCES 'Users' ('UserId')]	
	Type	ENUM('JGr', 'IVT', 'AR', 'OTH') Default 'OTH'	
	Data	????	

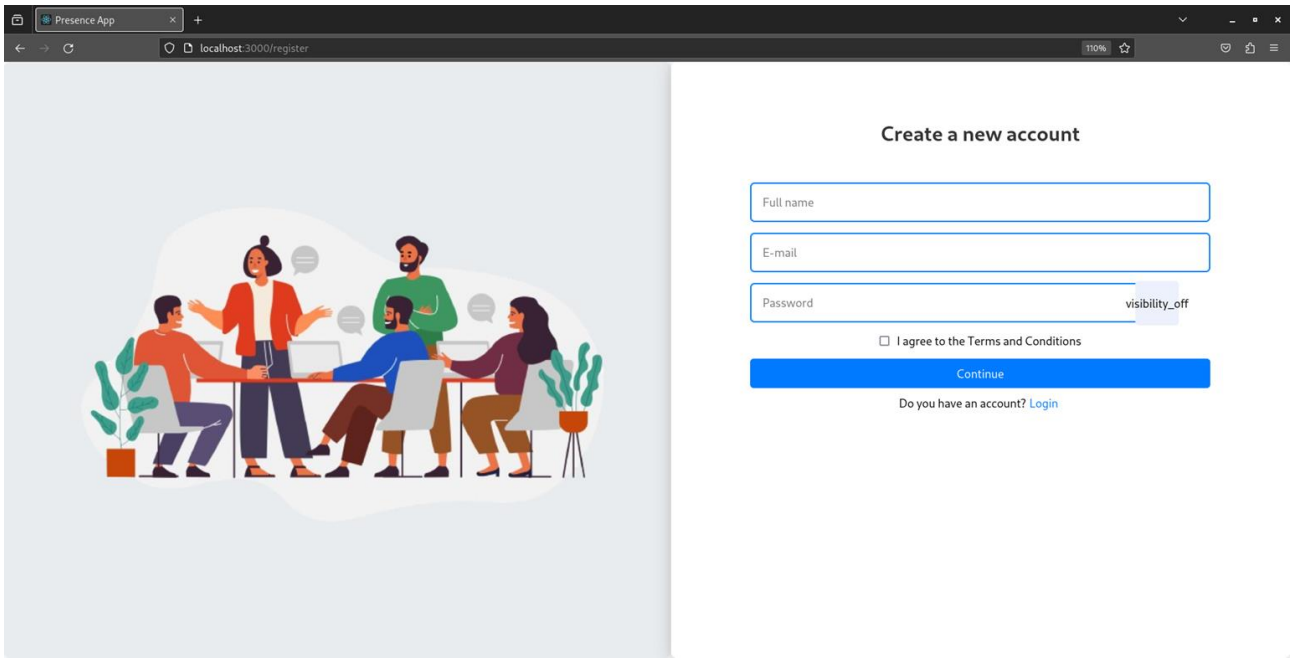
	Created_at	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	API à Implémenter.
Others	OtherId	INT AUTO_INCREMENT PRIMARY KEY	Id de Other. Pas trop d'importance en vrai □□
	IsExternalNotification	BOOL NOT NULL	Active et désactive les notifications externes
	IsSmsNotif	BOOL NOT NULL	Active et désactive les notifications SMS
	SMS Numer	???	Le numéro de téléphone pour recevoir le sms

*Register

Description : enregistrer user dans la database.
Nom de la route : /Register
input : {Name: 'Ken Takakura', Mail : 'dadadan@anime.fr', Password : 'Mémé turbo'**}**
output :
- {'token' : 'iorahgedkrzmouvzmddep' } en cas de succes
- le code http en cas d'échec

Vérifications :

