

Génie Logiciel Avancé

Nightjawaj

Premier Livrable - analyse groupe 7

Lucas Salvato, Jeremy Quyen, Duc Nguyen

Introduction	4
Cahier des charges	5
Présentation des composantes techniques	7
Web	7
Java	7
API Google places	8
Spark	9
JDBC et MySQL	9
Diagramme de Cas d'Utilisation	10
Diagramme de Classe	11
Diagrammes de Séquences	12
Créer événement	12
Ajouter Ambiancé	13
Rechercher Activité	14
Ajouter Activité	15
Afficher Planning	16
Maquettes de l'application	17
Idée design	17
Maquette "Créer un événement"	18
Maquette recherche Activité	19
Maquette ajouté activite	20
Maquette Ajout Ambiancé	21
Maquette Afficher planning	22
Conclusion	23
Bibliographie	24

Introduction

Nightjawaj est un assistant de sortie entre amis, permettant à plusieurs personnes de participer à une suite d'activités, orchestré automatiquement par le logiciel à l'aide de critères défini par l'organisateur. Le logiciel prend en compte la position géographique de chaque participants pour définir les lieux qui composeront la sortie.

Il s'agit d'une application web Java s'appuyant sur la librairie Spark et l'API google places, relié à une base de données géré par la librairie JDBC (mysql Connector/J).

Nightjawaj un logiciel pouvant être utilisé par n'importe qui et sur n'importe quel plateforme. La partie client est en web ; rapide, simple d'utilisation et multiplateforme. Elle fournit aux utilisateurs une grande quantité d'informations en un temps très court. Le client se veut avant tout fonctionnel. La partie serveur est Java ; rapide, légère, sécurisé et multiplateforme, installable facilement et demandant peu ou pas de configuration.

Cahier des charges

- Multiplateforme : application web responsive design
- Design simple
- Facile d'utilisation
- Rapide
- Profil utilisateur :
 - Nom, prénom
 - Âge
 - Adresse
 - Préférences alimentaires (liste de tags)
 - Amis
 - Moyen transport (distance possible)
- L'utilisateur définit les horaires de la sortie et les activités qu'il veut faire durant celle-ci, ainsi que l'ordre dans lequel il veut les faire. Une fois toutes les activités définies par l'utilisateur, l'application renvoie une liste de lieux avec les horaires d'arrivées sur chacun des lieux.
- Si certaines activités ne sont pas possibles, l'application le notifie à l'utilisateur et propose éventuellement des alternatives.
- Les activités disponibles sont : (avec nom présenté dans l'API)
 - Restaurant (*restaurant*)
 - Bar (*bar*)
 - Café (*cafe*)
 - Boîte de nuit (*night_club*)
 - Musée (*museum*)
 - Parc (*park*)
 - Parc d'attraction (*amusement_park*)
 - Bibliothèque (*library*)
 - Casino (*casino*)

- Cinéma (*movie_theater*)
 - Centre commercial (*shopping_mall*)
 - Aquarium (*aquarium*)
 - Bowling (*bowling_alley*)
- Les horaires de chaque activités sont défini en prenant en compte les horaires d'ouverture du lieu et la distance qu'il faut pour s'y rendre depuis l'activité précédente.
 - Les recherches peuvent être appuyées par des mots-clés afin d'affiner celles-ci. Ces mots-clés seront choisi par l'utilisateur.
 - Suite à la recherche une liste de points d'intérêt seront proposé à l'utilisateur.
 - L'application affiche un lieu et une horaire de rendez-vous qui est le premier événement.
 - L'application prend en compte les habitudes alimentaires de chaque personnes pour définir le lieu. Le type d'habitude alimentaire est une liste de mots que l'utilisateur entre dans son profil utilisateur. Lors de la recherche on exclut les résultats contenant un des mots.

Présentation des composantes techniques

Web

Le support web a été choisi pour sa portabilité et sa facilité d'accès. C'est un média multiplateforme et utilisé par tous au quotidien, en un sens universel. C'est un média contemporain et d'avenir ce qui en fait un support de choix pour notre projet.



Java

Java est un langage de programmation orienté objet. C'est un langage contenant beaucoup d'outils intégrés rendant le développement simple et rapide. De plus c'est un langage sécurisé empêchant les fuites de mémoire. En utilisant ce langage nous pouvons être sûr d'avoir un logiciel stable et sécurisé.

