Cahier de conception.

GoGo-Night Projet.

Sara Hadj-ali, Daniela Pistol, Thileli Toursal

Sommaire du cahier de conception de GoGo-Night

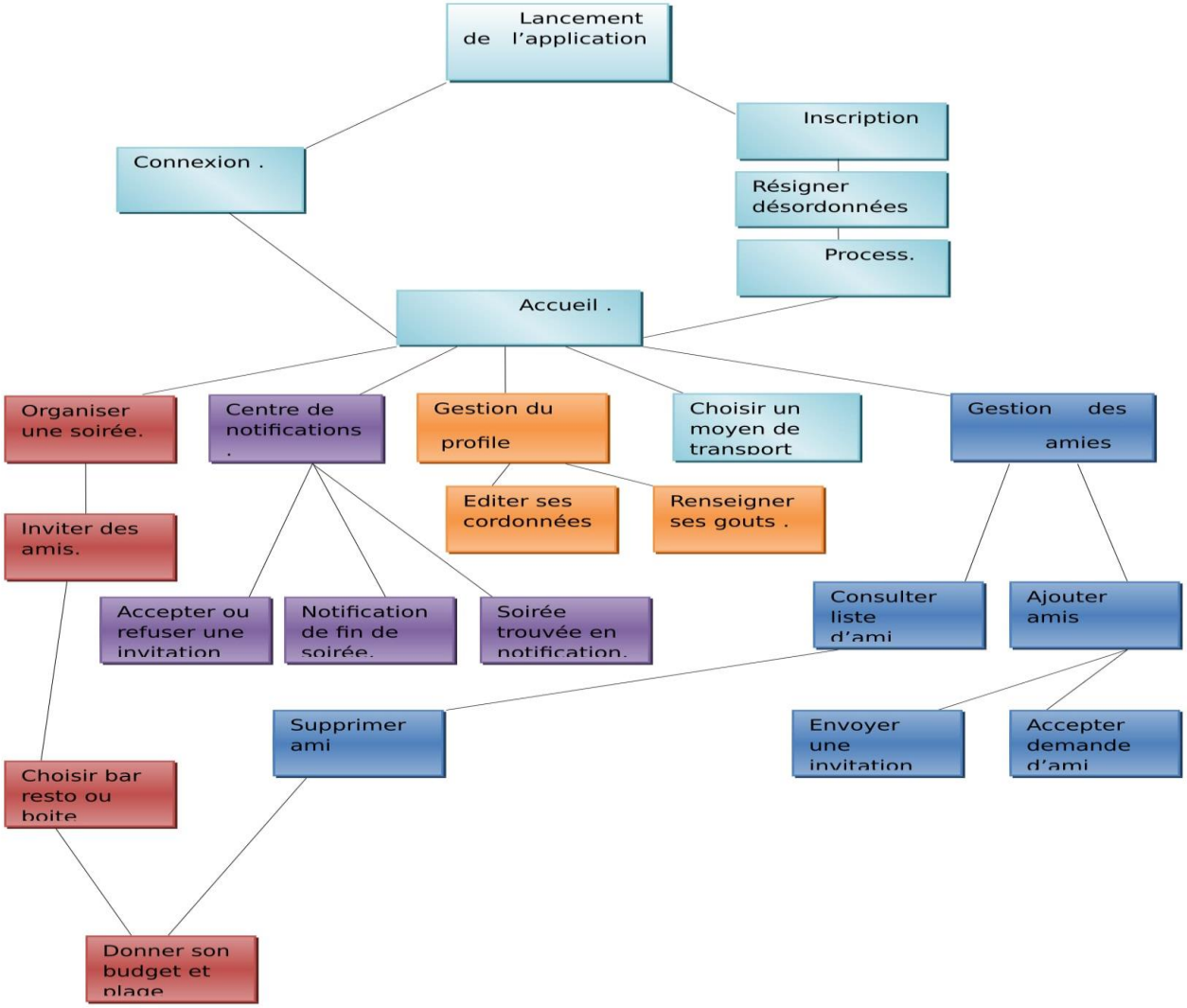
1 .Introduction

Le document présent est la suite de la phase d'analyse et traite des spécificités techniques de l'application. Il permet de rapporter les résultats de la recherche lors de la phase de conception où toutes les informations relatives à l'implémentation sont spécifiées.

Nous demandons aux codeurs de prendre soin de lire dans la totalité le document ci-présent et de respecter les règles qui y sont définies.

Nous nous engageons auprès du client sur le fait que toutes les spécifications rédigées dans le cahier des charges sont parfaitement retranscrites dans le cahier de conception ici présent.

Fig.1 : Schéma explicatif de la structure de l'application (repris du cahier des charges)



2. Contraints à vérifier

L'application prendra en compte la localisation de chaque personne du groupe. Afin de déterminer une zone qui convient à tout le monde, l'algorithme calcule le centre du cercle circonscrit de diamètre : les deux personnes du groupes les plus éloignées, pondéré par le nombre de personnes présentes dans le demi cercle perpendiculaire à la droite correspondant au diamètre choisi. Ce point sera utilisé pour la recherche du point de rendez-vous au restaurant.

Les goûts de chacun

- Afin de choisir le restaurant, l'application oubliera tout d'abord les restaurant qui correspondent à ce que les différents personnes du groupe n'aiment pas . Ensuite, le choix s'effectuera sur le goût présent dans la majorité des usagers dans le groupe de la soirée.

- S'il s'avère qu'il n'y a aucun point commun entre les goûts des différentes personnes du groupe, l'algorithme choisira le restaurant ayant la meilleure note parmi tous les restaurants correspondant aux goûts de chacun des participants de la soirée

Les moyens de transport

- Les trajets de groupe de la soirée ne peuvent pas être choisis par un utilisateur non organisateur de la soirée.

