Génie Logiciel Avancé

L'application en java pour faire la java

Second Livrable - conception groupe C

Lucas Salvato, Jeremy Quyen, Duc Nguyen

Sommaire

Vocabulaire

Pour des raisons de clarté nous allons expliquer les termes spécifiques au projet :

Utilisateur : Personne inscrite sur le site possédant un nom, une adresse et éventuellement un liste de préférences alimentaires
Ambiancé : Utilisateur participant à un événement
Organisateur : Utilisateur créant un événement
Événement : Liste d'activités avec une heure de début et une heure de fin, un organisateur et des ambiancés.
Activité : composantes d'un événement (cinéma, restaurant, bar etc.)

État de l'art

N'ayant pu couvrir cette partie dans l'analyse nous profitons de ce nouveau dossier pour le faire.

Il n'existe pas réellement d'équivalent de notre application sur le marché. Des outils similaire sont cependant utilisé.

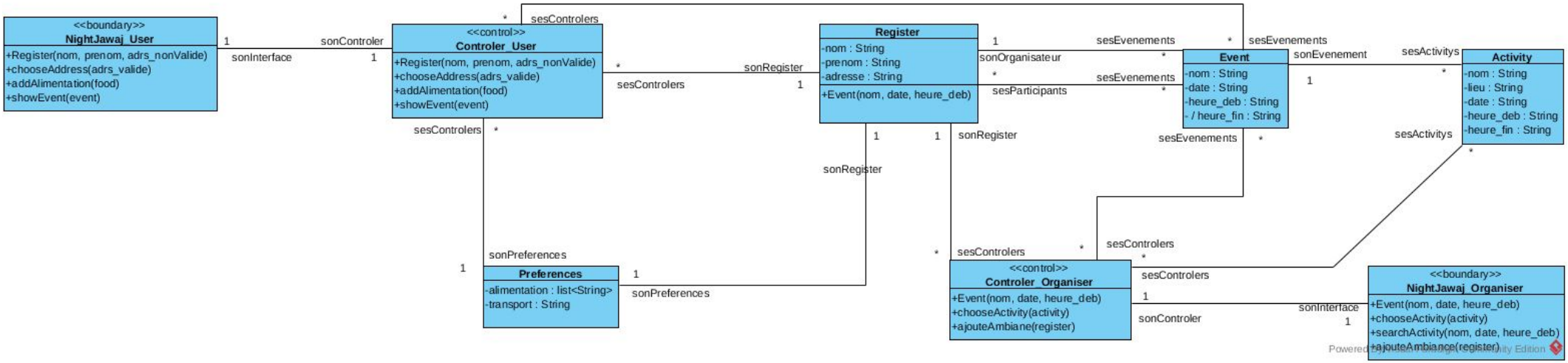
Le plus largement répandu est sans contexte Facebook et son système d'événement. Il permet à n'importe quel utilisateur du réseau social de créer un événement en spécifiant une date et un lieu puis de le partager à ses amis.
Avantages : Simple, rapide et pratique car tout le monde est déjà inscrit sur facebook pas besoin de se réinscrire et apprendre à utiliser un nouveau service. Multi plateforme.
Inconvénients : Par rapport à notre service le créateur d'événement Facebook ne permet pas de générer automatiquement une liste de différents événements en se basant sur la position géographique des utilisateurs.
Conclusion : Le créateur d'événement Facebook est sensiblement différent de notre outil dans l'approche. Facebook oriente son système sur la large diffusion d'un événement global. Il ne gère qu'un seul lieu et la liste des invités et souvent faites pour être très grande.
Notre outil quand à lui se rapproche plus d'un site de rencontre dans le sens où le nombre de personne invités aux événements est plus réduit. De plus le système d'automatisation à l'aide des positions géographiques de chaque participants est un vrai plus non négligeable.

