# Réseau social d'organisation de sorties Osons sortir

Rim El jraidi Idriss Sajide Maxime Vernet Gabriel Manissadjian



#### Sommaire

- I. Rappel du sujet et cahier des charges
- II. Technologies employées
- III. Architecture technique
- IV. Réalisations techniques
- V. Gestion de projet
- VI. Outils
- VII. Métriques logiciels
- VIII. Conclusion
  - IX. Démonstration

#### Contexte et sujet

Il existe des applications d'organisation de sorties ou d'événements avec des fonctionnalités payantes.

On aimerait proposer ce service gratuitement aux étudiants de l'université de Grenoble.

Si notre public test valide l'application, pourquoi pas étendre la tranche d'utilisateurs à n'importe qui et dans toutes les villes de France.

#### Objectifs du projet

• Développer une plateforme de participation à des activités proposées directement et librement par les utilisateurs.

(Tout en limitant certaines actions comme en ayant une pré sélection d'images de couvertures d'activité ou en modérant les utilisateurs en fonction de leurs notes et de leur avis)

=> Afin d'éviter que la plateforme regorge d'utilisateurs malveillants

Laisser notre application en open source sur Github.

#### Cahier des charges

- S'inscrire sur le site et recevoir un email de confirmation
- Se connecter
- Changer son mot de passe en cas d'oubli
- Modifier son profil
- Rechercher des sorties
- S'inscrire à une sortie
- Créer une sortie
- Éditer les informations de la sortie
- Consulter sa liste de sorties à venir
- Avoir accès à son historique de sorties
- Consulter des informations plus détaillées sur la sortie que l'on a créé (liste d'inscrits, date, lieu etc.)
- Créer un groupe
- Rechercher des groupes

- Demander l'adhésion à un groupe
- Consulter ses groupes
- Accepter / refuser un membre dans le groupe que l'on a créé
- Laisser un commentaire sur une sortie
- Utilisateur administrateur
- Consulter et gérer la liste des utilisateurs mal notés
- Consulter la liste des utilisateurs
- Bannir un utilisateur
- Supprimer / modifier un commentaire
- Inviter un utilisateur à rejoindre un groupe
- Accepter / refuser une invitation à rejoindre un groupe

## Équipiers et rôles

Rim

**Product Owner** 

Développeur :

Backend

Maxime

Scrum Master

Développeur :

rontend

Gabriel

Développeur :

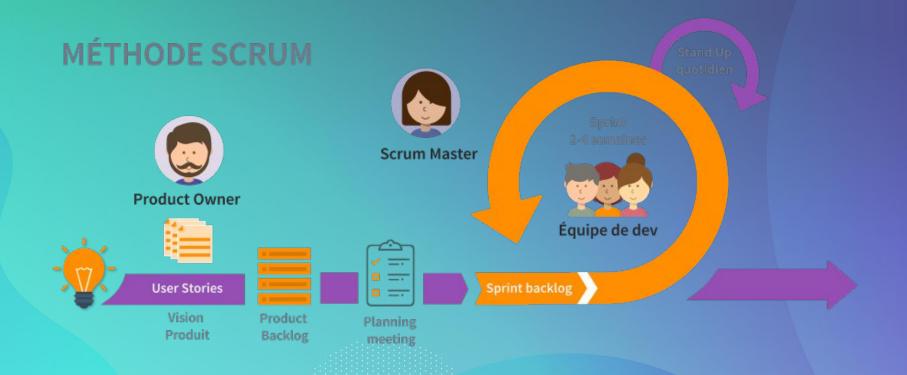
Backend

**Idriss** 

Développeur :

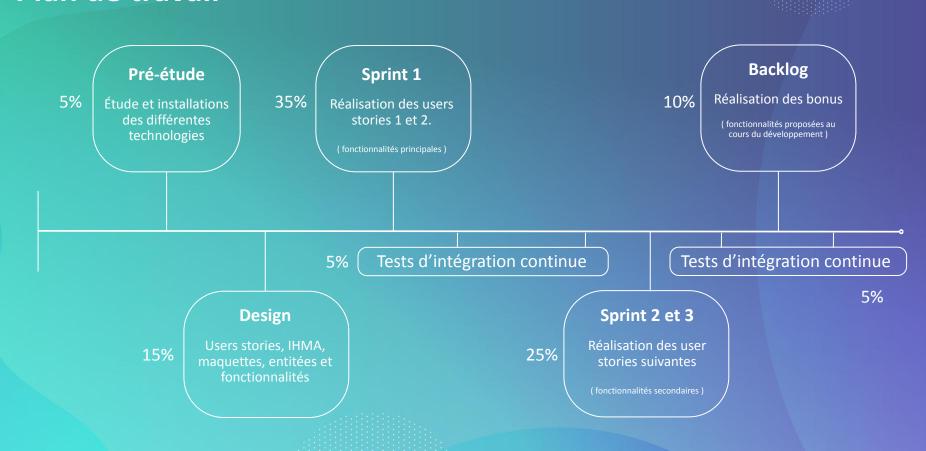
Frontend

### Fonctionnement du groupe





avancement du projet



#### Technologies utilisées - Backend



Fournit des outils et des fonctionnalités utiles qui facilitent la création d'applications web en Python **SQLAlchemy** 

Toolkit open source + mapping objet relationnel

#### **Technologies utilisées - Frontend**





Langage très documenté et rapide à apprendre Beaucoup de ressources des autres utilisateurs en ligne Fournit des composants graphiques Très documenté Beaucoup de composants facilement réutilisable

#### **Outils**

#### Gestion de projet



Partage des documents, du code, communication interne sur le code.