La méthode du choix des lieux

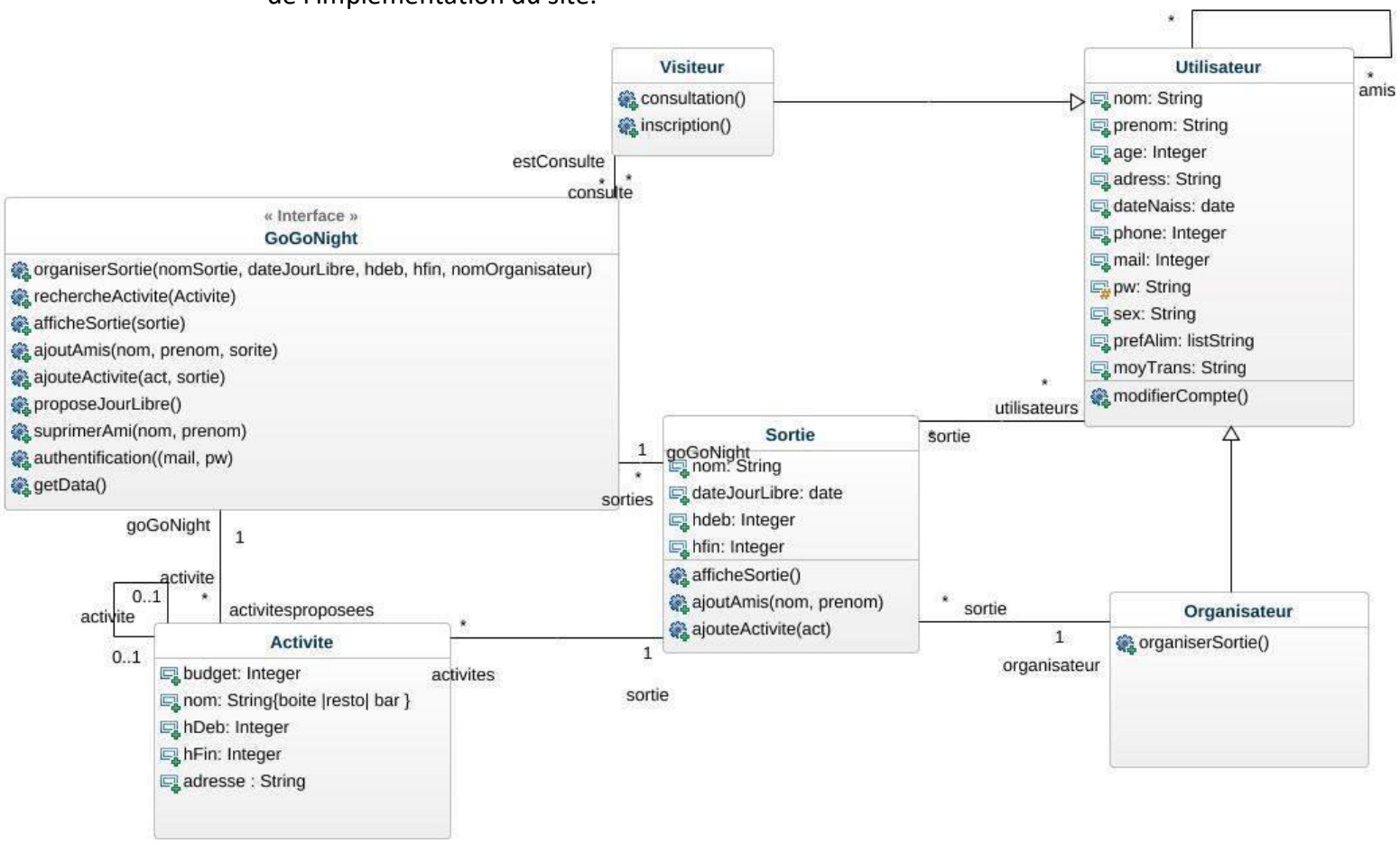
Après avoir quitté le premier lieu (bar, restaurant ou boite), les usagers du groupe ne doivent pas effectuer un trajet d'une durée supérieure à 20 min pour aller faire une deuxième activité, de même pour la troisième activité.
Ainsi, nous limiterons la recherche du restaurant à 6 km autour du bar et à 6 km autour du restaurant .
Lors du choix du lieu, l'application liste tout d'abord les lieux correspondant au goût choisi par la majorité. Ensuite, elle sélectionne les endroits correspondant au budget général du groupe.
L'application devra choisir parmi cette liste l'endroit ayant été le mieux noté par leavis Google.

3. Diagramme de classe de GoGo-Night

➢

Notre diagramme de classe contient 6 classes pour le moment (éventuelles modifications). On a choisi de faire

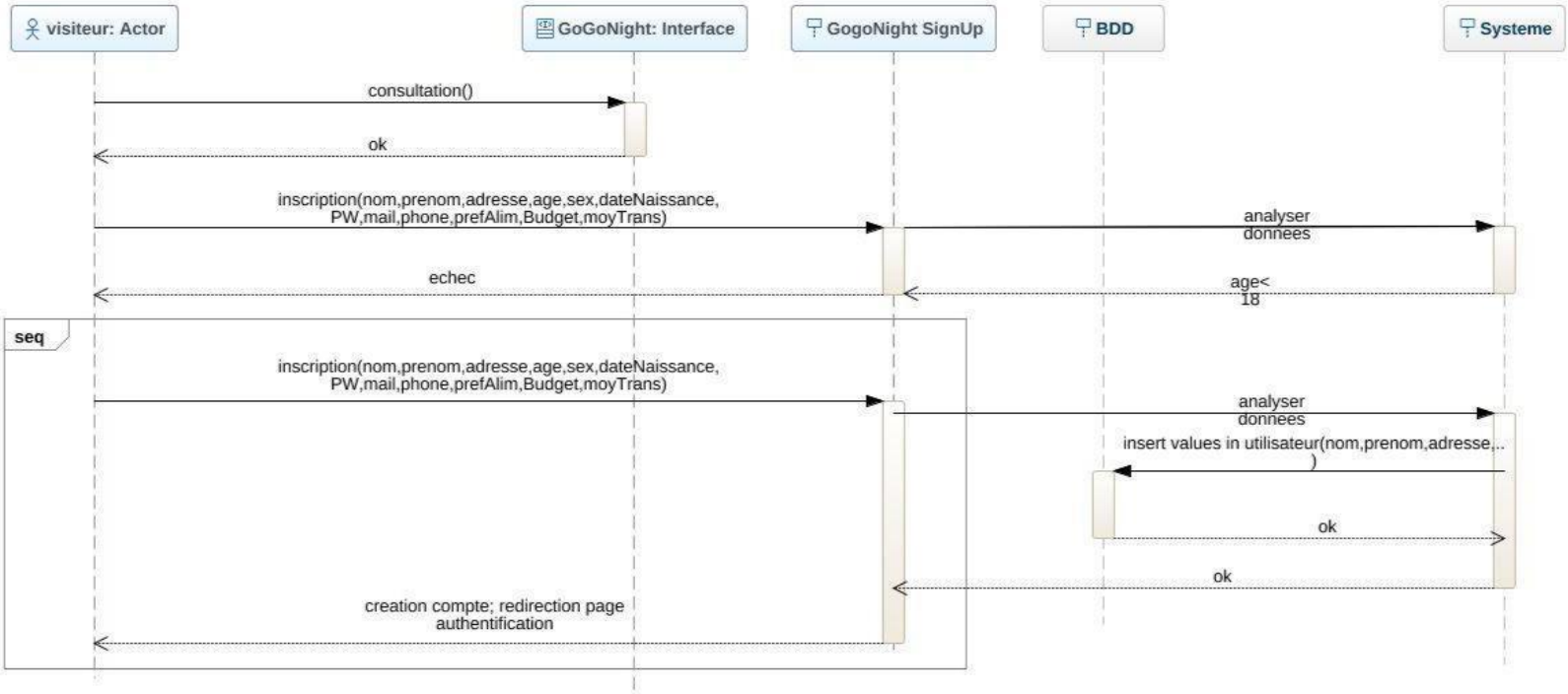
- une classe pour les visiteurs qui peuvent consulter et s'inscrire a GoGo-Night - plusieurs visiteurs peuvent consulter plusieurs fois l'interface de notre site.
- Une classe pour les utilisateurs avec les informations données à l'inscription les attributs - et ils peuvent modifier leur profil. Un organisateur peut créer plusieurs sorties.
- une classe sortie, avec les détails importants d'une sortie - attributs - et des fonctions pour ajouter un ami, afficher le planning et aoute d'une activité à l'évènement. Une sortie a un seul organisateur et peut contenir plusieurs activités et utilisateurs.
- une classe activité ou chaque activité (resto, bar, boite) va avoir un budget, une adresse, une heure de début et une heure de fin.
- et finalement l'interface GoGo-Night qui continent les fonctions ou les procédures de l'implémentation du site.