Un autre service similaire mais beaucoup moins utilisé est Tinder Social. C'est une application mobile basé sur le principe de Tinder sauf qu'il permet de se faire des amis. On se créer un réseau de personnes que l'on connaît puis l'application suggère des personnes se trouvant à proximité géographique de chez nous. L'idée propose ensuite de chatter avec ses amis pour organiser des soirées
Avantages : Système de *match* basé sur Tinder qui permet de rencontrer des nouvelles personnes avec la position géographique.
Inconvénients : Ne permet pas d'organiser automatiquement la soirée. N'existe pas sur ordinateur.
Conclusion : Encore une fois cette application n'a pas tout à fait la même orientation que la nôtre. Cette application se concentre sur le fait de rencontrer de nouvelles personnes tanti que la nôtre sur l'organisation d'événement entre ami. L'utilisation de notre application pourrait être complémentaire à celle de Tinder Social. L'une pour se faire des amis, l'autre pour organiser des événements avec eux.

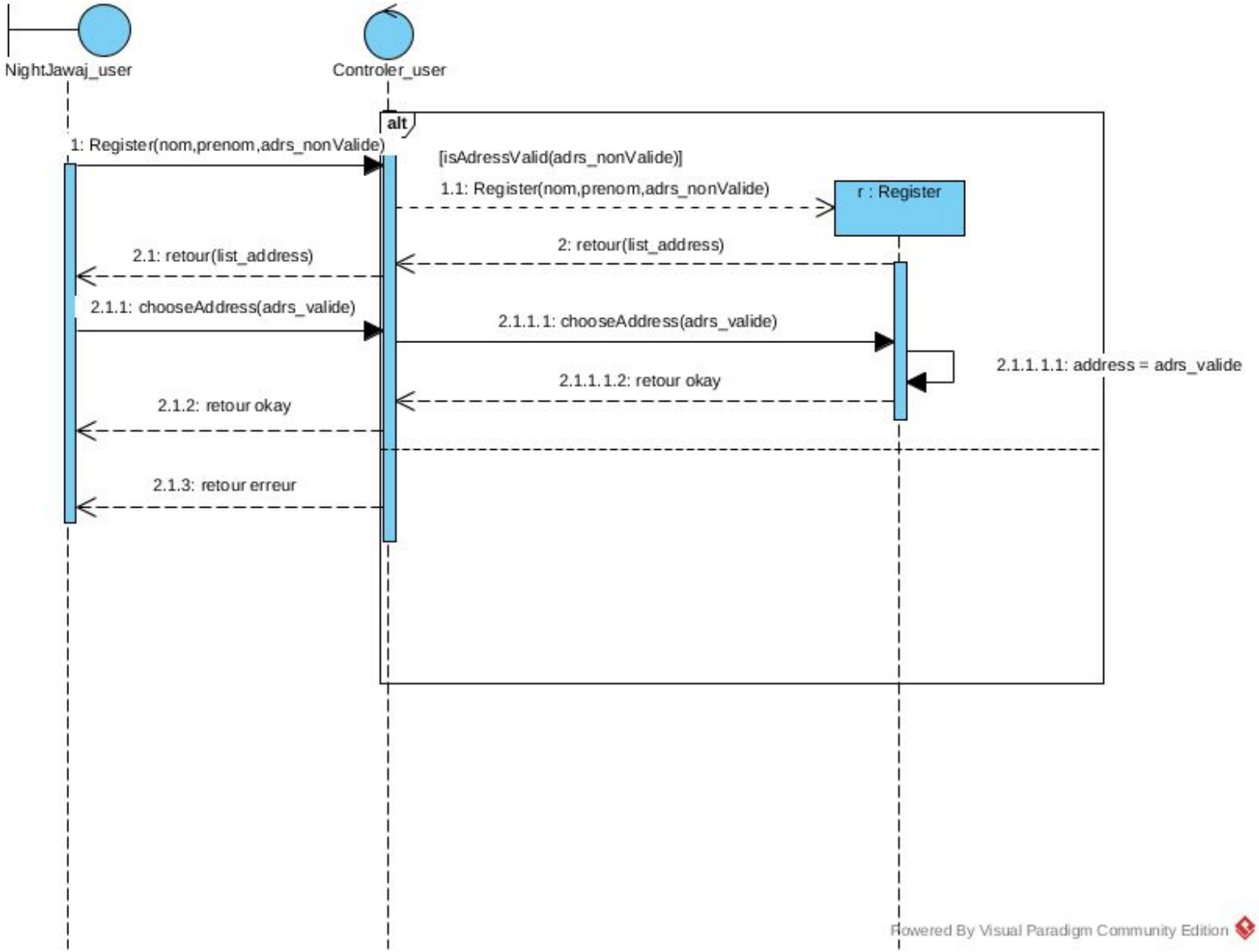
Diagrammes

Notre programme suit un système de modèle, vue et contrôleur (MVC).

