

INFO608:
Jygo, créer et localiser des évènements

I – Cahier des charges

L'application Jygo est un gestionnaire d'évènements. Les utilisateurs peuvent créer et participer à des évènements. L'application est destinée à toutes les personnes qui ont besoin de créer des évènements, que ce soit pour une organisation (associations etc...) ou pour une utilisation privée. En fonction du temps disponible nous pourrions implémenter des fonctions supplémentaires.

Voici un récapitulatif de tout ce que doit faire l'application:

Catégorie	Fonctionnalités
Gestion des utilisateurs	<ul style="list-style-type: none">• Inscription classique: comme sur la plupart des sites web actuels• connexion classique également: login/mot de passe• Gestion des droits utilisateurs: administrateur, utilisateur inscrit, utilisateur non inscrit
Gestion des évènements	<ul style="list-style-type: none">• Création d'évènement : évènement physique seulement (pas d'évènement online), annulation possible• Participation à un évènement: l'utilisateur indique qu'il participe à un évènement• Historique des évènements: les utilisateurs peuvent voir les évènements auxquels ils participent et ceux qu'ils ont créés.• Noter un évènement: l'utilisateur peut noter l'évènement et une note sera attribuée à l'organisateur• Recherche basique d'évènement: recherche en fonction de la localisation de l'utilisateur et dans un rayon fixé• Affichage d'un évènement en fonction de leur date: sous forme d'un calendrier
Ajouts possibles	<ul style="list-style-type: none">• Prise en charge des évènements online• Prise en charge d'évènements réservés aux adultes, prise en charge d'un âge minimal/maximal en fonction de l'évènement• Système de commentaires/tchat pour chaque évènement• Système d'ami: l'utilisateur peut inviter ses contacts sur l'application et devenir ami avec les autres utilisateurs• Prise en charge de la confidentialité: l'utilisateur change la visibilité des évènements qu'il a créés• Bloquer un utilisateur: l'utilisateur peut choisir de bloquer un utilisateur en particulier, cet utilisateur ne pourra pas voir les évènements de l'utilisateur qui l'a bloqué• Ajout de critères supplémentaires pour les recherches/affichage

II – Veille concurrentielle

Nous avons effectuer ces comparaisons en partant du principe que Jygo possédera toute les fonctionnalités inscrites dans le cahier des charges, évidemment le résultat dépendra du temps de développement.

Il y a évidemment plusieurs points commun récurrents entre Jygo et ses concurrents:

- recherche d'évènement
- inscription/connexion
- affichage sous forme d'un calendrier

Application	Points commun	Différences
Balumpa	<ul style="list-style-type: none">• Géolocalisation• utilisateurs amis(via facebook)	<ul style="list-style-type: none">• Pas de création d'évènement• Affichage d'une carte avec les événements proche de l'utilisateur• impossible de noter un événement
ExitToLive	<ul style="list-style-type: none">• Géolocaision en fonction de la position et d'un rayon• utilisateurs amis	<ul style="list-style-type: none">• Pas de création d'évènement• impossible de noter un événement
SoonNight		<ul style="list-style-type: none">• Spécialisé dans les soirées/fêtes• offre de pass de réduction pour ces soirées• album photo• impossible de noter un événement
Facebook	<ul style="list-style-type: none">• Création d'évènement• visibilité et confidentialité(privé/public)• système de commentaire	<ul style="list-style-type: none">• impossible de noter un événement(ou indirectement)• l'application n'est pas dédiée à la gestion d'évènement