a. Inscription

- **Diagramme de séquence**

Soit un visiteur de site GoGo-Night. Il consulte la page et il choisit de s'inscrire. Sur la page de "SignUp" (exemple maquette page 7). Il s'inscrit en fournissant les détails comme son nom, prénom, adresse, âge, sexe, date de naissance, mot de passe, mail, phone, ses préférences alimentaires, son budget et les moyens de transport qu'il préfère. L'inscription échoue (ca peut être le cas ou le visiteur est mineur). Un autre visiteur majeur essaye de s'inscrire. Conséquence: son compte est créé.



PseudoCode :

Nouveau utilisateur u;

Si u.nom, u.prenom, u.adresse, u.age, u.dateNaiss, u.pw, u.mail, u.prefAlim, u.moyTrans !=null et u.age>=18

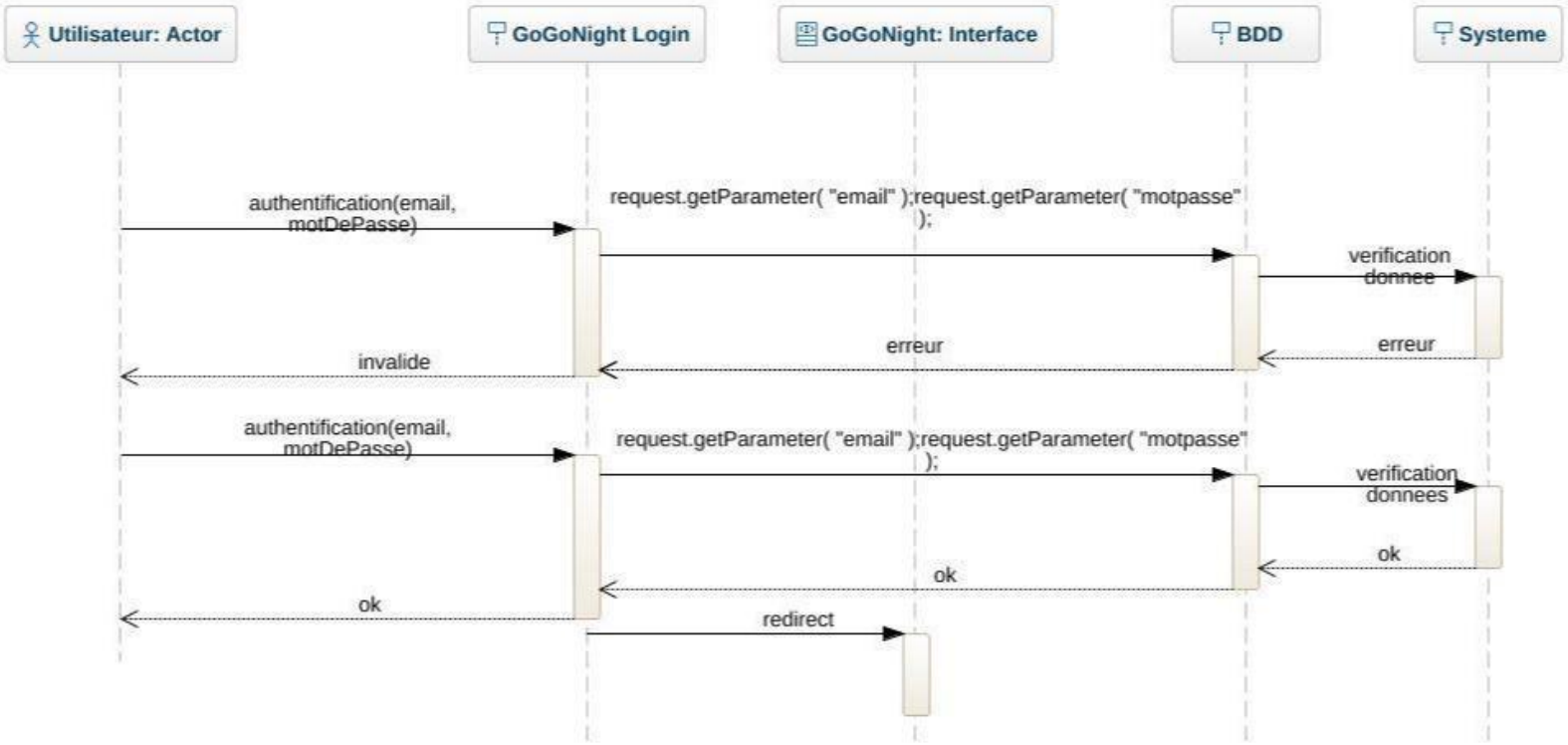
Alors compte créer, utilisateur ajoute dans la Base de données et redirection sur la page d'authentification;

Sinon redirection sur la page d'inscription + message d'erreur;

b. Authentification

- **Diagramme de sequence**

Soit u un utilisateur de notre site GoGo-Night. Il veut s'authentifier. Il fourni le mail et le mot de passe, mais un des deux est mal écrit. L'application répond "invalide". Il ressaye en fournissant cette fois le bon mail et le bon mot de passe. L'authentification, c'est bien passe et l'utilisateur est redirectionne sur la page de GoGo-Night.