Seul le serveur est codé en Java. Toute la partie interface utilisateur est en web, c'est à dire en HTML et CSS. Ainsi l'application est utilisable partout sur n'importe quel appareil et le serveur étant en Java, il fonctionne sur n'importe quel système d'exploitation.



API Google places

API Google places est un service créé par Google permettant de renvoyer des informations sur un lieu géographique ou encore un point d'intérêt important à l'aide de requêtes HTTP à partir de critères précis de recherche. Nous utilisons cette API afin de rechercher les différentes coordonnées des activités prévues dans la sortie à partir de la positions des différents utilisateurs. Les résultats sont affichés sous la forme des données JSON contenant les informations nécessaire. Le logiciels doit traiter ces informations et tirer celles qu'il affichera à l'utilisateur.



Spark

Spark est une librairie Java permettant de faire des applications web. Nous l'avons choisi car elle est légère et simple d'utilisation et ses fonctionnalités couvrent tout ce dont nous avons besoin pour réaliser ce projet. Nous avons choisi d'utiliser une librairie Java plutôt que du PHP ou du nodeJS car Java nous semblait plus adapté et sécurisé pour faire de grosses applications et c'est un langage que nous maîtrisons.

Le serveur sera facile d'installation lui permettant d'être lancé sur n'importe quel machine et consommera très peu de mémoire. De plus il ne nécessitera aucune dépendance supplémentaire. Spark contenant déjà un serveur web, il ne sera pas nécessaire d'en installer un supplémentaire tel que Apache ou nginx. Un simple fichier « jar » permettra de lancer le serveur sur toute machine possédant Java sans faire de configuration supplémentaire.

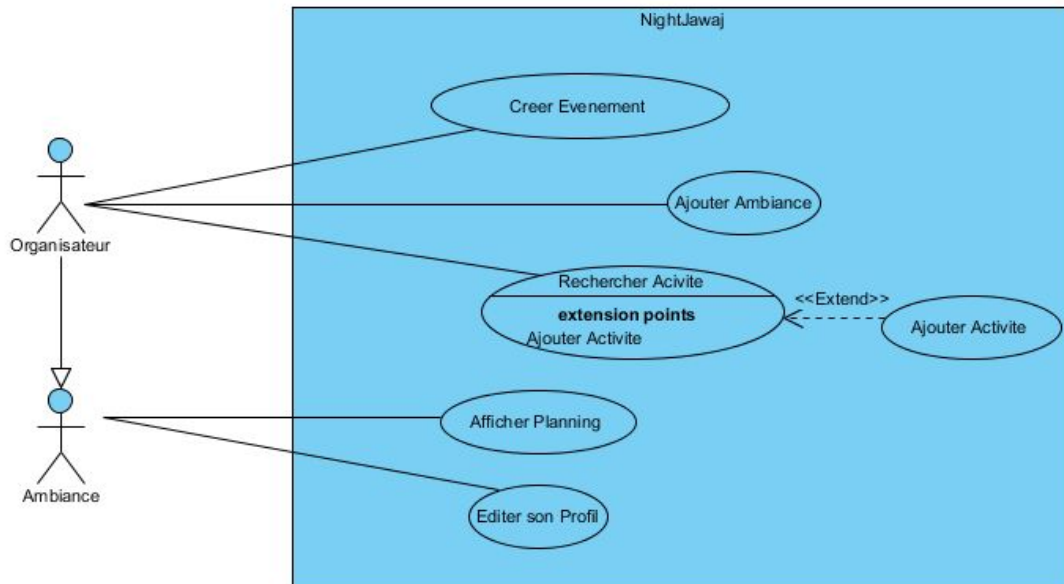


JDBC et MySQL

JDBC est la librairie nous permettant d'utiliser des bases de données dans Java. Nous avons décidé d'utiliser une base de donnée MySQL car c'est un logiciel libre et gratuit que nous avons déjà utilisé par le passé. Il est simple d'utilisation, répandu et sécurisé.



Diagramme de Cas d'Utilisation



NightJawaj comporte deux types d'utilisateurs : les ambiancés et les organisateurs qui sont une extension des ambiancés (qui peuvent donc effectuer les actions de ceux-ci).