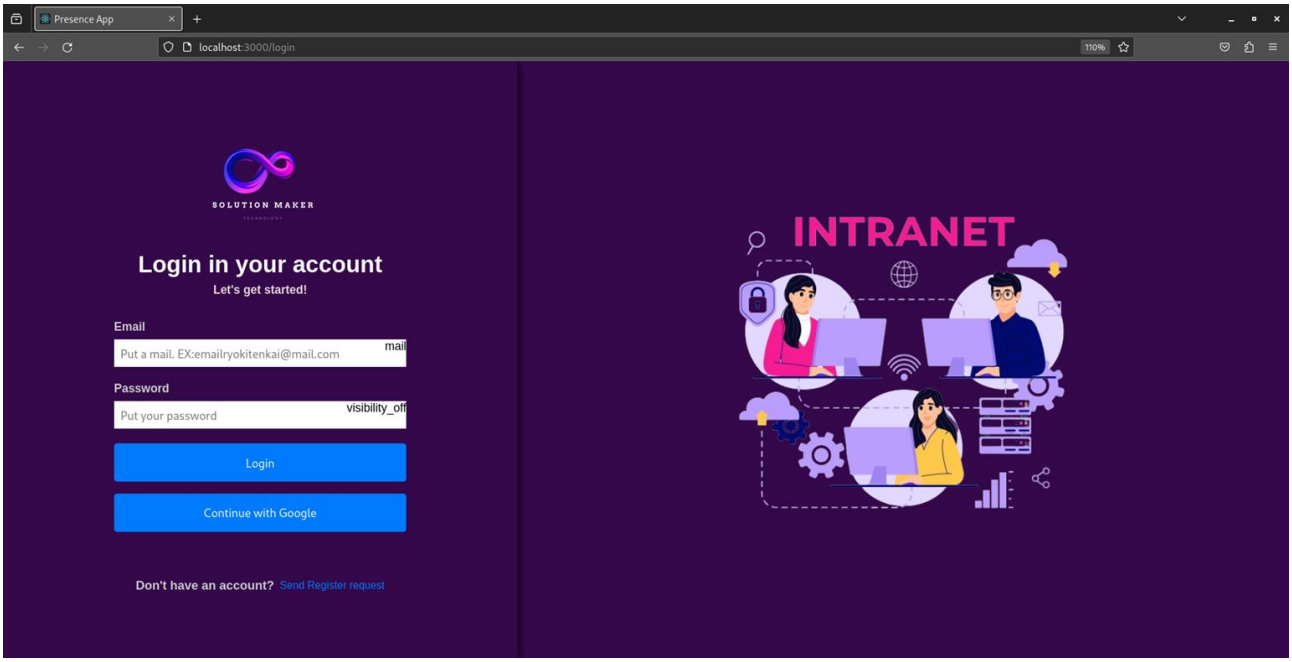
-Deux utilisateurs ne puvent avoir le même mail.



*Login

Description : connecter user à l'application
Nom de la route : /Login
input : {Mail : 'dadadan@anime.fr', Password : 'Mémé turbo'**}**
output :
- {'token' : 'iorahgedkrzmouvzmddep'} en cas de succes
- {'token' : '' ou NULL si l'utilisateur n'est pas enregistré

- - le code http en cas d’échecs



* Groupes :

1. Obtenir tous les groupe dans lesquels user est admin:

Description : Le travail ici consciste à récupérer les groupes que possedent user et de les retournés (nom, photo de profil).
Nom de la route :???
input : RAS
output : L’ensemble de ces goupes doit être retourné sous format json.
Nom de variable : AdminGroups

2. Obtenir tous les groupes dans lesquels users est membre

Description : Le travail ici consciste à récupérer les groupes auquels user appartient.
Nom de la route :???
input : RAS
output : L’ensemble de ces goupes doit être retourné sous format json.
Nom de variable : MemberGroups

3. Creer un nouveau groupe par user

   Description : Ici, un utilisateur peux creer autant de groupes à partir du nom qu’il veux afin d’y administer la présence. Le travail consciste donc le lui attribuer au niveau de base la de donnée. Toutefois, Il ne peux pas posséder deux groupes de même nom a son actif

Nom de la route :???
Input : Nom du groupe à creer
output : le code http.

