Cahier de conception.

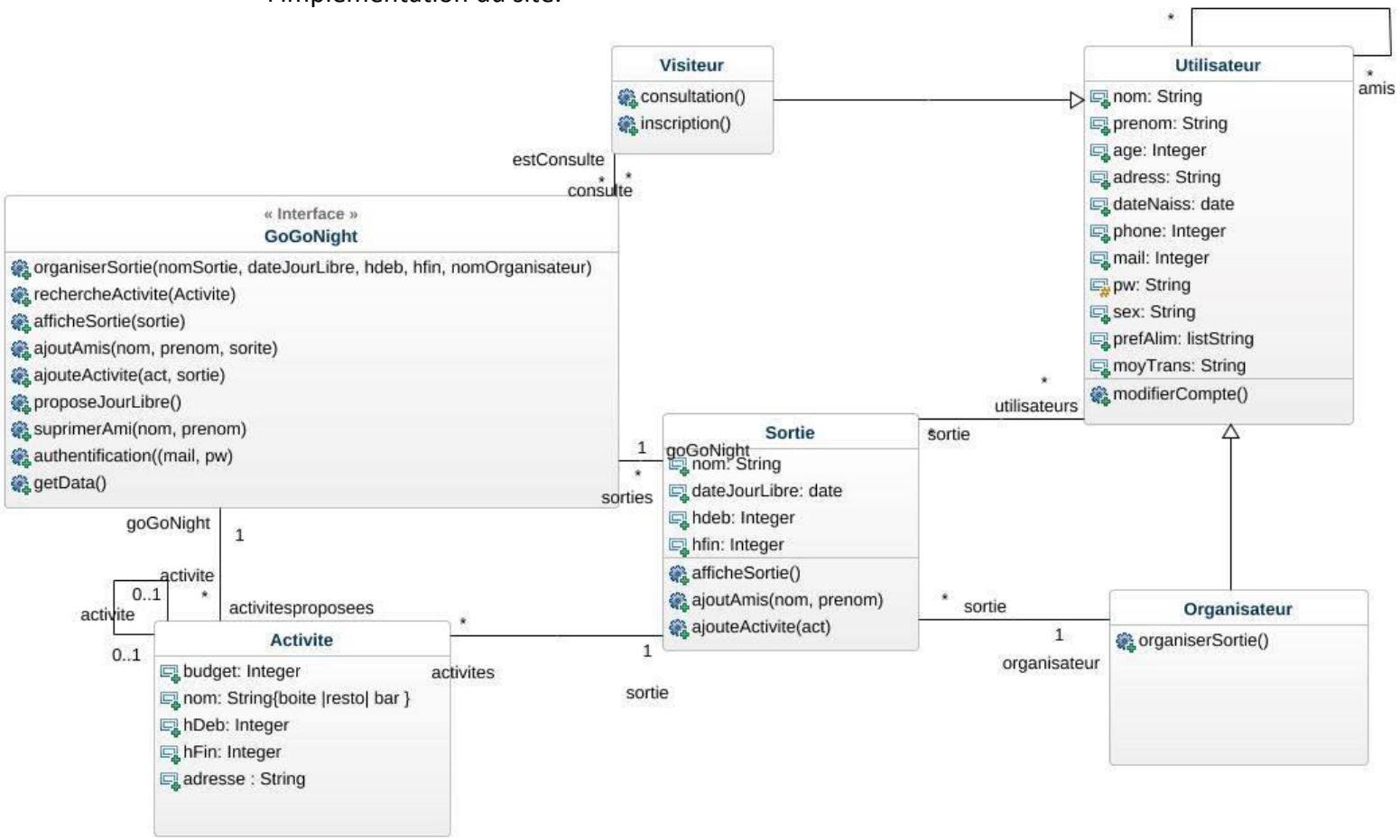
GoGo-Night Projet.