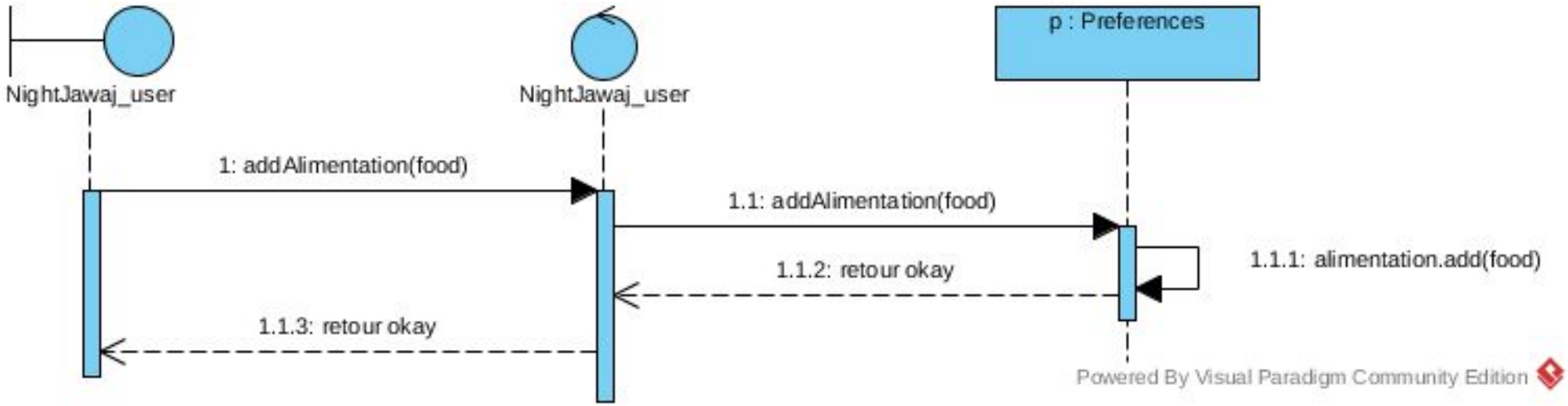
Diagramme de classe



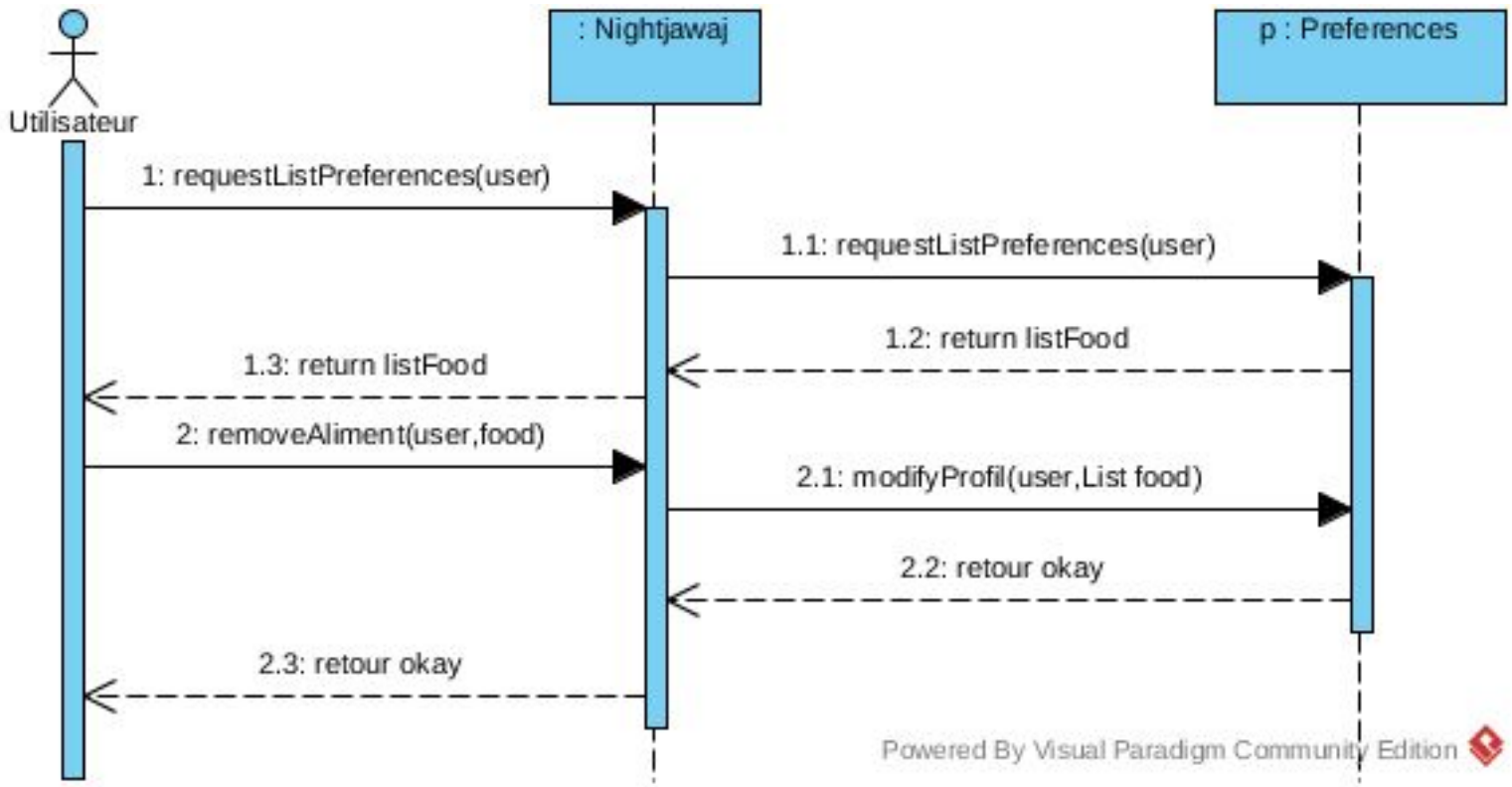
Inscription



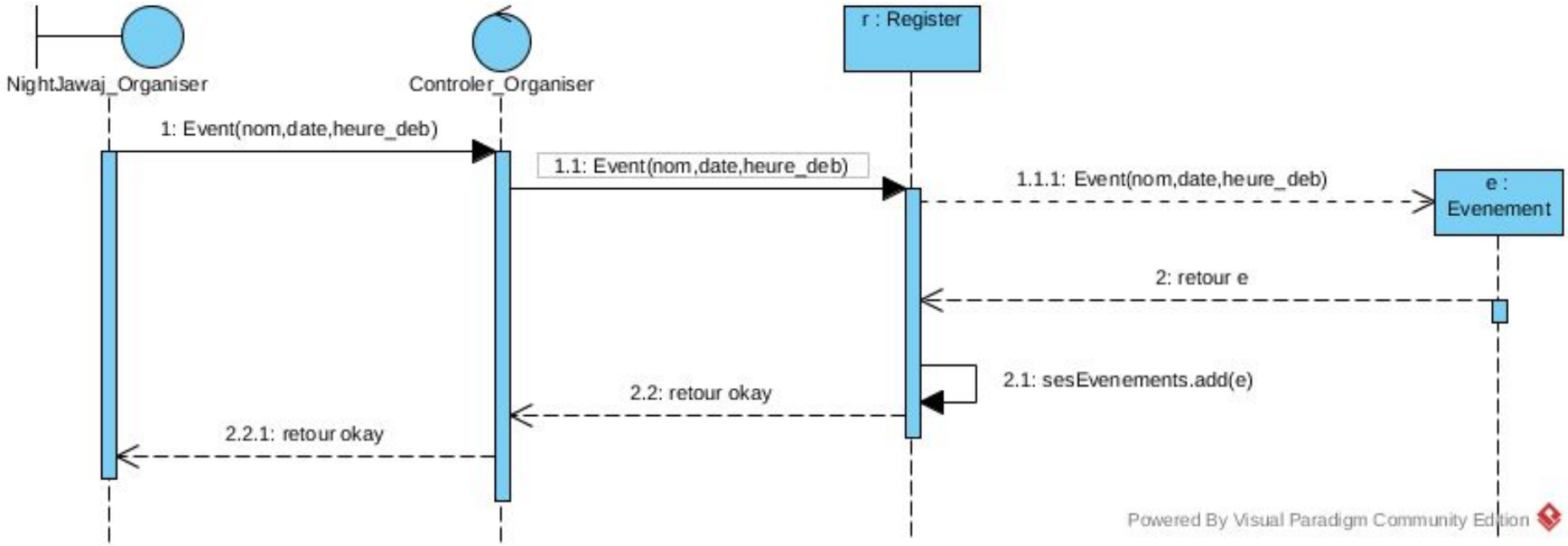
Ajouter aliment



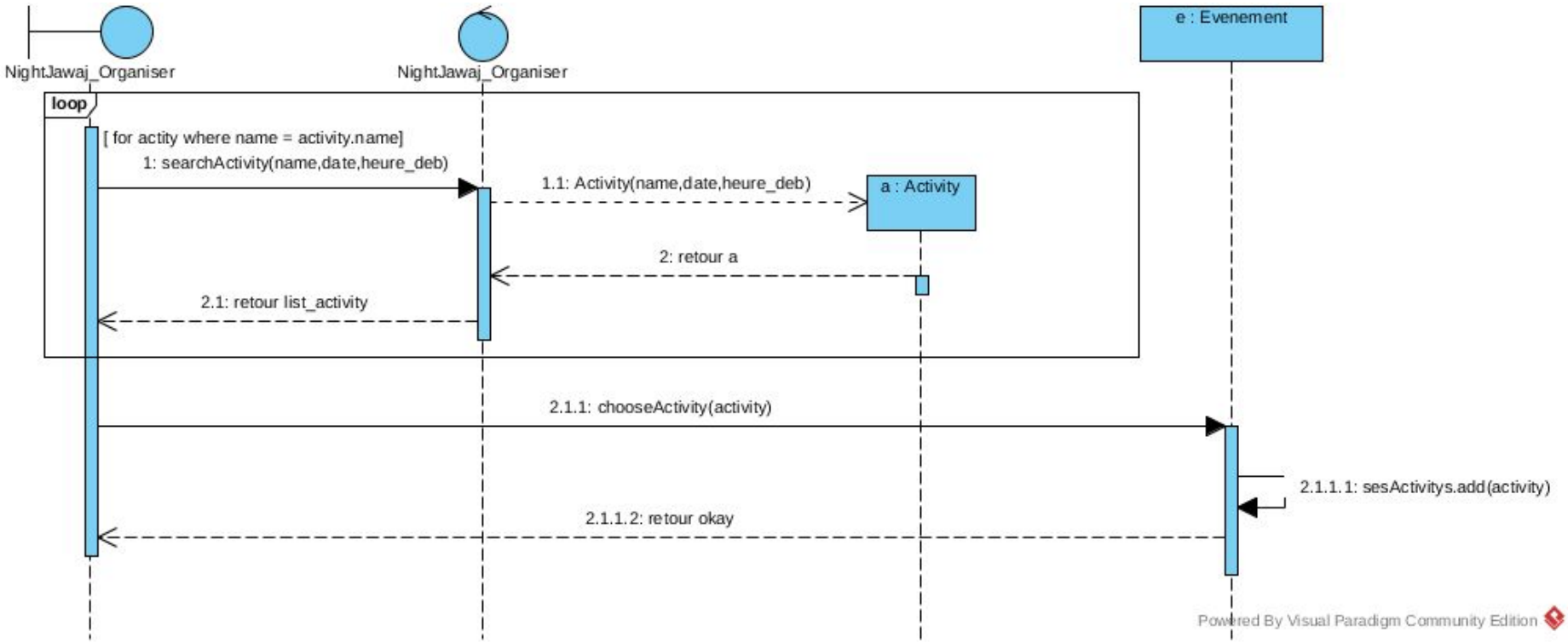
Retirer Aliment



