

# Cahier des charges v1

URL de l'application : <https://totum.ovh>

## **Présentation du projet**

Vous vouliez faire du tennis, mais votre ami, avec qui vous deviez jouer, n'est finalement plus disponible.

Vous vouliez faire un foot, mais vos amis ont finalement préféré profiter de leur dimanche aprem pour faire la sieste.

Vous voulez organiser une soirée jeux de rôle, mais l'un des participants vous a fait faux bond.

Afin de remédier à ça, il y a Totum.

Totum est une application qui met en relation des particuliers d'une même ville afin qu'il puisse avoir une personne avec qui faire des activités sportives et ludiques.

Cela permettra à des passionnés d'une même activité d'entrer en contact afin qu'ils puissent partager et pratiquer leur activité favorite.

En plus du fait de pouvoir rechercher une activité à rejoindre, les utilisateurs pourront également proposer des activités.

## **Définition des besoins**

Les besoins :

- Beaucoup de personnes manquent de partenaires pour participer à des activités collectives sportives ou ludiques
- Totum facilite la mise en relation et l'organisation de telles activités en prenant en compte la localisation et les aspirations de chaque participant-e-s

Solution :

- Totum prend la forme d'un réseau social qui permet de s'inscrire pour rejoindre une communauté d'utilisateurs qui souhaitent partager leurs passions
- Chaque utilisateur peut proposer des activités et participer à celles proposées par les autres
- L'application permet également d'archiver les activités passées et propose un chat entre participants afin de peaufiner l'organisation de l'évènement futur
- Elle permet aussi de visualiser les activités disponibles sur une zone géographique donnée

## **MVP**

- Pouvoir faire une recherche d'activité dans une zone
- Pouvoir faire une recherche d'activité par catégorie
- Création de compte
- Gestion du compte
- Connexion
- Se déconnecter du compte
- Suppression du compte
- Création d'activité
- Suppression d'activité
- Modification d'activité
- Accepter/ refuser participant dans l'activité
- Se désinscrire de l'activité
- Zone commentaires
- Consulter les profils utilisateurs
- Formulaire de contact
- Accès zone administrateur / visu des comptes utilisateurs, messages de réclamations, (stats) , envoyer mail

### évolution potentielle :

- Ajout d'une alerte météo
- Liste d'amis
- Chat de discussion entre amis
- Connexion avec login Google et Facebook
- Sortir un participant de l'activité
- Modération du chat par IA
- Archivage des activités
- Album photos (????)
- Géolocalisation / interface carte
- Bannir utilisateur
- Mail de récupération de mdp
- Chat de discussion dans l'activité
- L'archivage des activités
- Accès au profil utilisateur
- Modification du profil utilisateur / photo / pseudo / mail

## **Spécifications techniques**

Frontend :

- JavaScript/REACT
- Frameworks CSS : Tailwind, Bulma??
- Axios

Backend :

- JavaScript
- NodeJS
- Express
- Postgresql (database)
- MongoDB (database chat)
- Sqitch (management database)
- JSDoc
- Bcrypt

Backend et frontend :

- JEST
- Eslint
- Git / Github

Déploiement :

- Docker / Docker compose
- Github action

## **Définition de la cible du projet**

- Les sportifs voulant faire des activités a plusieurs
- Les personnes arrivant dans une nouvelle ville et ne connaissant personne
- Les personnes voulant faire des activités ludiques à plusieurs
- Les personnes voulant proposer à d'autres personnes de les rejoindre pour participer à des activités

## **Les navigateurs compatibles**

- Safari
- Chrome

- Firefox
- Edge
- Opera
- IE (Sans forcer)

## Arborescence de l'application

Routes back :