Discord - Communication interne

#### Design



Balsamiq - Réalisation de l'IHM abstraite



Figma - Réalisation de la maquette

#### **Architecture - Single page application**

- Accessible via une adresse URL dans un navigateur
- Une seule application pour tous les supports : comportement responsive (la page adapte dynamiquement son contenu à la taille du dispositif).
- Données accessibles via notre base de donnée privée et sécurisée.
- Chargement à la volée (le chargement des données nécessaires s'effectue au fur et à mesure des fonctionnalités et des filtrages activés par l'utilisateur).
- Optimisation statique : peu importe la complexité fonctionnelle de l'application, elle sera vue comme une collection de simples pages html.

### **Architecture - Composants**



**Interface utilisateur** 







Serveur



**GQLAlchemy** 



#### Réalisations techniques

- Création base de données
- Développement API
- Requêtes
- Réalisation de tests
- Sécurité : mot de passes cryptés dans la base de données

http://localhost:5000/api/swagger.json Explore

#### **Swagger REST Article**

Commentaire

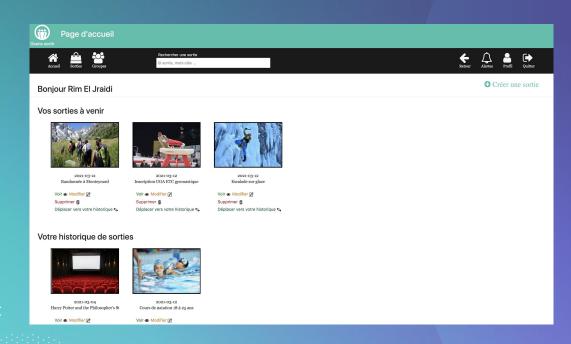
This is the swagger file that goes with the server code

Commentaire		Show/Hide List Operations Expand Operations
GET	/com	Get all commentaires
GET	/sortie/{id_sortie}/com	Get list of comments of an activity
POST	/sortie/{id_sortie}/com	Comment an acitivity
DELETE	/sortie/{id_sortie}/com/{id_com}	Delete a comment from the sortie
GET	/sortie/{id_sortie}/com/{id_com}	Get one comment of one activity by id
PUT	/sortie/{id_sortie}/com/{id_com}	Update a comment
Demande		Show/Hide   List Operations   Expand Operations
GET	/demandes	Get all demandes
DELETE	/groupe/{id_groupe}/demandes	Refuse an invitation to join a groupe
GET	/groupe/{id_groupe}/demandes	Get list of demandes of a groupe
POST	/groupe/{id_groupe}/demandes	Add someone to a group
GET	/groupe/{id_groupe}/demandes/{id_demande}	Get one demande of one groupe by id
Sortie		Show/Hide   List Operations   Expand Operations
GET	/filter/{type_sortie}	The sortie data structure supported by the server application
GET	/search/{search}	Get one sortie by name
GET	/sortie	Get all the sortie
POST	/sortie	Create a sortie
DELETE	/sortie/{id_sortie}	Delete a sortie from the sortie list
GET	/sortie/{id_sortie}	Get one sortie by id
PUT	/sortie/{id_sortie}	Update a sortie

#### Réalisations techniques

#### **Fonctionnalités**

- Inscription
- Création, modification et suppression de groupes et de sorties
- Inscription aux sorties et adhésion aux groupes
- Visualisation des sorties à venir et de l'historique des sorties
- Système d'envoi d'invitations pour rejoindre un groupe
- Recherche et filtrage des sorties par catégorie, nom ou ID
- Réalisation d'un header, d'une navbar et d'un footer



#### Métriques logiciels

→ COCOMO (Constructive Cost Model)

Notre application est relativement petite, on considère donc qu'elle est organique

- Serveur: 2370 lignes

- Client: 4130 lignes

Effort = 2.4\* (6500/1000)  $^{1,05}$  = 17.1 mois-homme

Un développeur débutant en France a un salaire de 3100€ brut/mois

Charges: 150€ pour les déplacements

★ Développer l'application coûterait donc environ 54000€

Mais tous les scénarios ne sont pas finis

#### Conclusion

- Projet intéressant
- Découverte de nouvelles technologies
- Découverte de nouvelles méthodes de gestion de projet (Poker planning)
- Approfondissement des méthodes de gestion de projet
- Autonomie gestion du projet, prises de décisions
- Manque de temps pour la réalisation de toutes les fonctionnalités

## Démonstration

## Merci de votre attention!



#### Rétrospectives et axes d'amélioration

- Se concentrer davantage sur les tests
- Améliorer la qualité des tests
- S'assurer continuellement de la qualité de la structure du code
- Ne pas se précipiter, réflexion sur le choix de certaines technologies en amont

#### Continuité du projet

- Faire fonctionner la connexion du côté Frontend
- Fonctionnalités liées à la modération des utilisateurs
- Lier l'espace commentaire (côté front) aux requêtes liées à cette classe (côté back)
- Sécurité
- Déploiement
- Passer les sorties finies dans l'historique automatiquement