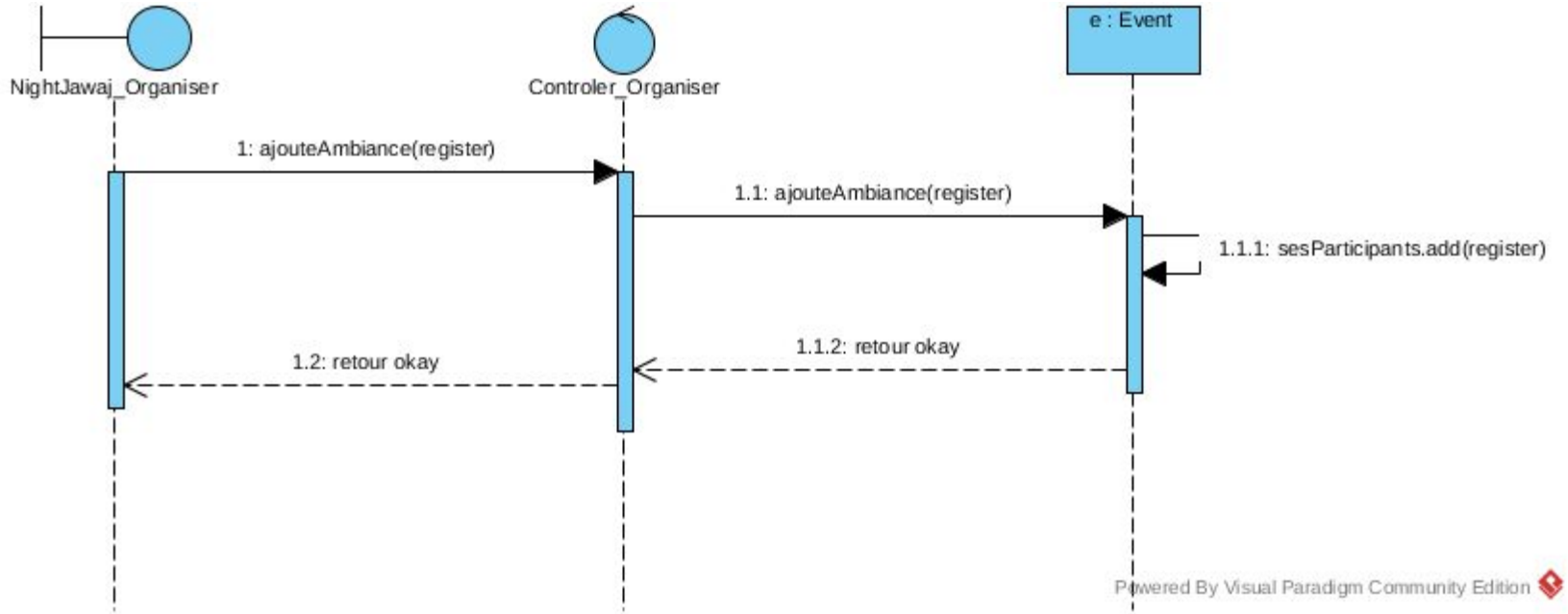
Créer évènement



Ajouter activité



Ajouter Ambiancé



Afficher événement

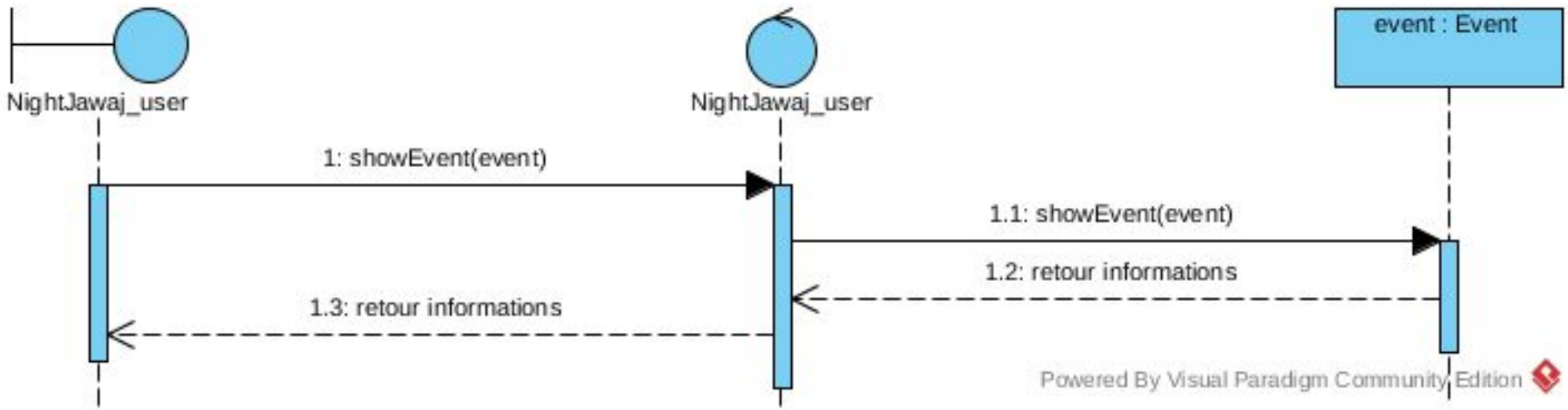
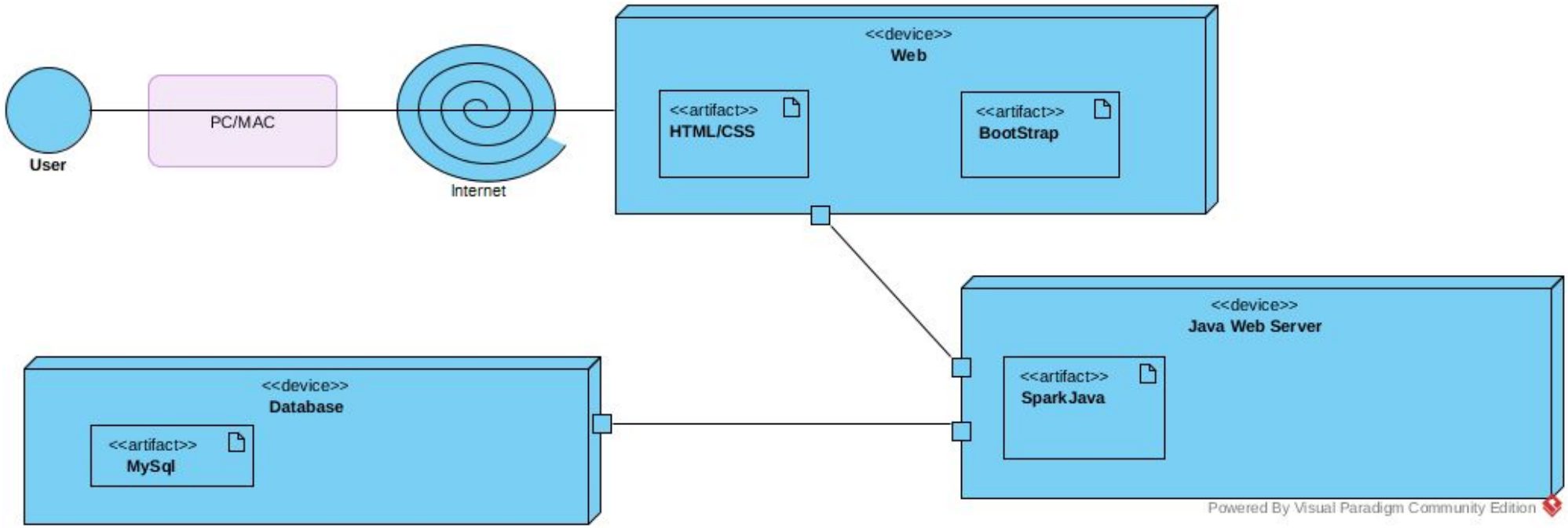


Diagramme de déploiement



UNIVERSITÉ PARIS SUD

Pseudo code

Inscription

```
// register informations
last_name = request.queryParams( queryParam: "last_name");
first_name = request.queryParams( queryParam: "first_name");
address = request.queryParams( queryParam: "address");
System.out.println(last_name+first_name+address);

//Verify address
if (isAddressValid(address)){
    ArrayList<String> list_address = registerAddress(address);
    // Create session from now on
    request.session( create: true);
    request.session().attribute( name: "lname",last_name);
    request.session().attribute( name: "fname",first_name);

    chooseAddress(list_address);
    response.redirect( location: "/choose_address");
    return "<p>It does exist!</p>";
}
else return "<p>This address doesn't exist</p>";
// Return to previous page
```