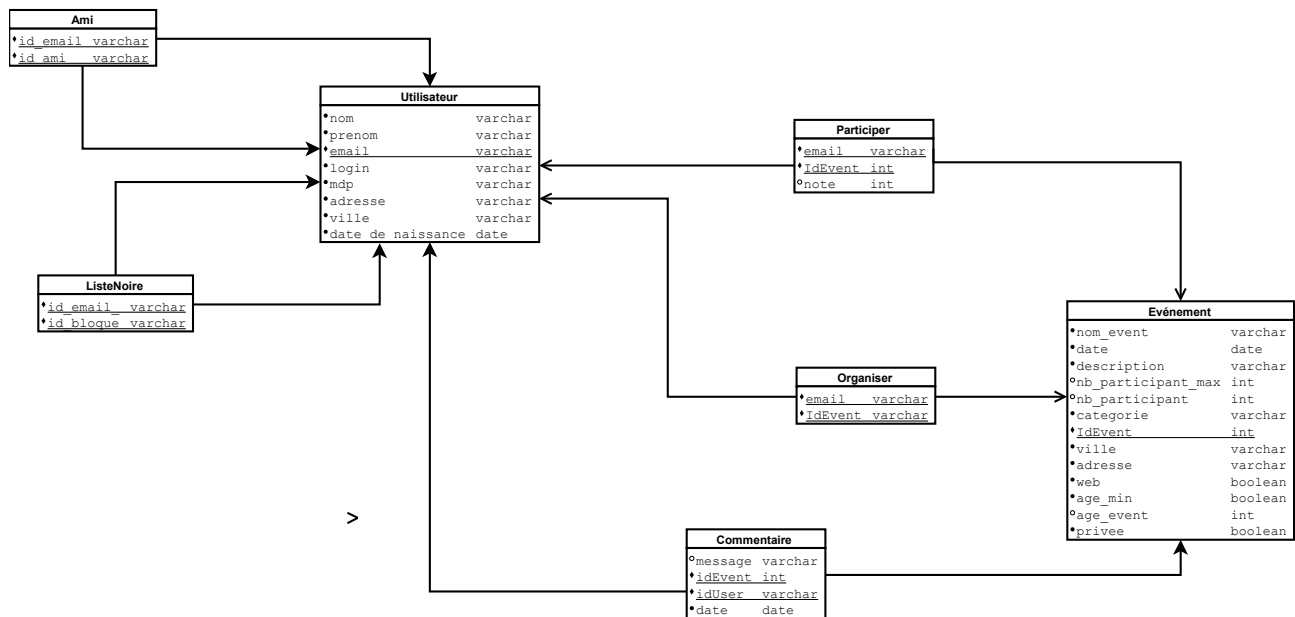
La partie "événement" de Facebook est pour l'instant ce qu'il y a de plus proche de notre application. La plus différence principale est que la gestion d'évènement est la base de notre application, contrairement à Facebook où elle a été rajouté par la suite et n'est pas son but premier. Et aussi, Facebook est de très loin le concurrent le plus influent puisque qu'il y a 1,4 milliard de compte Facebook actif actuellement(presque la moitié des internautes).

III – Modélisation de la base de données

La modélisation prends en compte toutes les fonctionnalités inscrites dans le cahier des charges:

- inscription/connexion
- événements avec prise en charge de l'âge, des événement online etc...
- gestion des utilisateurs amis/bloqués
- système de commentaires pour chaque événement

Voici le modèle logique de données que l'on pourra utiliser pour créer notre base de données:



IV – API: communications serveurs/application

Nous utiliserons la librairie Jdom pour gérer les informations au format XML. Nous choisissons Jdom parce que tous les membres du groupe ont eu l'occasion de s'en servir au premier semestre(UE programmation client/serveur).

L'objet Document est l'objet utilisé par Jdom pour décrire un fichier XML.

Voici les entêtes des fonctions que nous programmerons et utiliserons pour communiquer avec le serveur depuis l'application:

Entête	Description
Document sql_query(String query)	Effectue une requête SQL et retourne un Document Jdom en réponse. Pour simplifier nous supposons que les informations nécessaires pour se connecter à la base de données(url,login, mot de passe etc...) sont constant.
Boolean connect(String login, String pwd)	Vérifie si l'utilisateur (login,pwd) est inscrit.
Document user(String login, String pwd)	Récupère les informations concernant l'utilisateur depuis le serveur.
Boolean register(Document user)	Ajoute un utilisateur à la base de donnée si il n'est pas déjà inscrit.
Document getEvents() Document getEvents(String date1, String date2)	Par défaut: récupère les événements programmés pour les 7 prochains jours. Sinon: récupère les événements programmés entre date1 et date2
Boolean addEvents(String login, String mdp, Document event)	Ajoute un nouvel événement associé à l'utilisateur dans la base de données.
Boolean followEvents(String login, String mdp, int event)	Ajoute l'événement(représenté par sa clé) à la liste de participation de l'utilisateur.
Double getScore(int event) Double getScore(String login)	Récupère la note associé à un événement ou à un organisateur.
Boolean followUser(String user,String pwd, String login)	Ajoute un utilisateur à la liste d'amis de l'utilisateur courant.
Boolean block(String user, String pwd, String login)	Ajoute un utilisateur à la liste noire de l'utilisateur courant.
Boolean comment(String user, String pwd, int event, String msg)	Ajoute un commentaire à un événement.