Sara Hadj-ali, Daniela Pistol, Thileli Toursal

Sommaire du cahier de conception de GoGo-Night

Diagramme de classe de GoGonight

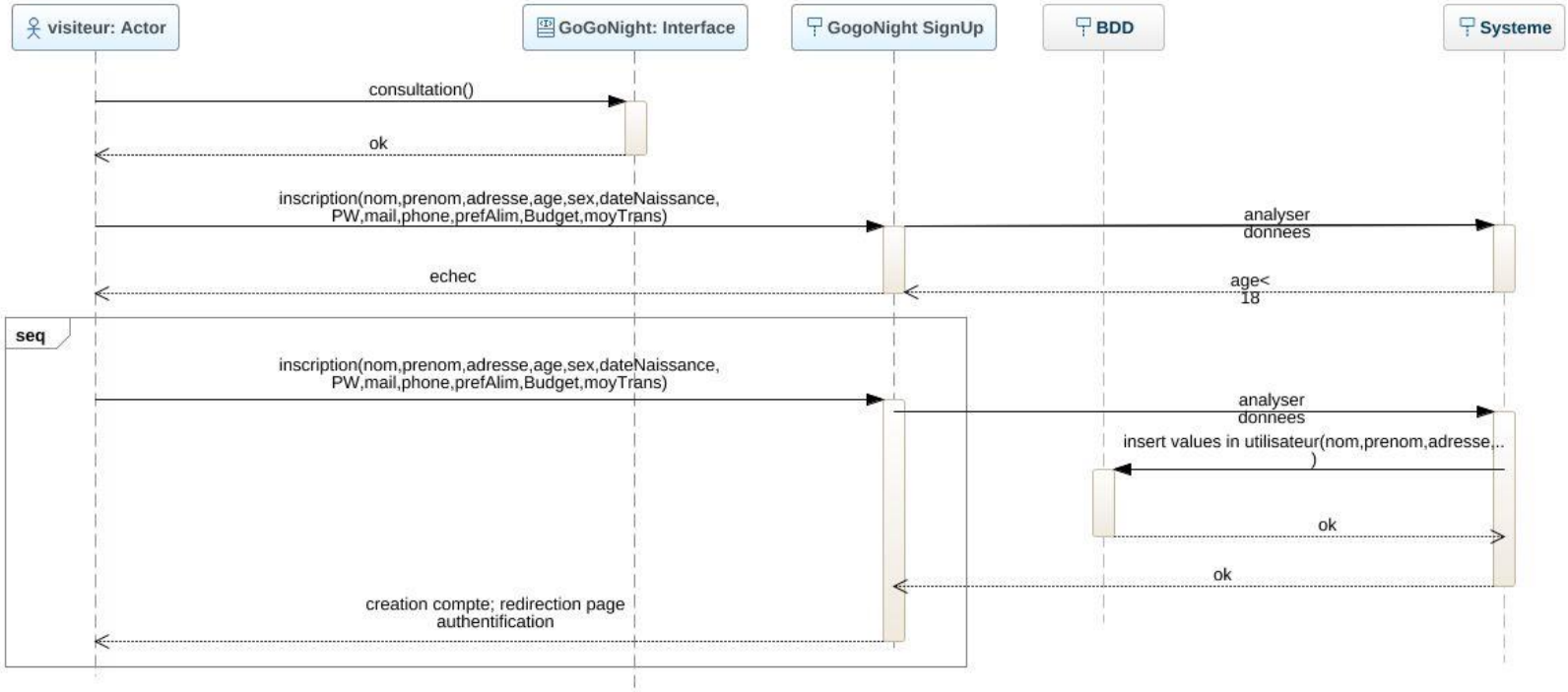
- Notre diagramme de classe contient 6 classes pour le moment (éventuelles modifications). On a choisi de faire
  - une classe pour les visiteurs qui peuvent consulter et s'inscrire a GoGo-Night - plusieurs visiteurs peuvent consulter plusieurs fois l'interface de notre site.
  - Une classe pour les 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.



I. Inscription

- **Diagramme de sequence**

  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, age, 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