4. Supprimer un nouveau groupe par son utilisateur admin

Description : Le travail ici consciste à supprimer un goupe donnée par l'utilisateur admin à partir de son nom au niveau de la base de donnée.
Nom de la route :???
Input : l'id du groupe à supprimer
output : Le code de la requête

5. Obtenir tous les membres d'un groupe donné dont l'utilisateur est admin

Description : Le travail ici consciste à tous les récupérer membres d'un groupe donné à partir de son nom.
Nom de la route :??
Input : l'id du groupe sélectionné
Output : L'api retourne tous les membres appartenant à ce groupe dans une variable AdminMembers

6. Ajouter des membres dans un groupe

Description : Ici, un utilisateur peux ajouter autant de membres qu'il veux dans un groupe dont il est admin si et seulement le membre est enregistré sur la plateforme (un peu comme sur whaths'app) afin d'y administer la présence.

Le travail à faire consciste a creer dans la database, une invitation (un record dans la base de donnée de type IVT) de la part de user, destinée à l'utilisateur à ajouté afin que ce dernier valide ou non son adhésion

6-1. Individuellement

Nom de la route :???
Input : Id du groupe sélectionné + email ou nom de l'utilisateur à ajouter
Output : le code http

6-2. En gros ou création de masssess

Nom de la route :???
Input : Un fichier exel qui sera fourni contenant les mail tous les mails des members à ajouter + GroupId.
Output : L'api retourne d'une part la liste des inputs qui ont pu être ajoutés (nom / mail) et de l'autre, une liste des inputs qui ont pas pus.

```
{
        'MemberAdded' : {'freepalestine@mail.com', 'StopAgressionInGaza', ...}
        'MemberRejected' : {'BenyaminShaitanNyaou', 'HamasHezbollah', ....}
}
```

Le code http en cas d'échec

7. Suprimer un utilisateur d'un groupe

Description : Le travail ici consciste à supprimer un membre d'un goupe donnée par l'utilisateur admin à paritir de son nom au niveau de la base de donnée.
Nom de la route : ???
Input : Id du groupe sélectionné et de l'utilisateur à suprimer
output : Le code http

9. Joindre un groupe en tant que membre

Description : Le travail ici consciste à creer dans la database une demande d'adhésion (Type DA dans la table Adhésion Request) adressée à l'admin du groupe que user aimerais rejoindre. Son status demeure temporairement 'Pending' Tant que L'admin n'a pas validé ou refusé sa demande.
Nom de la route :???
Input : GroupeId

Output : le code http

10. Obtenir tous les groupes disponibles et qui coincident avec l 'input

Description : Le travail ici consciste à trier tous les groupes externes semblables au mot ou à l'expression entrée par l'utilisateur.
Nom de la route :???
input : User input
output : L'api retournera tous les groupes dans qu"elle a trouver stocké dans une varaible 'SortedGroups'

*Attendance launching

1. via code qr

Description : Le travail ici consciste à crrer une cession dans le groupe sélectionné par user puis à y générer des code qr uniques. Tant que l'admin stop pas le process, ces derniers doivent se renouveler toute les 20s pour des raisons de sécurité et sont destinés à être scanné par le membres du groupe afin qu'ils soient marqués présent.
Nom de la route :
Input : l'id du groupe et nom de la cession lancée
output : Pour l'output le back doit renvoyer des code (on choisira le nombre de charactère) au front et qui seront transcrit et affichés sous forme de qr code. Il faux que ces codes se renouvelent toute les 5s pour empêcher toutes tentative de fraude. Donc tant que l'admin ne stop pas, on aura :

{
'Qrcode' : {'l mg55gz4sht4dt4j4drtrxtsjwg4rtj7jtresu@',
'CongoleseBabyaimeraissabblMwashishukali@', ?.....}
}