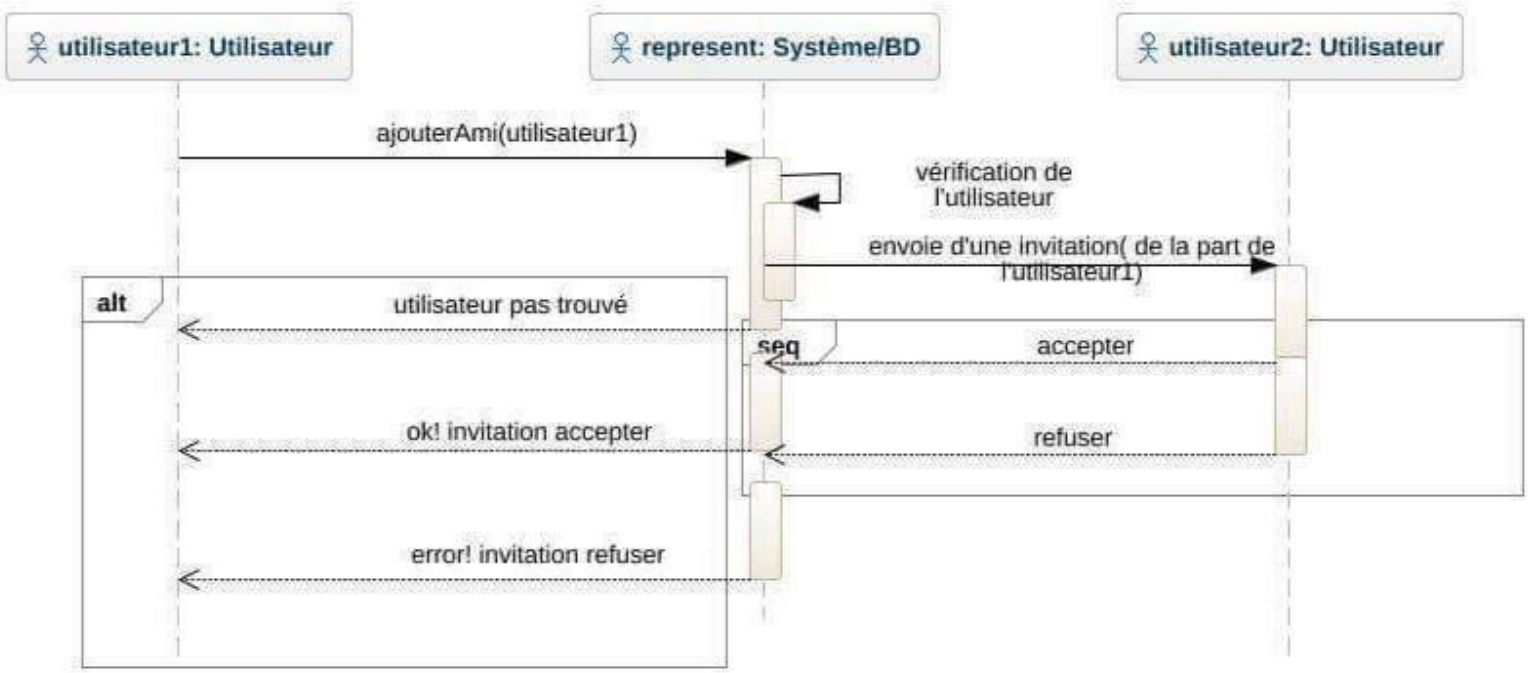
Pseudo-code:

Nouveau utilisateur u;
String paramEmail = request.getParameter("email");
String paramMotDePasse = request.getParameter("motdepasse"); Si
    les deux correspond au meme utilisateur et
    sont correctement saisis alors
        authentification reussi
        redirection sur l'interface GogoNight;
Sinon
    redirection sur page de authentification
    message error;

c. Ajouter un ami

- **Diagramme de sequence**

Pour ajouter un ami, l'utilisateur va chercher d'abord la personne qui veut ajouter à sa liste d'amis, ensuite le site vas vérifier si la personne recherchée est un utilisateur de notre site, si c'est le cas, il va envoyer une invitation à cette personne, la personne qui reçoit l'invitation elle peut accepter comme refuser cette invitation.



Pour le pseudo-code:

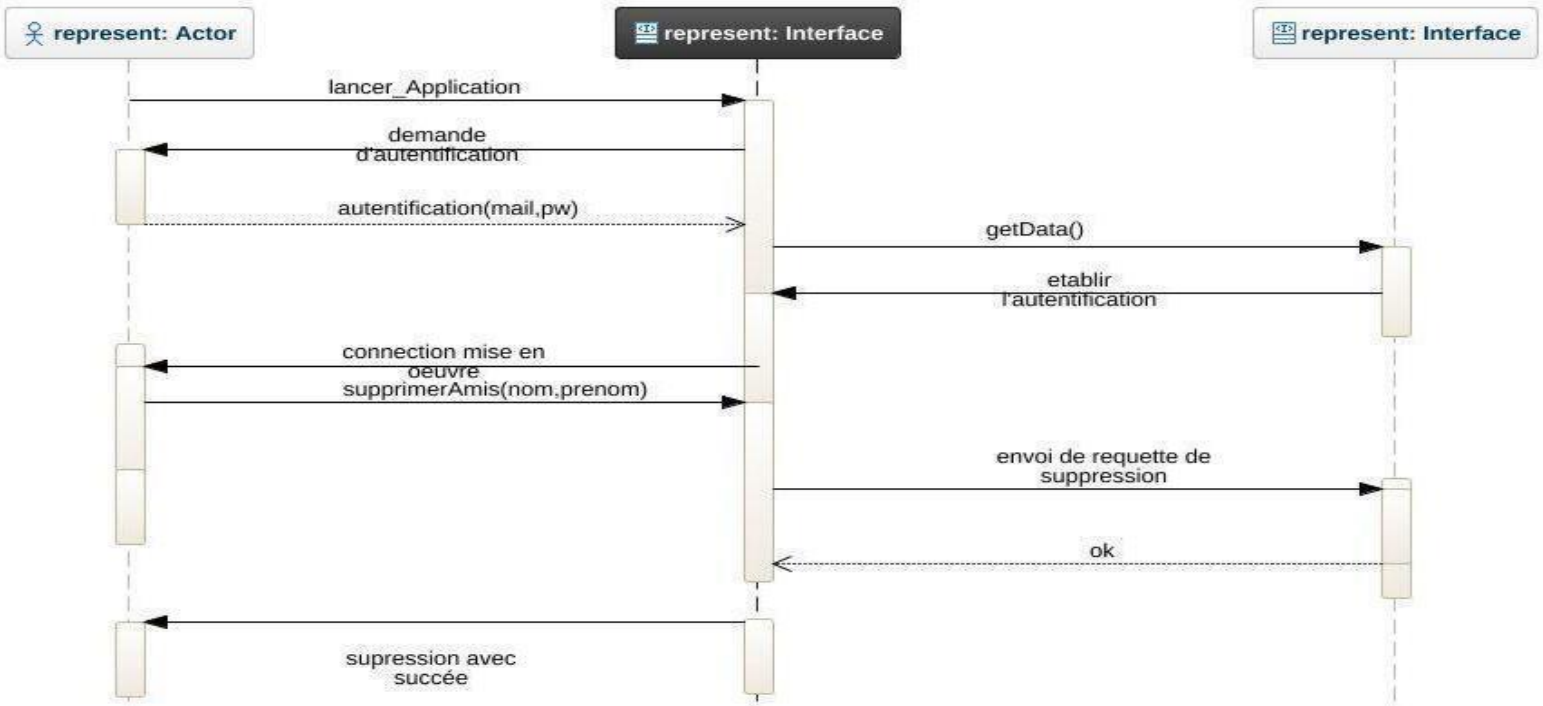
```
 utilisateur u;
si
  ajouterAmi(u1)
alors
      envoiInvitation(u)
       si le retour est « accepter » alors le statut = « amis »
      sinon le retour est « décliner » alors le statut = « pas amis » ;
```

d. Suppression d'un ami

- **Diagramme de séquence**

La suppression va se faire ainsi:

Un utilisateur veut supprimer un ami ou plusieurs amies de sa liste. Tout d'abord, celui-ci lance l'application GoGoNight qui demande une authentification de ce dernier avec son mail et mot de passe, GoGoNight fait demande au system avec un getData() pour retrouver les informations sur l'utilisateur, le system établi l'authentification et une connexion est mise en œuvre. Puis l'utilisateur décide de supprimer un de ces amis ou plusieurs d'entre eux, en lançant la fonction supprimerAmis(nom, prénom) qui prend un nom et prénom d'utilisateur en paramètres. GoGoNight envoi une requête au système et celui-ci répond par une confirmation et supprime cet utilisateur.



Pseudo-code:

utilisateur u;

supprimerAmis(nom,presnom)

String paramEmail = request.getParameter("nom");

String paramMotDePasse = request.getParameter("prenom");

Si

les deux paramètres correspond au même utilisateur qui est dans la liste des amis de l'utilisateur et sont correctement saisis

alors

la suppression reussi : redirection sur l'interface GogoNight;

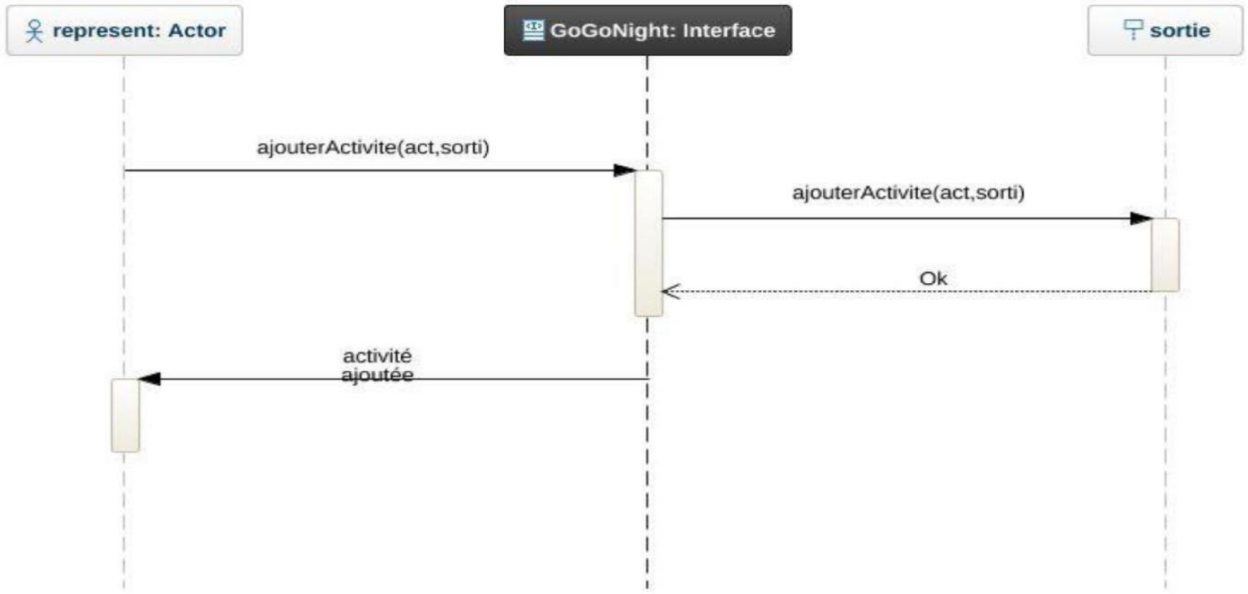
Sinon

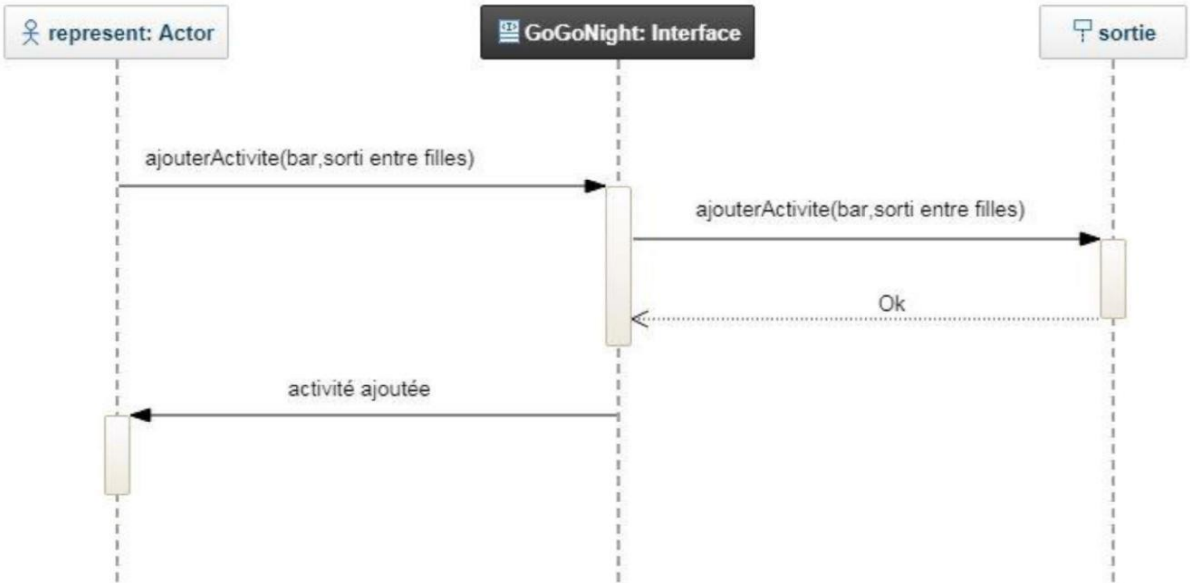
affichage de message error « aucun ami de ce nom sur votre liste d'amis

e. Ajout d'une activité a une soirée

- **Diagramme de sequence**

L'ajout se fait par un utilisateur avant d'avoir était authentifié celui-ci ajoute avec la procedure ajouterAmi(act,sorti) prenant en paramètre le act : activité étant bar ou restaurent ou boite, et sorti étant le nom de la sorti défini par 'utilisateur ,puis une sortie est ajoutée avec un acquittement au retour et l'activité est ajouté avec succès.