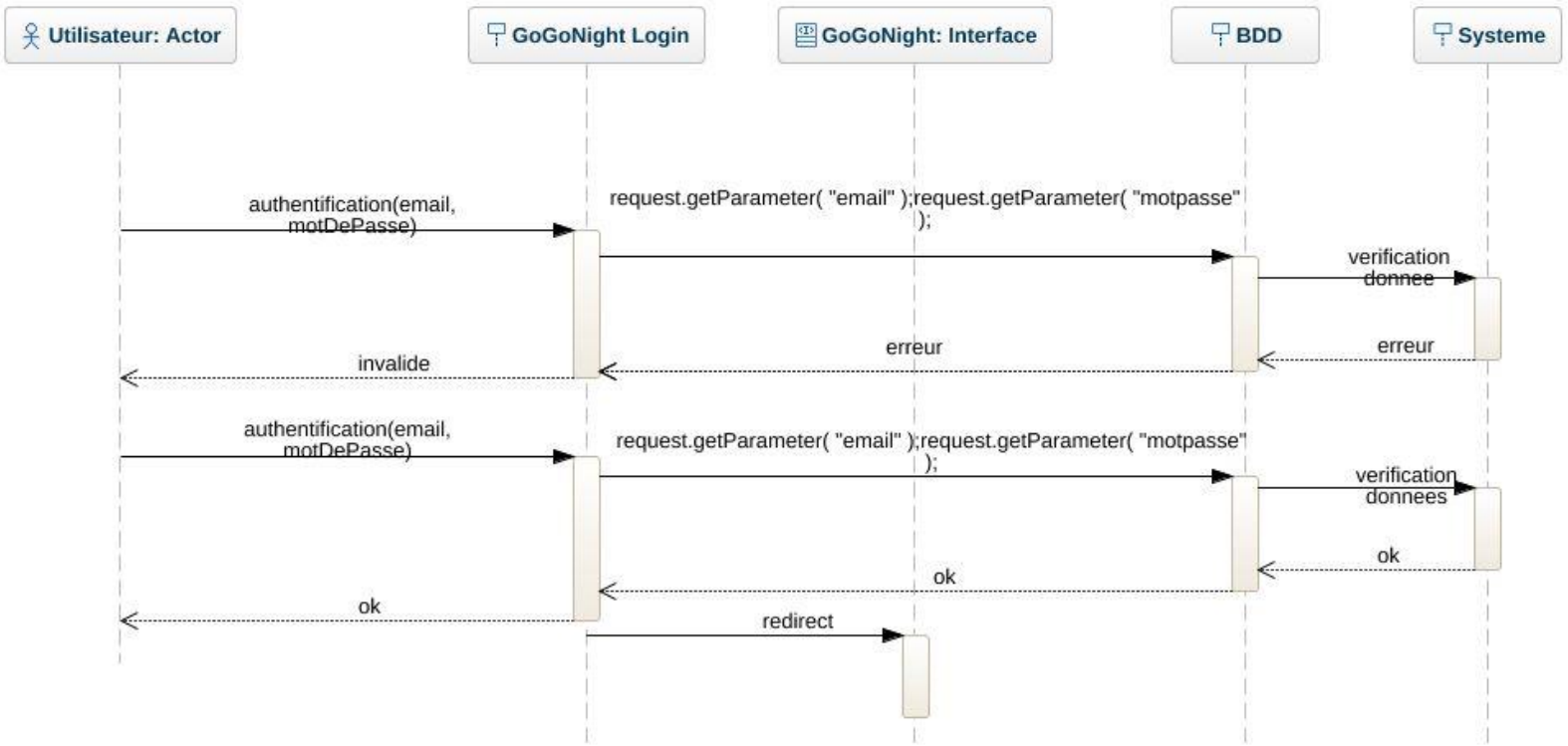
Alors compte creer, utilisateur ajoute dans le bdd et redirection sur la page d'authentification;

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

II. 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.



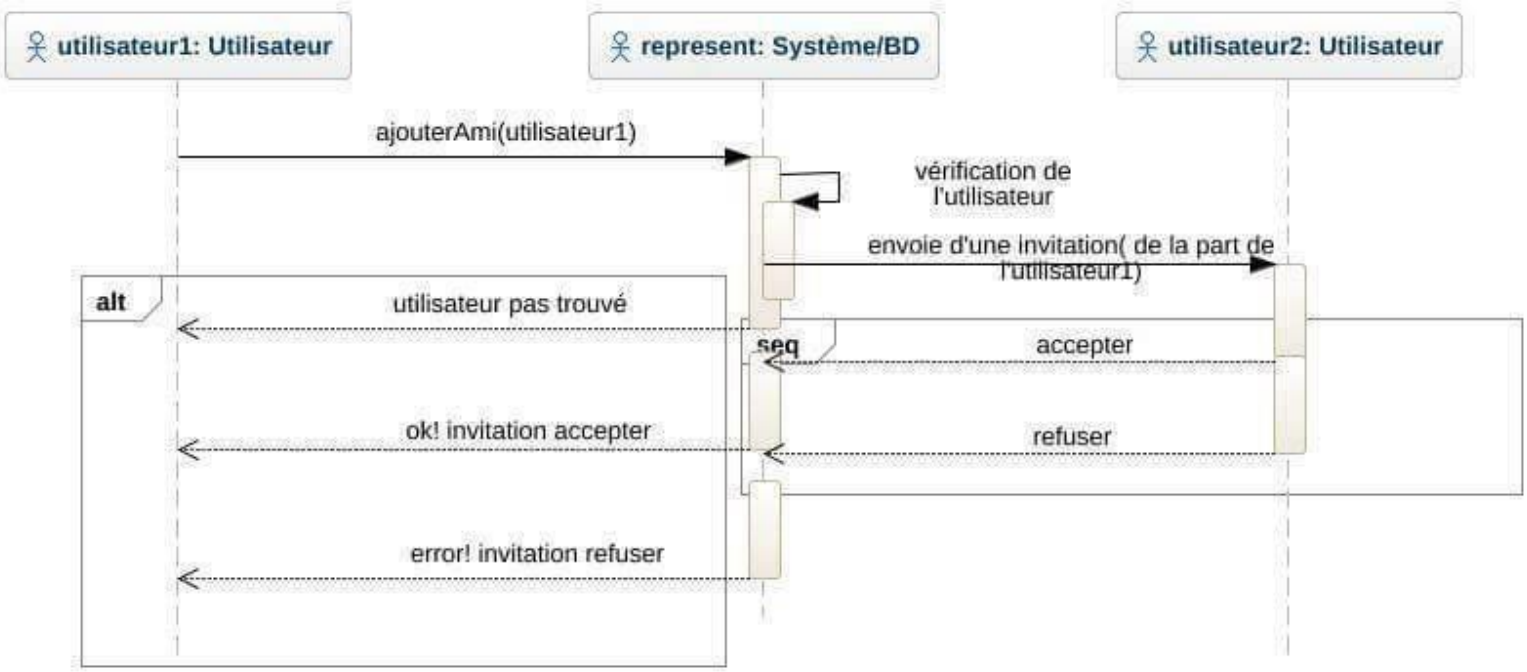
Pseudocode:

```
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;
```