2.Via camera

/*
Description : Ici, le but ici est de procéder à une authentification massive des membres d'un groupe via la camera de l'admin. Nous songeons à ne rendre cette fonctionnalité accessible que sur mobile (la camera du pc n'etant pas souvent claire et adaptée à ce genre de situation). Ma théorie a ce sujet est que nous récupérons des séquences d'images et de vidéos puis les envoyos au back. Ce dernier les analysera et marquera présent toutes les personnes qui y sont. Des notifications sont progressivement envoyés aux membres marqués présents à la cession.
Nom de la route :???
Input : Id du groupe sélectionné et nom de la cession lancée , séquences de vidéos de 10s.
Output : création d'une nouvelle session avec session name et le marquage présent à tous ceux qui étaient présents. Crée une notification dès que quelqu'un est marqué présent.
*/

*Planification d'une session

1. Création d'une cession planifiée:

Description : Le travail ici consciste à creer dans la base de donnée, une session à l'admin d'un groupe à partir du nom de la session, du groupe concerné.
Nom de route:???
Input : {GroupId : 1, SessionName : 'Tépitech', Time : "08:15", date: "17/06/24", group: "Tatakaé", gateway: 1}
output : le code http.

2. Obtention de toutes les cessions d’un utilisateur ;

Description : Le travail ici consciste à récupérer toutes les sessions d’un utilisateur donnée.
Nom de la route :???
Input : RAS
Output : L’api retourne toute les sessions trouvées dans une variable Sessions.

3. Modification des paramètres d’une cession planifiée mais qui n’est pas encore réalisée

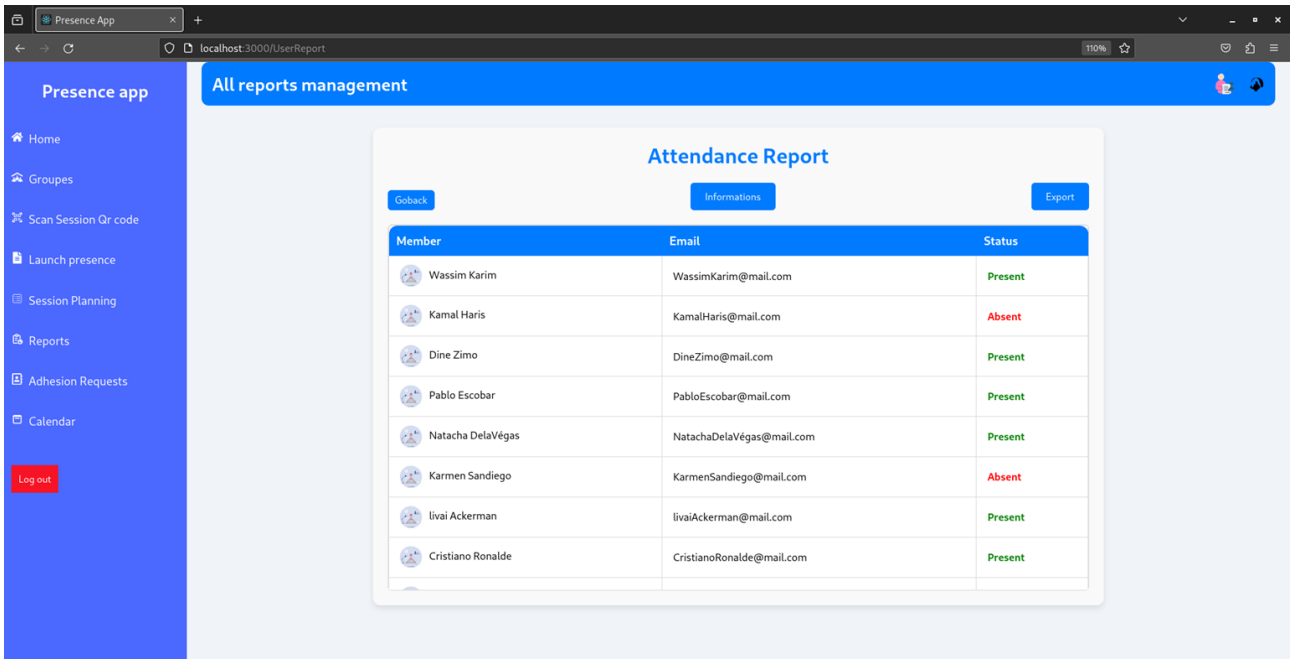
Description : Le travail ici consciste à modifier un ou plusieurs éléments (la date ou le jour, ...) au niveau d’ une session donnée.
Nom de la route : ???
Input : l’id de la session + l’élément ou les éléments modifié(s).
Output : le status ou le code http
Interdiction: Il ne peux pas modifier mette une date et heure passée