Créer Evenement : Comme son nom l'indique, va permettre aux organisateurs de créer leurs événements

Ajouter Ambiancé : Pour ajouter des ambiancés aux événements.

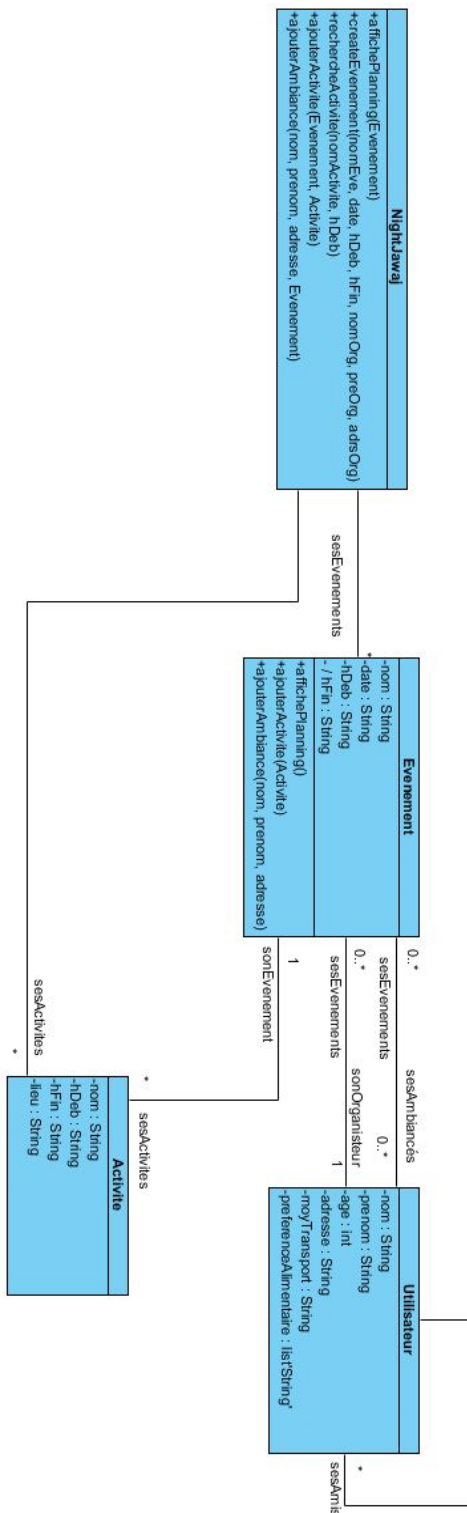
Rechercher Activité : Va permettre aux Organisateur de rechercher des activités à partir de mot-clés.

Ajouter Activité : Ne pourra être exécuté seulement après avoir fait la recherche.

Afficher Planning : Faisable par tous les Utilisateurs ayant au moins un événement, permet d'afficher l'ensemble des informations d'un événement.

Editer Profil : Permet aux Utilisateurs de changer leurs informations mais aussi de rajouter certaines informations non-obligatoires : moyens de transport, préférences alimentaires, amis, etc...