III. 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.



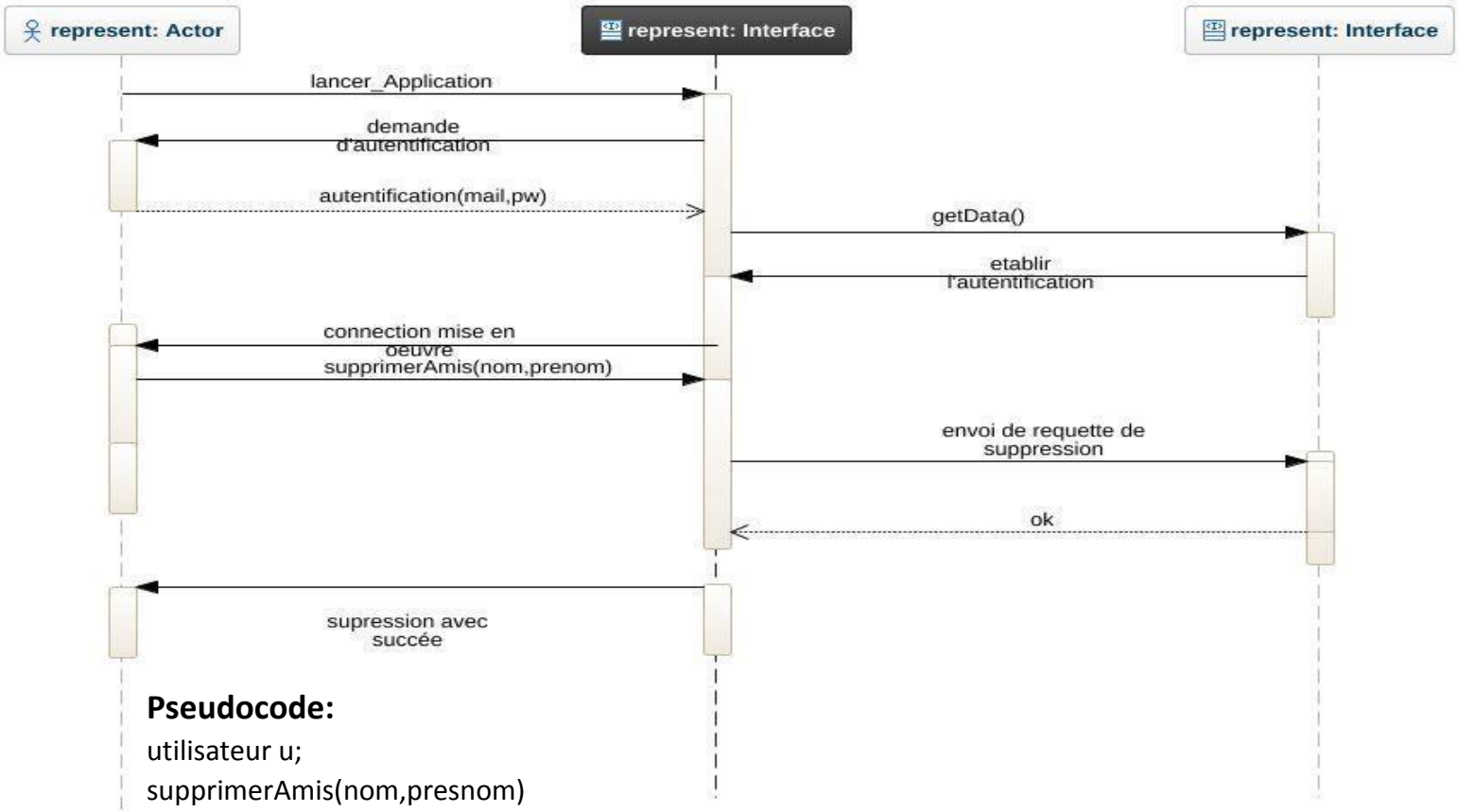
Pour le pseudocode:

```
 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 » ;
```