4. Suppression d’une cession par un utilisateur

Description : Le travail ici consciste à supprimer une session chez un utilisateur donné.
Nom de la route :???
Input : SessionId
Output : le status ou le code http

*Raports de présences

A chaque cession de présence achevée ; un rapport de présence est créée dans la base de donnée et contient la liste des membres dans une premere colone, s’il leur status vis a vis de la session et leur mails



1. Obtention de toutes les sessions achevées

Description : Le travail ici consciste à récupérer toutes les sessions achevées d’un utilisateur donnée.
Nom de la route :???

Output : AchievedSessions

2. Obtention du rapport d'une session achevée

Description : Le travail ici consciste à récupérer le rapport d'une session achevée donnée.
Nom de la route :???
Input: SessionId
Output : le rapport en cas de succès dans une variable Repports et le status ou le code http en cas d'échec

3. Obtention des informations la session achevée

Input : SessionId
Nom de la route : ????
Output : Il retourne le résultat dans la variable Selected AchievedSessions

4. Generation d'un fichier exel téléchargeable pour le rapport

Input : ReportId
Nom de la route :????
Output : Un fichier exel contenant la liste des étudiants présents à la cession dans une variable DownloadableRepport.

*Adhesion request

1. Obtention de toute les demandes de user envers d'autres groupes;

Description : Le travail ici est de récupérer toutes les récentes demandes d'adhésions de user envers d'autres groupes.
Nom de la route :???
Input : RAS
Output : l'api retourne toutes les demandes d'adhésions de user envers d'autres groupes dans la Variable UserDA.

2. Obtention de toutes les demandes de user à un autre utilisateur afin que ce dernier adhere a son groupes

Description : Le travail ici est de récupérer toutes les récentes invitations de user envers d'autres utilisateurs.
Nom de la route : ???
Input : RAS
Output: l'api retourne toutes les Invitations de user envers d'autres utilisateurs envers lui dans une variable UserIVT.

3. Obtention de toute les requetes d'autres utilisateurs voulant rejoindre un des groupes de user

Description : Le travail ici est de récupérer toutes les récentes requetes d'adhésion en provenance d'autres utilisateurs et destinée un des groupes de user.
Nom de la route :????
Input : RAS
Output: l'api retourne toutes les demandes d'adhésions d'autres utilisateurs envers un des groupes de user dans une variable OtherDA

3'. Validation ou rejet de ces demandes

Description : Le travail ici est d'actueliser le status de la requête jusque là 'pending' ('rejected' / 'accepted') en fonction du choix de user.
Nom de la route : ????
Input : AdhesionRequestId , IsAccepted (True/False)
Output: le status code http

4. Obtention de tous les requetes émises par d'autres groupes et demandant l'adhésion de user

Description : Le travail ici est de récupérer toutes les récentes Invitations en provenance d'autres groupes et destinée à user.

Nom de la route : ????

Input : RAS

Output: L'api retourne toutes les récentes Invitations en provenance d'autres groupes et destinée à user dans la variable OtherIVT

4' Validation ou rejet de ces demandes

Description : Le travail ici est d'actueliser le status de l'invitation jusque là 'pending' ('rejected' / 'accepted') en fonction du choix de user.