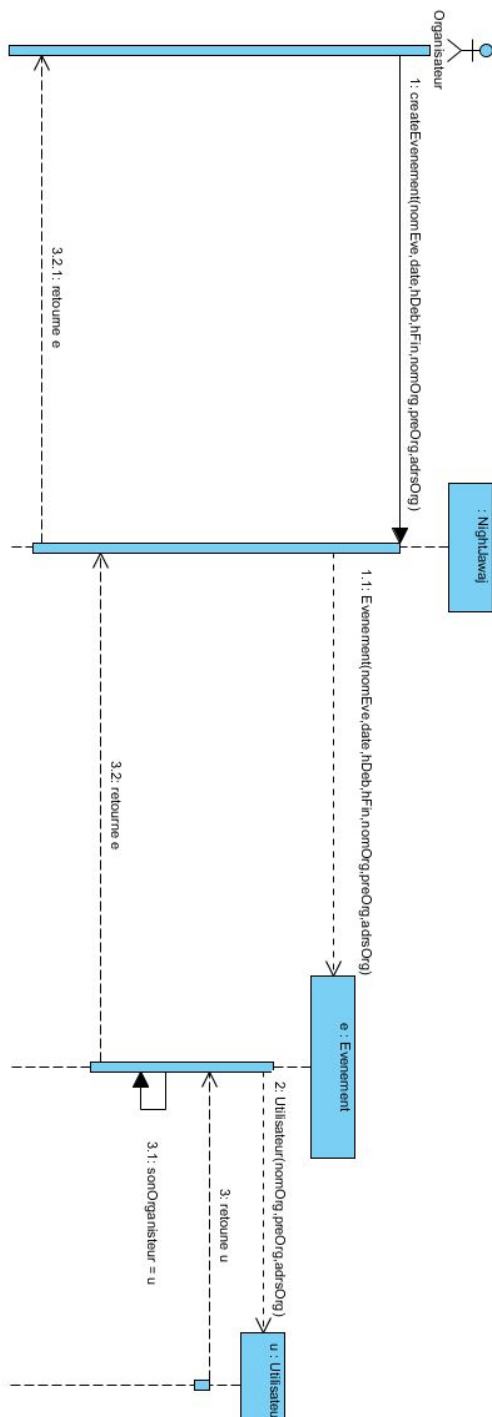
Diagramme de Classe



Précision : Le hFin d'événement est un calculable correspondant au hFin de la dernière activité de l'événement. Les constructeurs, setters et getters ne sont pas affichés.