V. Supression d'un ami

- **Diagramme de sequence**

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 requette au système et celui-ci répond par une confirmation et supprime cet utilisateur.



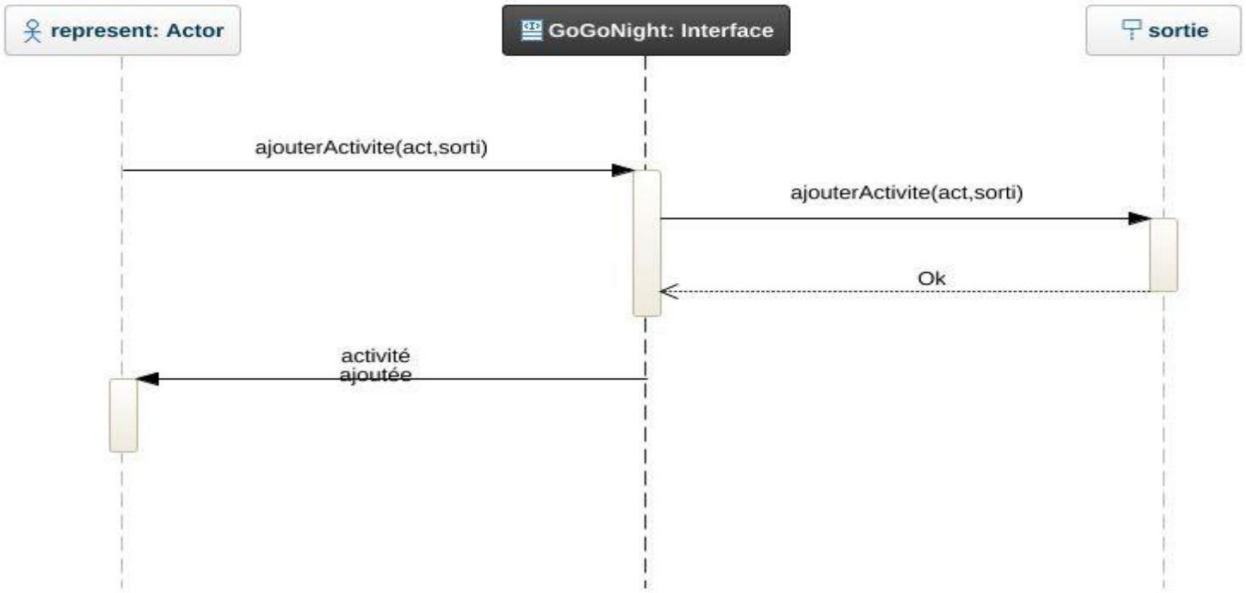
Pseudocode:

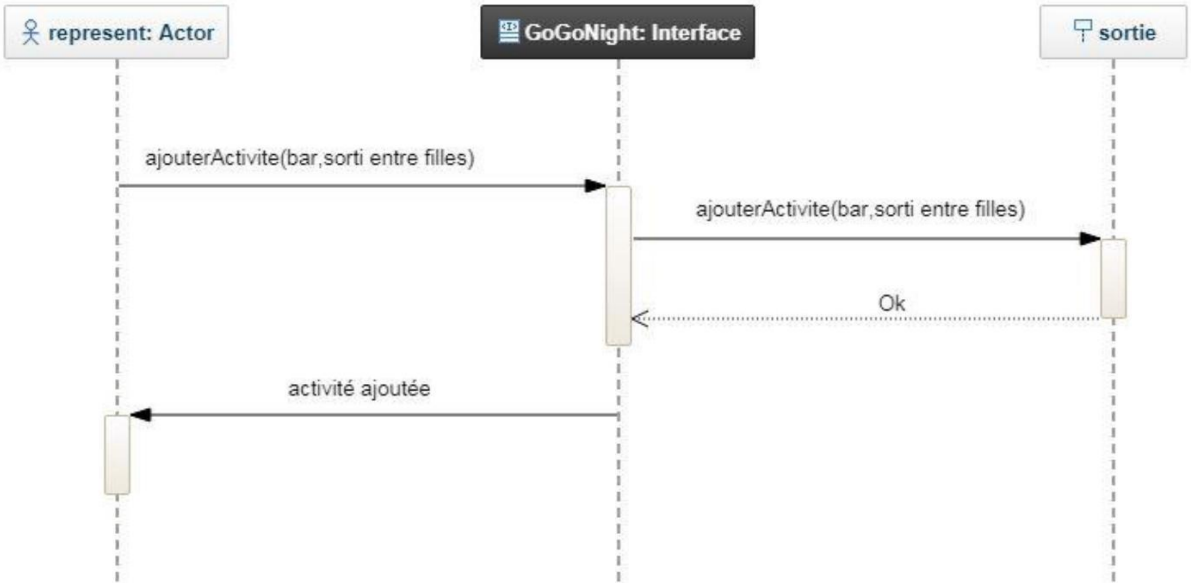
utilisateur u;
supprimerAmis(nom,presnom)
String paramEmail = request.getParameter("nom");
String paramMotDePasse = request.getParameter("prenom");
Si
les deux paramètres correspond au même utilisateur qui est dans la liste des amis de l'utilisateur et sont correctement saisis
alors
la suppression reussi : redirection sur l'interface GogoNight;
Sinon
affichage de message error « aucun ami de ce nom sur votre liste d'amis »;

VII. 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.





Pseudocode:

utilisateur u;

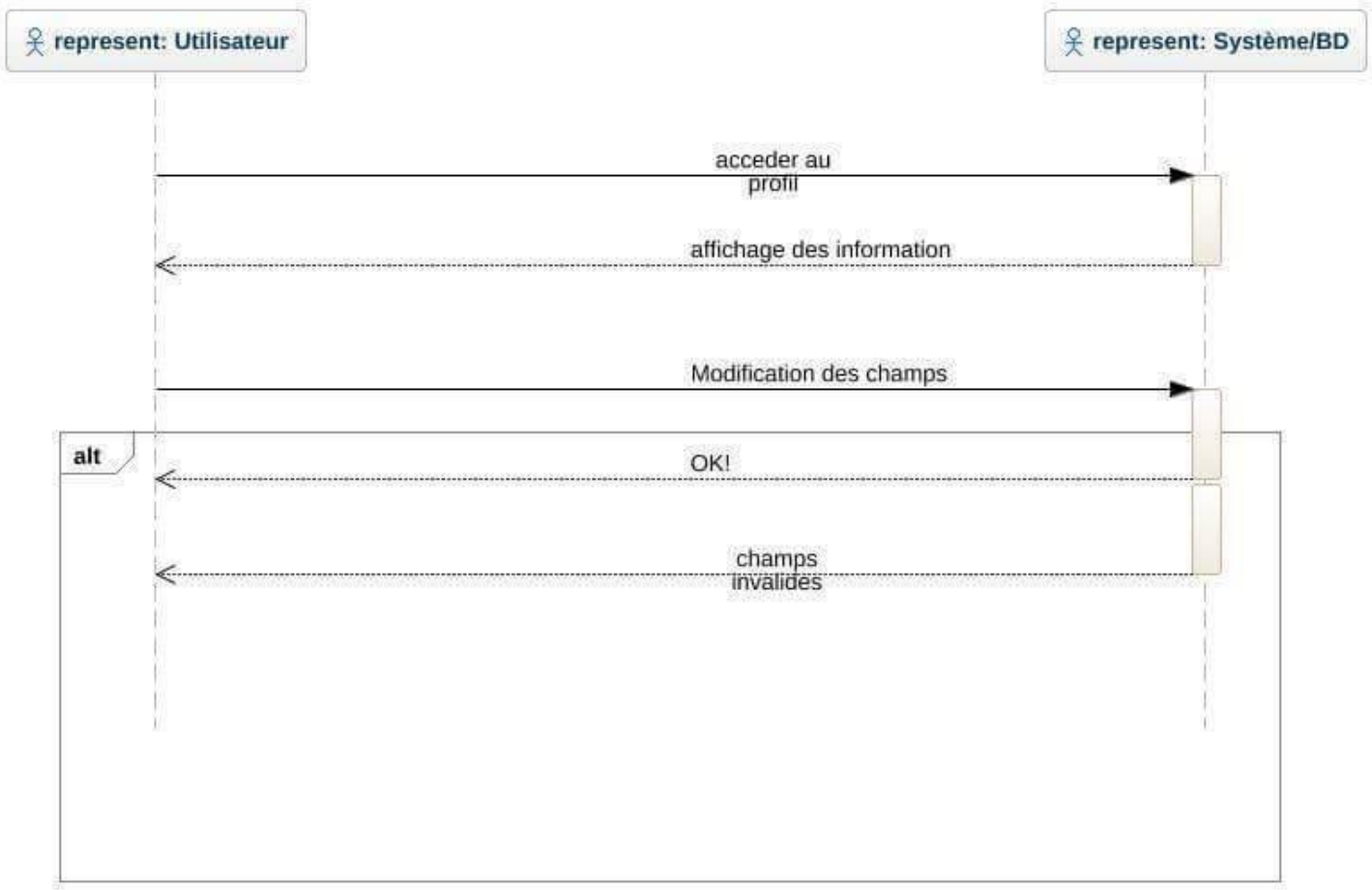
ajouteActivite (act,sortie) ;