Pseudo-code:

utilisateur u;

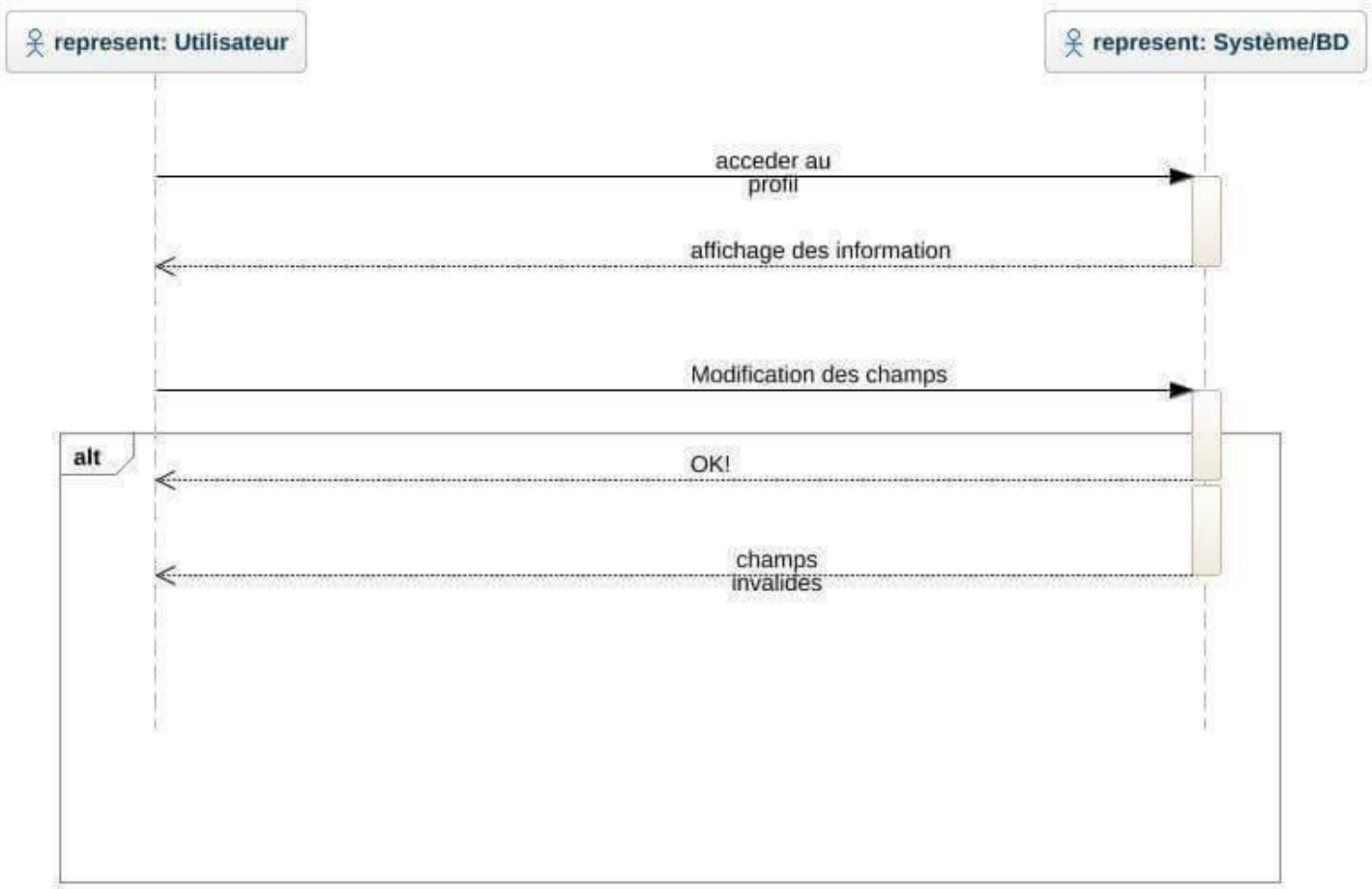
ajouteActivite (act,sortie) ;

renvoi de message pour confirmer que l'activité est ajoutée.

f. Consultation et modifications profil

Un utilisateur qui souhaite consulter son profil, il doit se rendre sur la rubrique profil de GoGoNight pour accéder à son profil et là se trouve toutes ses informations personnelles(nom, prénom, adresse, etc.), si il veut modifier ses informations, il clique sur le champ(case) qu'il veut modifier et il insère la nouvelle valeur, le site vérifie si l'information est valide, si le champ est valide on enregistre les modifications, sinon on rend un message d'erreur : [valeur] non valide.

- **Diagramme de séquence**



Pseudo-code :

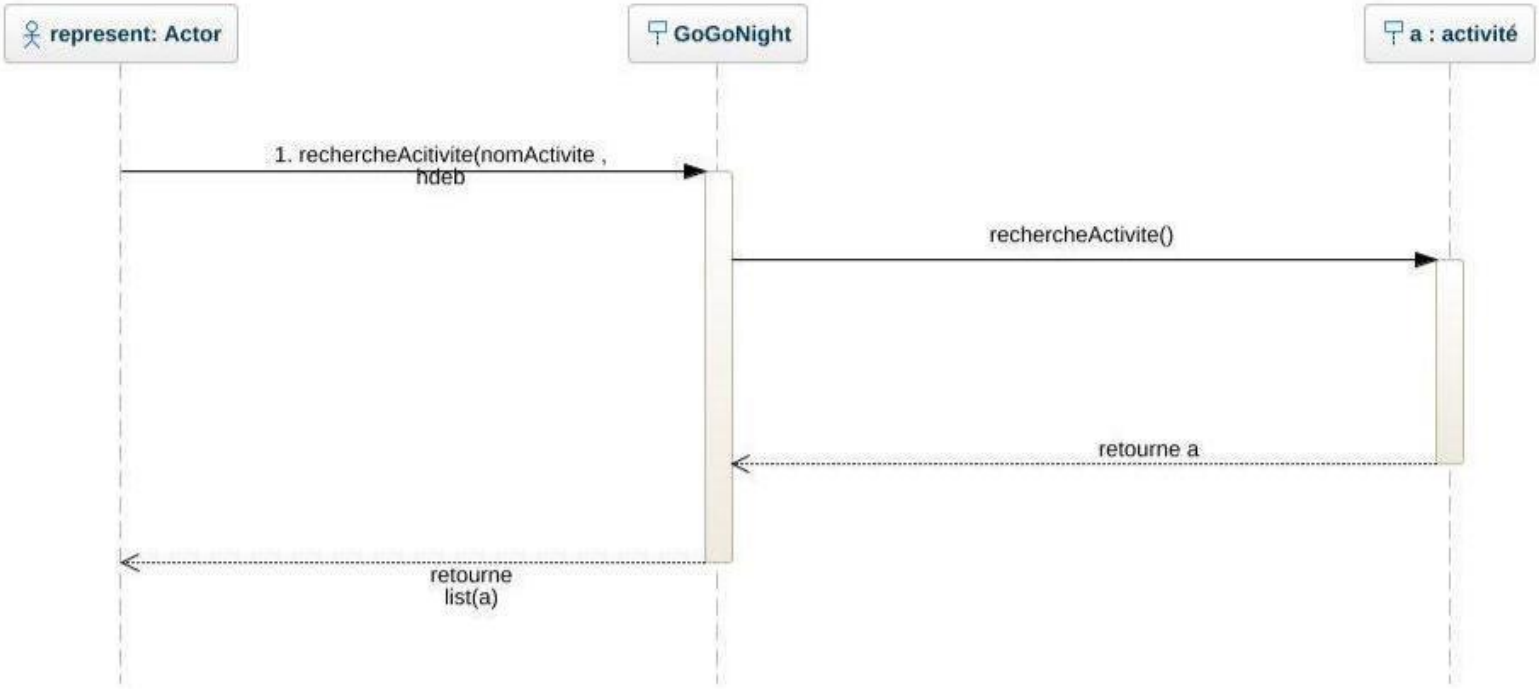
```
Utilisateur u ;
consulterProfil(u)
 u.nom = u.getNom()
u.prenom =u.getPrenom()
u.age=u,getAge()
u.adresse =u.getAdresse()
u.age=u.getAge()
u.telepone =u,getTelephone()
u.email=u.getEmail()
u.motDePasse=u,getMotDePasse()
u.preferencesAlimentaires =u.getPreferencesAlimentaires()
u.moyenDeTransport=u.getMoyenDeTraansport()
u.dateDeNaissance =u.getDateDeNaissance()
u.getListeAmis()
modificationProfil(u)

modifierProfil(u)
 u.nom = u.setNom()
u.prenom =u.setPrenom()
u.age=u,setAge()
u.adresse =u.setAdresse()
u.age=u.setAge()
u.telepone =u.setTelephone()
u.email=u.setEmail()
u.motDePasse=u.setMotDePasse()
u.preferencesAlimentaires =u.setPreferencesAlimentaires()
u.moyenDeTransport=u.setMoyenDeTraansport()
u.dateDeNaissance =u.setDateDeNaissance()
u.setListeAmis()
```