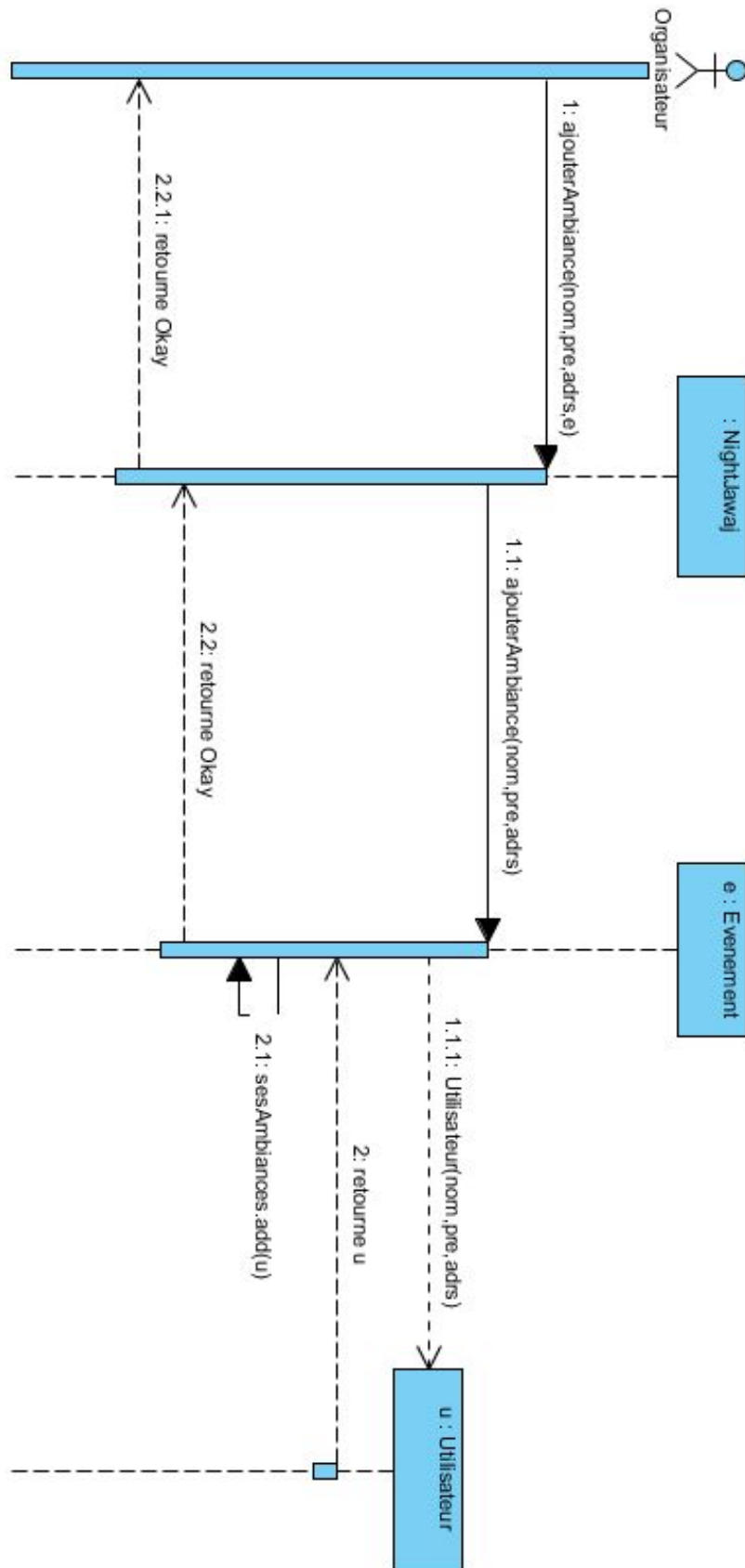
Diagrammes de Séquences

Créer événement

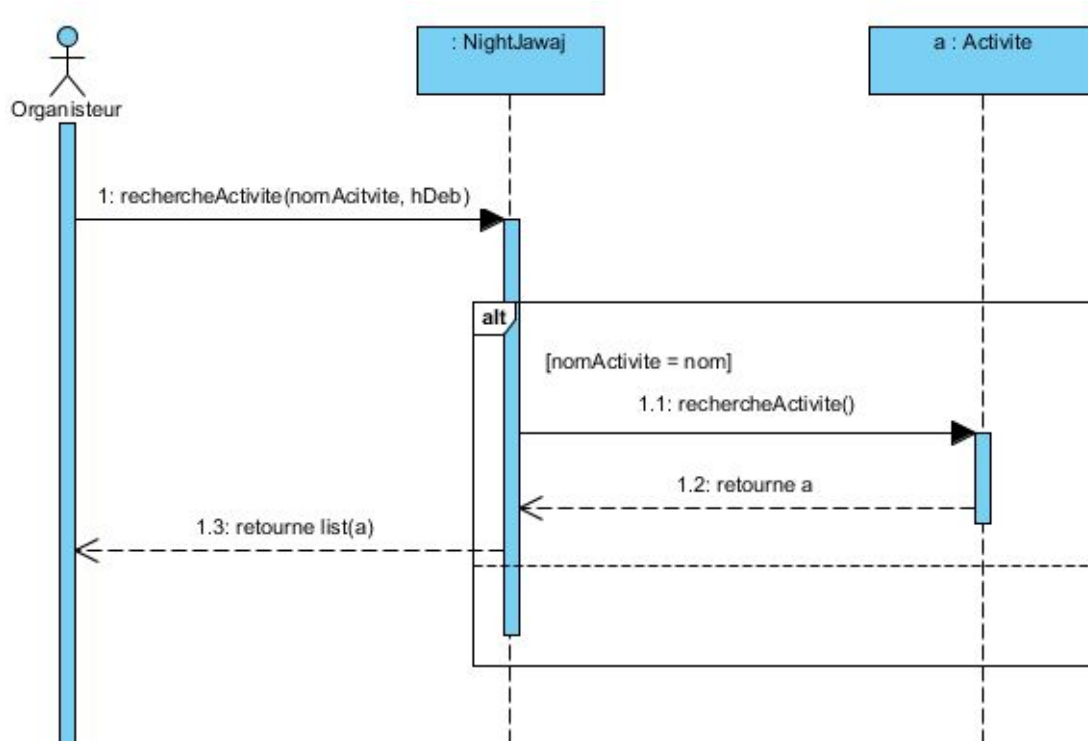


Lorsqu'un Utilisateur crée un événement, il se crée aussi une identité d'organisateur. Liant ainsi l'événement à un organisateur qui lui est propre.