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

VIII. Consultation et modifiaction 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 sequence**



Pseudocode :

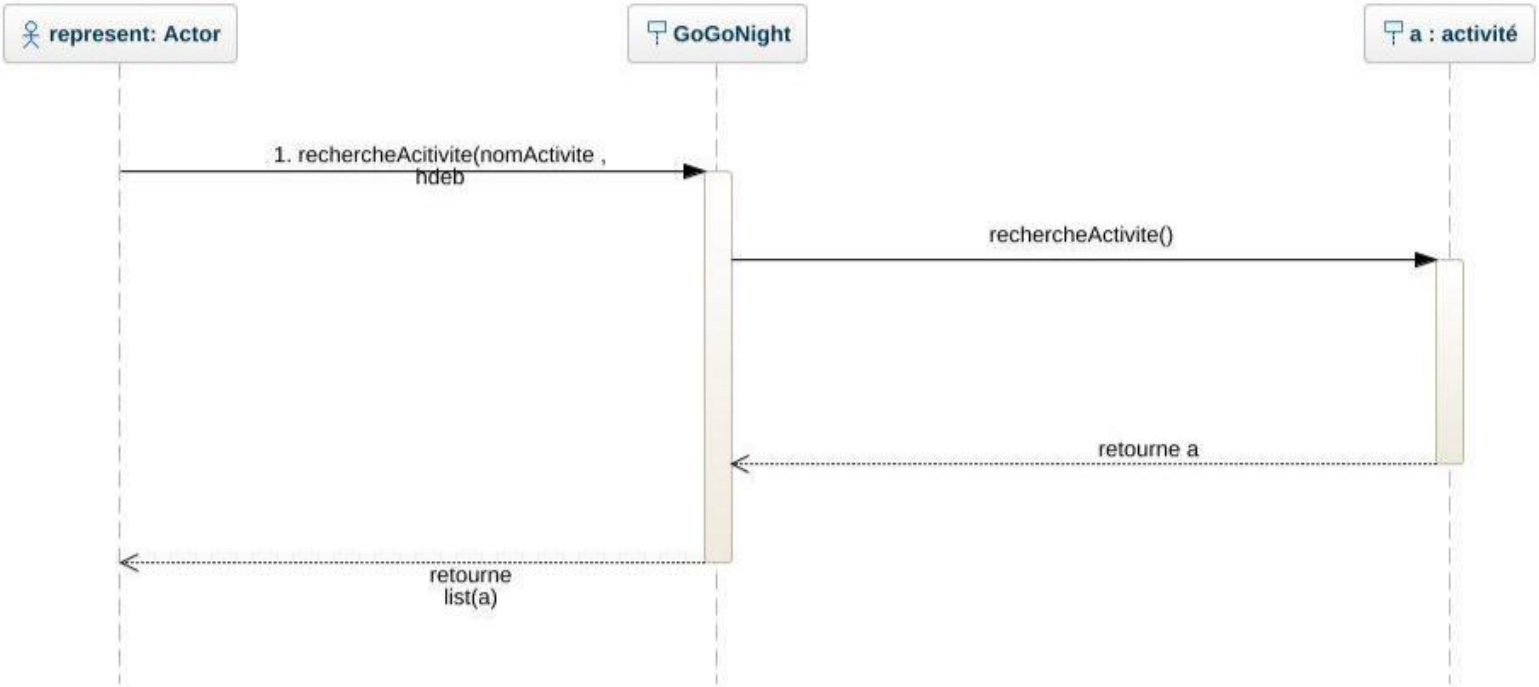
```
Utilisateur u ;
consulterProfil(u)
 u.nom = u.getNom()
u.prenom =u.getPrenom()
u.age=u,getAge()
u.adresse =u.getAdresse()
u.age=u.getAge()
u.telepone =u,getTelephone()
u.email=u.getEmail()
u.motDePasse=u,getMotDePasse()
u.preferencesAlimentaires =u.getPreferencesAlimentaires()
u.moyenDeTransport=u.getMoyenDeTraansport()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

VI. Recherche activite

- **Diagramme de sequence**