Nom de la route :???

Input : AdhesionRequestId, IsAccepted (true / false)

Output: le status ou le code http

*Scanning du QRCode

Description : Le travail ici consciste à vérifier le code renvoyé correspond belle et bien à une session en cours et marqué l'interressé présent à cette cession.

Nom de la route :

Input: Code + Time

Output : le rapport en cas de succès et le status ou le code http en cas d'échec;

*Notifications

Description : Le travail ici consciste à creer des notifications pour l'utilisateur en fonction des différentes situations qu'il rencontre. Ainsi on distingue :

1. Si une de ses demandes d'adhésion est acceptée ;
2. Si un groupe lui envoie une invitation
3. Si un utilisateur lui envoie une demande d'adhésion pour un de ses groupes
4. 10 min avant le début de la présence et lorsque le temps de présence est venue.

Certaines notification du 4. peuvent être à charactère externes mais les autres restent à characteres internes.

1. Obtenir les dernieres notifications des 72heures

Description : Ici il s'agit de retourner toutes les notifications des dernières 72heures.

Nom de la Route : ???

Input : RAS.

Output: Chaque situation sera représenté par des types.

- AR : Si un utilisateur lui envoie une demande d'adhésion pour un de ses groupes.

{data: {name: 'Bola Accambi', Mygroup: 'Ladybug'}, redirecto: '/AdhesionRequest', type: 'AR', date: 10/08/2018, Time: 10:30, id: 1},

- IVT : Si un groupe lui envoie une invitation

{data: {IVTgroup: 'Bien être'}, redirecto: '/AdhesionRequest', type: 'IVT', id: 1},

- JGr : Si une de ses demandes d'adhésion acceptée ;

{data: {name: 'shaley', JGrgroup: 'Bien être'}, redirecto: '/AdhesionRequest', type: 'JGr', id: 1},

- OTH : Others ntotification

{data: {msg: 'Vous avez une cession dans 2 min'}, redirecto: '/Calendar', type: 'OTH', id: 1},

*Logout

Description : Le travail ici consciste à déconnecter user
Nom de la route : ???
Input : RAS
Output : le status ou le code http résultant

*UserProfile

1. Obtenir le Profile de l'utilisateurs

Description : Le travail ici consciste à récupérer le profile de user.
Nom de la route : ???
Input: RAS
Output : Retourne les informations de l'utilisateur dans la variable UserProfile

2. changer une ou des information(s) dans le profile de l'utilisateur

Description : Le travail ici consciste à modifier une donnée dans le profile de user
Nom de la route : ???
Input : RAS
Output : Retourne le nouveau profile ou le code http

*Others

1. Les notification externes

Description : Un utilisateur doit pouvoir activé et ou désactiver les notifications externes sur l'application. Donc le boulot ici est de le renseigné dans la base de donnée
Nom de la route : ???
Input: Isactivated (true / false);
Output : le code http

2. Les notifications et rappels via sms

Description : Un utilisateur doit pouvoir activé ou désactiver les rappels via sms sur l'application. Donc le boulot ici est de le renseigné dans la base de donnée. Cela n'est évidement possible que si user a préalablement enregistré un numéro de telephone sur l'app.
Nom de la route : ???
Input: Isactivated (true / false);
Output : le code http

3. Enregistrement d'un numéro pour les sms

Description : Le travail ici est d'attribuer un numero de tel à user dans la database
Nom de la route : ???
Input: +22825256647
Output : le code http

4. Modification d'un numéro pour les sms

Description : Le travail ici est demodifier le numero de tel à user dans la database par un nouveau qu'aura saisit le user
Nom de la route :???
Input: +22825879641

Output : le code http

5. Inviter un ami à utiliser la plateforme

Description : Le travail ici est de pouvoir permettre à l'utilisateur d'ajouter un ami si ce dernier est présent dans ses contacts téléphonique ou whats'app.