Ajouter Ambiance

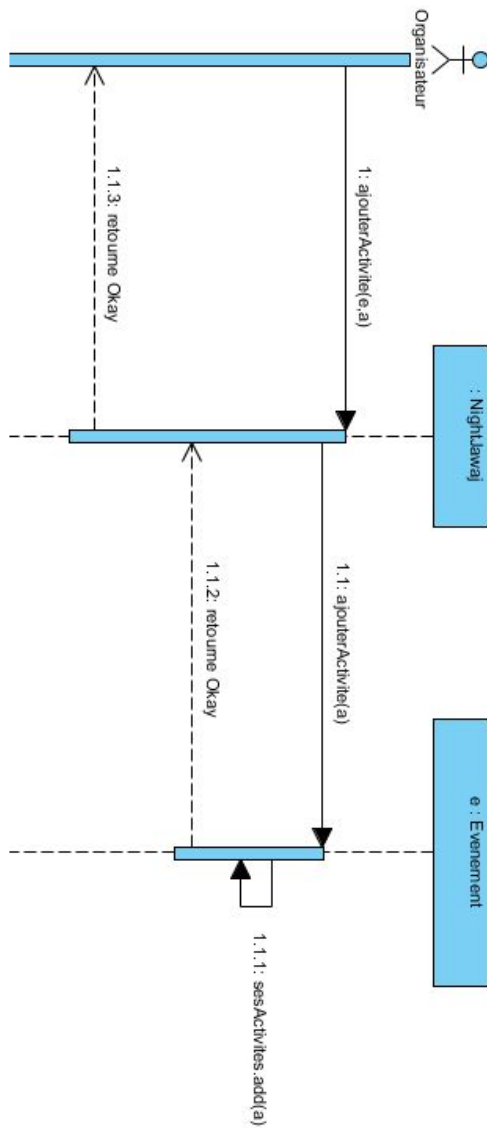


Rechercher Activité

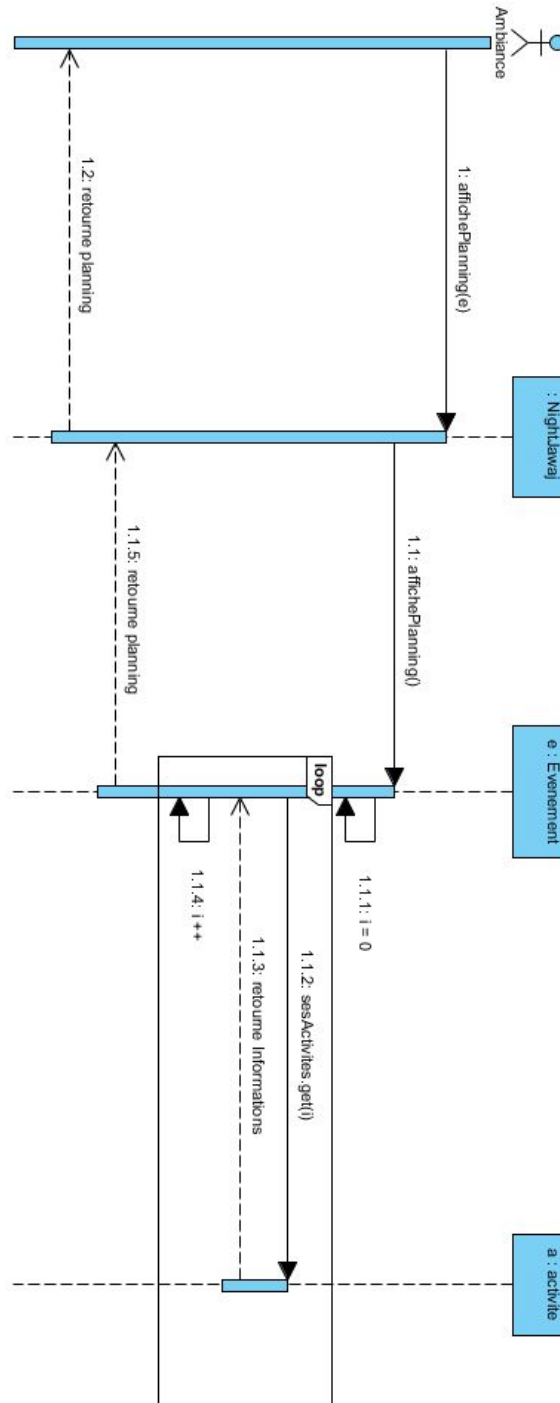


La recherche est faite à partir de mots-clés représentés ici par `nomActivite` mais le résultat dépend aussi de l'heure de début demandée afin de donner les lieux ouverts. Les résultats seront affichés dans l'ordre de plus au moins proche de la dernière activité avant le `hDeb` demandé. Si aucune activité n'est trouvée, alors un message s'affichera et invitera l'utilisateur à modifier sa requête.