UNIVERSITÉ PARIS SUD

Valider adresse

```java
public boolean isAddressValid(String address) throws IOException, JSONException {
    String addressEncoded = URLEncoder.encode(address, enc: "UTF-8"); // Make the address URL compliant
    URL url = new URL( spec: "https://maps.googleapis.com/maps/api/place/textsearch/json?query="+
            addressEncoded+"&key=" + GooglePlacesKey); // Search for the address via API
    Scanner scan = new Scanner(url.openStream());
    String html_output = new String();
    while (scan.hasNext())
        html_output += scan.nextLine();
    scan.close();
    JSONObject j = new JSONObject(html_output);

    System.out.println(j.getString( key: "status"));
    return (j.getString( key: "status").equals("OK")); // if status is OK then address is valid
}

public ArrayList<String> registerAddress(String address) throws IOException, JSONException {
    String addressEncoded = URLEncoder.encode(address,  enc: "UTF-8"); // Make the address URL compliant
    URL url = new URL("https://maps.googleapis.com/maps/api/place/textsearch/json?query=" +
            screaddressEncoded + "&key=" + GooglePlacesKey); // Search for the address via API
    Scanner scan = new Scanner(url.openStream());
    String html_output = new String();
    while (scan.hasNext())
        html_output += scan.nextLine();
    scan.close();
    JSONObject j = new JSONObject(html_output);

    ArrayList<String> results = new ArrayList<>();
    System.out.println("nb address : "+j.getJSONArray( key: "results").length());
    for (int i = 0; i < j.getJSONArray( key: "results").length(); i++) { // Getting results in an Array
        JSONObject addr = j.getJSONArray( key: "results").getJSONObject(i);
        results.add(addr.getString( key: "formatted_address"));
        System.out.println(results.get(i));
    }

    return results;
}
```

Interface web

Captures d’écrans de l’interface utilisateur

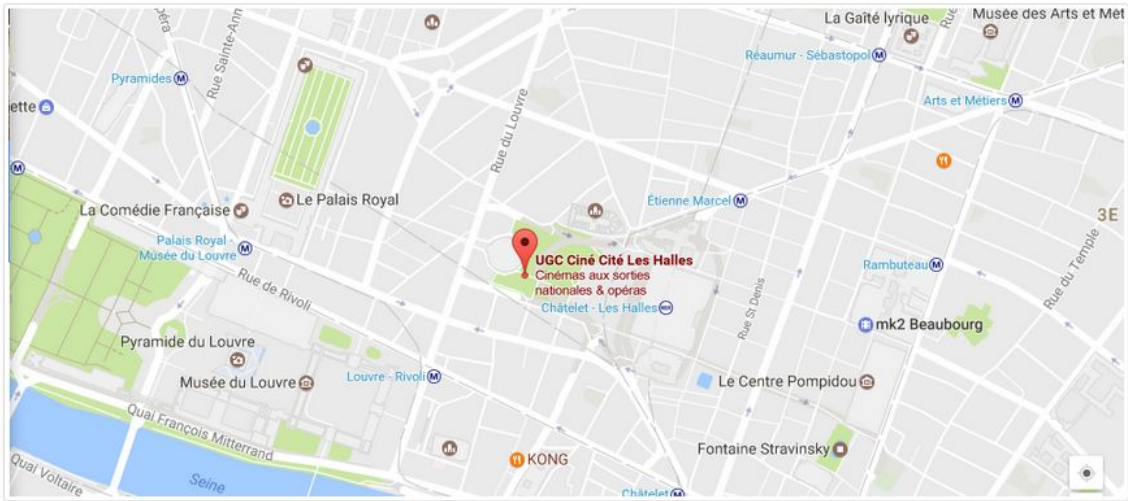
UNIVERSITÉ PARIS SUD

Recherche activité

nom activite

HH:MM:SS

search



restaurant
pizza hut
HD Diner

Ok! L'activité a bien été ajouté.

restaurant	cinéma
pizza hut	cinéma ugc chatelet
HD Diner	cinéma ugc bercy