g. Recherche activite

- **Diagramme de sequence**

La recherche est faite à l'aide d'un mot clé, représenté ici par nomActivite, et éventuellement l'heure de début d'activité demandé (hDeb), afin de pouvoir retourner la liste des lieux ouverts. les résultats seront affichés selon la distance c'est-à-dire de plus proche au moins proche de la dernière activité avant me hDeb demandée. si aucune activité correspondante à la recherche est trouvée, un message s'affichera et invitera l'utilisateur à changer sa recherche.



Pseudo-code

rechercheActivite(string nomActivite , Date hdeb)

Array <string> listActivite = liste des restaurants, bars ou boite retourner par API Google si listActivite = null

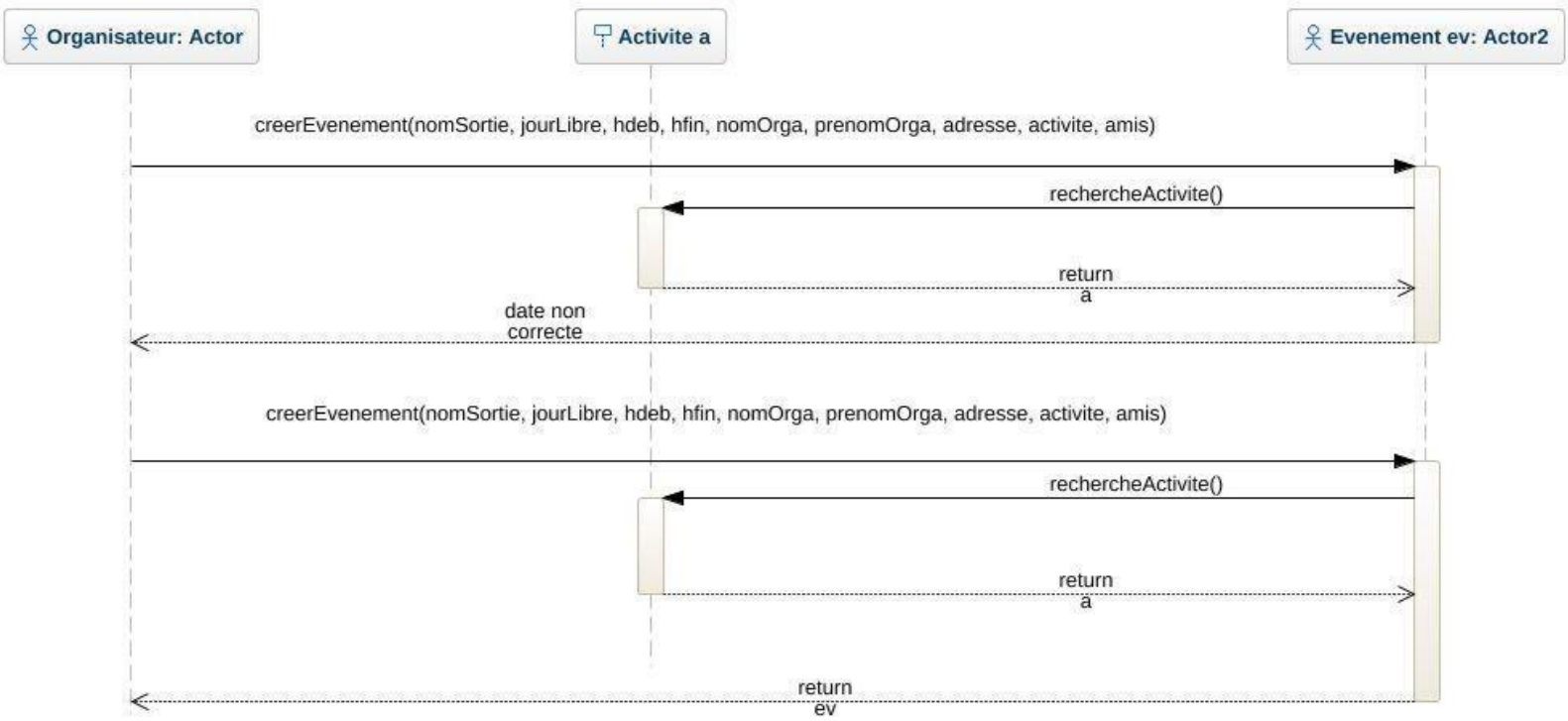
alors : afficher message « aucune activité de ce type, veuillez reessaye »

h. Créer événement

- **Diagramme de séquence**

Pour créer un événement, son organisateur doit fournir des informations comme une nomme de l'évènement, une date, son nom, prénom, adresse, les amis avec qui il veut

sortir et les activités qu'il veut faire. Un utilisateur veut créer un évènement, alors il dévient organisateur et il fournit les informations demandées. Il fournit une date inexistante alors une erreur est signale. Il ressaye avec les bonnes informations. L'évènement est alors créé.