Ajouter Activité



Afficher Planning



Cette fonctionnalité affiche aussi la totalité des informations concernant l'événement : Les participants, la date, le nom de l'événements, etc...

Maquettes de l'application

Idée design

Le design de l'application se veut épuré et le plus simple possible pour être compréhensible instantanément par n'importe qui. Notre priorité est la clarté et la simplicité. L'application contiendra donc peu de couleurs ou tout autre éléments pouvant distraire l'utilisateur. Celui-ci devra passer le moins de temps possible sur l'application tout en récupérant le maximum d'information possible durant sa visite. Le but étant que celle-ci soit utilisée régulièrement mais durant de très courte période. C'est de cette idée directrice que découle donc le design « épuré » de l'interface. Cependant cette simplicité n'est qu'apparente car l'interface contient toute les informations nécessaire à l'utilisateur lorsqu'il veut consulter une sortie et celle-ci sont repérable d'un seul coup d'œil. Ainsi l'utilisateur ne perd pas de temps à chercher ce dont il a besoin. Tout lui est présenté instantanément de façon clair.

Maquette “Créer un événement”

Nom événement : nom orga: prénom orga :

Heure début : heure fin : adresse orga :

↓

Nom evenement	nom orga	prenom orga	heure début	heure fin	adresse orga
Nightjawaj	Allen	Barry	19h	23H30	11 rue Saint-denis 75001 Paris

Lorsqu'un utilisateur créer un événement, il se crée aussi une identité d'organisateur. Liant ainsi l'événement à un organisateur qui lui est propre. Il ajoute tous les champs puis appui sur le bouton créer et un tableau avec les coordonnées de l'utilisateur et celles de l'événement apparaissent.

Maquette recherche Activité

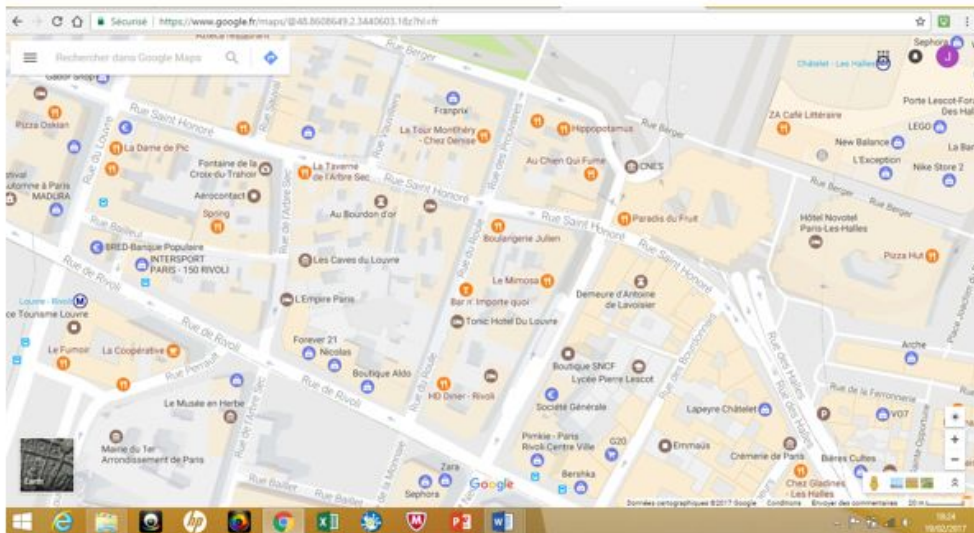
Activités :

Restaurant

heure :

19H

search



RESTAURANT

Pizza hut

HD DINER

L'utilisateur ajoute une heure et activité puis appui sur search. La carte s'affiche avec un tableau et une liste d'activités correspondant à sa recherche.

Maquette ajout  activite

Nom de l'activit  :

cin ma

Ajout 

L'activit  a bien  t  ajout e   votre liste

<u>restaurant</u>	Cin�ma
Pizza hut	UGC
HD DINNER	Chatelet

L'utilisateur rentre le nom de l'activit  et appui sur le bouton "ajout ". Le message « l'activit  a bien  t  ajout e   votre liste » est affich  et l'activit  qui a  t  ajout  s'affiche dans la liste des activit s correspondantes.