La recherche est faite à l'aide d'un mot clé, representé ici par nomActivite, et eventuellement l'heure de début d'activité demandé (hDeb), afin de pouvoir retourner la liste des lieux ouverts. les resultats seront affichés selon la destance c'est-à-dire de plus proche au moins proche de la derniere 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.



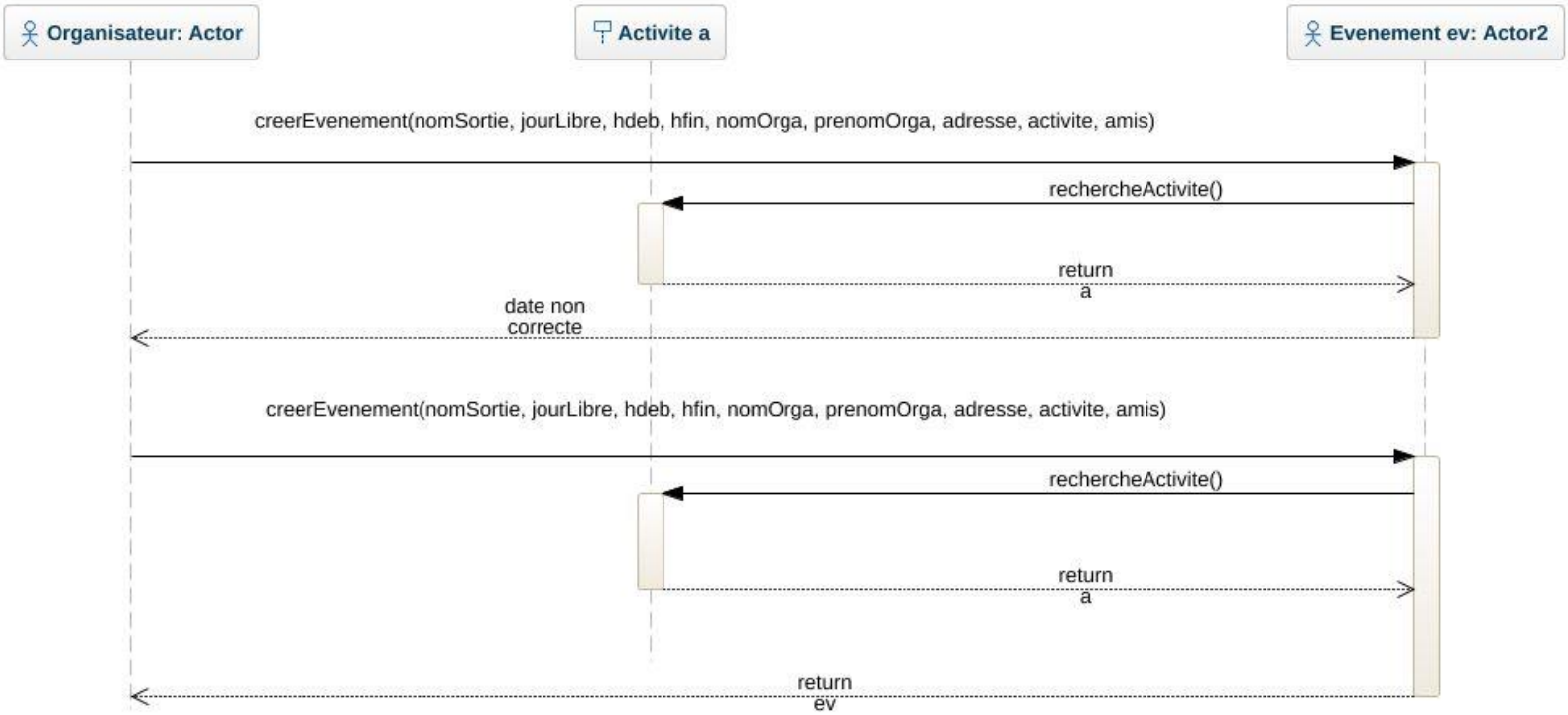
Pseudo code

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 »

IV. Creer evenement

- **Diagramme de sequence**

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.



Pseudocode:

```
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;
```

Presentation 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 outile nous permettant construir 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**

- **Google Places API Web Service**

- **MySQL**