Pseudo-code:

```
creerEvenement():
        afficheListeAmis()
        a = rechercheActivite();
        ajouteActivite(a);
        ArrayList<string> listeInvites = getAmis();
        inviterAmis();
        sortie s;
si sortie valide alors
        s.setNom=demandeNom();
        s.setDateJourLibre = demandeDate();
        s.setHdeb=demandeHdeb();
        s.setHfin = demandeHfin();
        organisateur o = setNomOrga();
        pour tout u Utilisateur dans listeInvites
                bool reponse= attentReponse();
                Si reponse = null alors supprimerInvite(u);
        Calculer lieux;
        Calculer hDeb, hfin;
Returner ev;
```

4. Modèle relationnel de données

Nous avons, à partir du diagramme des classes, établi le modèle relationnel de données qui représente la base de données que nous allons utiliser pour la gestion des différentes instances de classes.

Utilisateur (email , mdp, liste_soirees_invit, liste_soirees_particip, liste_amis, #id_itineraire, #id_profil);

Itinéraire(id_itineraire, durée, type_transport, liste_etapes, date);

inviter (email);

Soirée (id_soiree , nom, liste_Activites, liste_participants, heure_debut, heure_fin, theme, nb_participants, rdv_long, rdv_lat, #id_Organisateur);
Organisateur (id_organisateur , #id_soiree , #email);
Profil (id_profil , nom, prenom, adresse, mdp, liste_preferencesAlimentaire, #email);
activite (id_activite, type, nom, adresse, description, note, budget);
S'effectuer (id_soiree, id_lieu);
Participer(id_utilisateur, id_soiree);
Organiser (id_organisateur, id_soiree)

Dictionnaire des données

Utilisateur :

Attribut	**Définition de l'attribut**	**Type**	**Contraintes sur l'attribut**
Email	Adresse e-mail identifiant l'utilisateur	VARCHAR(30)	Doit être de format e-mail et unique et not null
Mdp	Mot de passe associé à l'e-mail	VARCHAR(30)	Compris entre 8 et 24 caractères
liste_soiree_invite	Liste des soirées auxquelles l'utilisateur est invité	VARCHAR(50)	Pas de contrainte
liste_soirees_participe	Liste des soirées auxquelles l'utilisateur participe	VARCHAR (50)	Deux soirées dans la liste ne peuvent avoir la même date
liste_amis	Liste des amis de l'utilisateur	VARCHAR(50)	

Organisateur :

Attribut	Définition de l'attribut	Type	Contraintes sur l'attribut
idOrganisateur	Identifiant de l'organisateur d'une soirée	INT	Unique et not null
idSoiree	Identidiant de la soirée organisée	INT	Clé étrangère référence à la table soiree
emailOrganisateur	Email de l'organisateur de la soirée	VARCHAR(30)	Clé étrangère référence à la table utilisateur

Profil :

Attribut	Définition de l'attribut	Type	Contraintes sur l'attribut
idProfil	Identifiant de profil	INT	Unique et not null
Nom	Nom de l'utilisateur	VARCHAR(30)	Compris entre 2 et 24 caractères
Prenom	Prénom de l'utilisateur	VARCHAR(30)	Compris entre 2 et 24 caractères
Age	Age de l'utilisateur	INT	Supérieur à 18 ans
Adresse	Adresse de domicile de l'utilisateur	VARCHAR(30)	Doit être écrite sous format d'adresse et doit être une adresse sur paris
dateNaissance	Day de naissance de l'utilisateur	VARCHAR(30)	Doit être au format Datre jj/mm/aaaa
Sex	Femme ou homme	VARCHAR(30)	Sex= femme ou sex=homme
Email	Email de l'utilisateur	VARCHAR(30)	Doit être de format e-mail et unique.
motPass	Mot de passe associé a l'email	VARCHAR(30)	Entre 8 et 24 caractères
moyenDeTransport	Moyen de tranport de l'utilisateur	VARCHAR(30)	Null possible
list_preferences_alim	Listes des préférences alimentaire de l'utilisateur	VARCHAR(30)	l'utilisateur doit au moins avoir une préférence
telephone	Numero de téléphone de l'utilisateur	VARCHAR(10)	Null possible, l'utilisateur peut ne pas renseigner son numéro de téléphone

Soirée :

Attribut	**Définition de l'attribut**	**Type**	**Contraintes sur l'attribut**
idSoiree	Identifiant de la soiréé	INT	Unique et not null
nom_Soiree	Nom ou theme de la soirée	VARCHAR(30)	Compris entre 3 et 24 caractères
Liste_activite_soiree	Listes des acitvités de la soirée	VARCHAR(30)	La taille de la liste doit être inférieur ou égale à 3

Activité :

Attribut	**Définition de l'attribut**	**Type**	**Contraintes sur l'attribut**
id_activite			
nomActivite	Nom de l'activité	VARCHAR(30)	nomActivite= bar || nomActivite= restaurant || nomActivite=boite

5. Présentation des composantes techniques

- **WEB**

  Notre projet est une application web, ca veut dire une application manipulable grâce à un navigateur web. On a choisi ce type de support car il est multiplatforme, tres pratique et facile a utiliser. De plus, l'interface est facile a faire, en utilisant CSS et BOOTSTRAP.

- **JAVA EE**

  Java EE est un outil nous permettant de construire une application web assez facilement. On va se baser sur le model MCV pour coder notre projet. On utilise aussi le JavaServer Pages (JSP) qui est une technique basée sur Java qui nous permet de créer dynamiquement du code HTML, XML, etc.

- **Google Maps Distance Matrix API**

  Le service Google Maps Distance Matrix API fournit les distances et les durées des trajets pour une matrice d'origines et de destinations en fonction de l'itinéraire recommandé entre les points de départ et d'arrivée nous accédons à Google Maps Distance Matrix API via une interface HTTP, avec des requêtes construites en tant que chaînes d'URL utilisant les paramètres origines et destinations en plus de notre clé d'API.

- **MySQL :**

  Notre application web sera mise en relation avec une base de données, afin de mettre en œuvre la gestion d'un système d'information MySQL sur les différentes tables qui stockeront les informations des classes réalisées en UML a l'aide de Genmymodel.

- **Google Places API Web Service**

  Le service permet de fournir la fonctionnalité de saisie semi-automatique pour les recherches géographiques textuelles et renvoie des lieux tels que des entreprises, des adresses et des points d'intérêt au fur et à mesure de la saisie par l'utilisateur.

- **Genmymodel :**

  On a utilisée Genmymodel pour la conception de notre diagramme de Classe UML vue précédâmes

Toutes les clefs des API seront communiquées par mail aux développeurs.

6. Sources:

Divers cahier de conception fournis par les proches exemple de cahier de conception de universitaire de Sonia Benbakli.