Maquette Ajout Ambiancé

Nom :
 Prénom :
 Age :
 Adresse :

3 rue Léon
Frot 75011
Paris

Préférence alimentaire :
 moyen de transport :

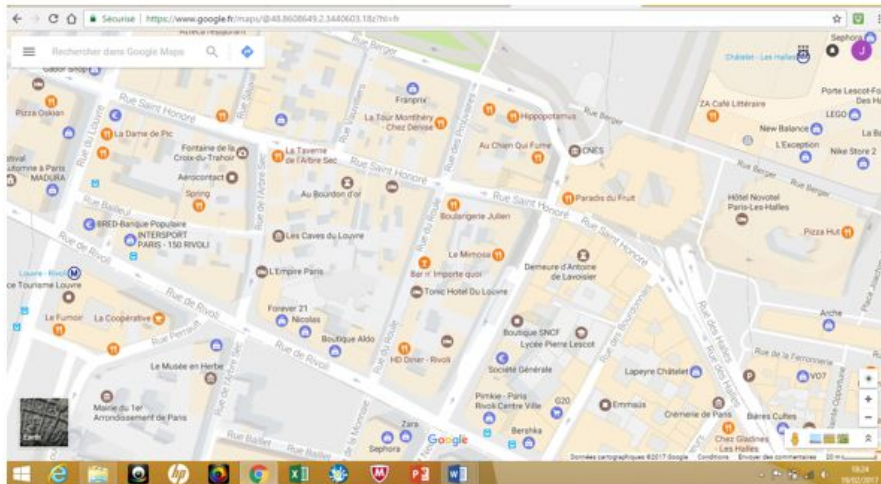
↓

Nom	Prénom	Age	Adresse	Préférence alimentaire	moyen de transport
Wayne	Bruce	25	3 rue Léon Frot 75011 Paris	Pizza	métro

Lorsque l'utilisateur remplit tous les champs, il appuie sur le bouton « ajouter » et on affiche dans un tableau les informations du participant correspondant.

Maquette Afficher planning

Heure de début :	<input type="text" value="19H"/>	Heure de fin :	<input type="text" value="23H30"/>	<input type="button" value="afficher"/>
------------------	----------------------------------	----------------	------------------------------------	---



19H : diner à pizza hut

21h : boire un verre au bar

22H : cinéma UGC chatelet

L'utilisateur rentre l'heure de début et l'heure de fin et appui sur afficher. La carte et le planning correspondant aux différentes activités et aux différentes horraires s'affiche.

Conclusion

Ce projet met en avant les possibilités qu'offre Java en matière d'application web. En développant notre application sur le web nous souhaitons obtenir un logiciel utilisable par tous rapidement et simplement par Internet. Ce dossier présente ses composantes principales ainsi que leur utilisation dans l'application. Le projet est présenté tel que nous l'imaginons et nous espérons pouvoir fournir un travail à la hauteur de ce qui est montré dans ce dossier. Celui-ci nous servira de support tout au long du développement et les différents schémas feront office de structure pour l'implémentation qui va suivre. Le développement d'essais d'applications ayant été fait parallèlement à ce dossier, nous avons déjà une idée précise de la direction dans laquelle porter le développement de l'application. Ce travail de préparation préalable va nous permettre de commencer rapidement la réalisation de l'application tout en ayant une structure nous montrant clairement les objectifs et les limites du projet.

Bibliographie

<https://github.com/Noneji/nightjawaj> (GitHub de Nightjawaj)
<http://sparkjava.com/> (Site officiel de la librairie Spark)
<http://sparkjava.com/documentation.html> (Documentation de la librairie Spark)
<https://developers.google.com/places/?hl=fr> (API Google Places)
<http://json.org/> (Documentation de JSON)
<https://dev.mysql.com/doc/connector-j/5.1/en/> (Mysql connector Java, JDBC mysql pour Java)
<https://fr.wikipedia.org/wiki/MySQL>
[https://fr.wikipedia.org/wiki/Java_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage))
https://fr.wikipedia.org/wiki/World_Wide_Web