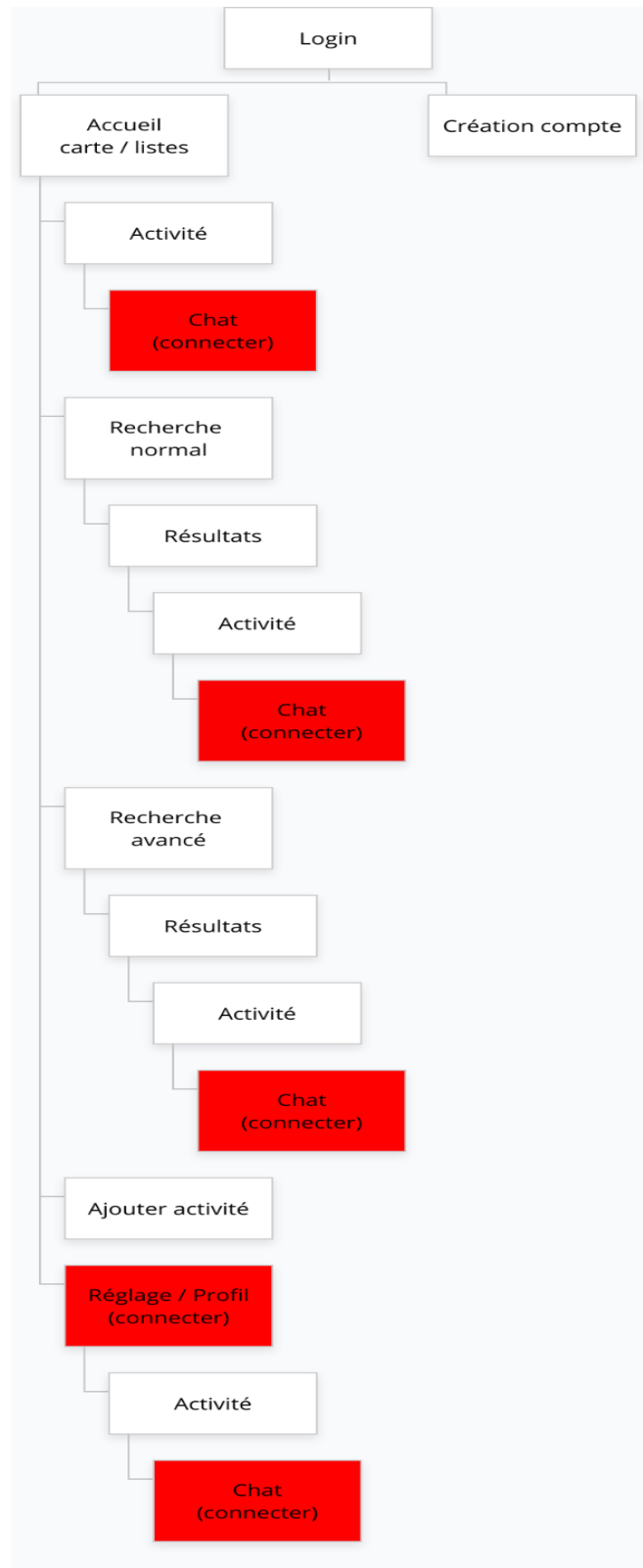
(<https://api.totum.ovh/v1>)

Routes	Contrôleur	GET	POST	PATCH	DELETE	DESCRIPTION
Routes utilisateurs						
/user/login	user/login	✗	✓	✗	✗	connexion et création de session
/user/logout	user/logout	✓	✗	✗	✗	destruction de session
/user/createNew	user/createUser	✗	✓	✗	✗	création nouvel utilisateur
/user/:id/manage	- user/getOneUser - user/updateUser - user/removeUser	✓	✗	✓	✓	gestion profile utilisateur
Routes activités						
/activities	activity/getAll	✓	✗	✗	✗	récupération de toutes les activités
/activities/:category	activity/getByCategory	✓	✗	✗	✗	récupération des activités par catégorie
/activities/:geo	activity/getByGeo	✓	✗	✗	✗	récupération des activités par géo-localisation
/activities/:search	activity/getBySearch	✓	✗	✗	✗	recherche activité par mot clé
/activity	activity/createActivity	✗	✓	✗	✗	création nouvelle activité
/activity/:id/manage	- activity/getOneActivity - activity/updateActivity - activity/removeActivity	✓	✗	✓	✓	gestion des activités

## Routes front :

Routes	Description
/login	connection
/logout	déconnecter
/home	accueil desktop
/home/map	accueil carte
/home/list	accueil liste
/search	page recherche
/activity/list	liste des activités
/activity/{name}	page de l'activité
/activity/{name}/chat	chat de l'activité
/activity/member	participant activité
/activity/{name}/manage	page de gestion activité
/activity/create	page de création activité
/profil/:id	page profil
/profil/create	création de profil
/profil/params	paramètre utilisateur
/profil/param/notifications	paramètre utilisateur notification
/mentions	mentions légale
/profil/help	formulaire de contact
/profil/changeMode	choix mode sombre

Structure :



## User stories

En tant que	Je veux pouvoir	Dans le but de	Version
Visiteur	avoir accès à une activité	voir l'activité en détails	
Visiteur	accéder à la liste des activités	avoir une vue d'ensemble des activités proposées	
Visiteur	faire une recherche d'activité	trouver une activité selon des critères précis	
Visiteur	choisir d'être géolocalisé	pour circonscrire ma recherche à une zone géographique	
Visiteur	créer un compte	pouvoir me loguer	
Visiteur	me loguer	changer de rôle et avoir accès aux fonctionnalités user	
Visiteur	écrire aux administrateurs	en cas de problème	
Visiteur	récupérer mdp du compte	récupérer mon mdp perdu	V2
Visiteur	visualiser les stats de connexion du site	connaître l'activité du réseau social	V2
Utilisateur	créer une activité	la proposer à la communauté	
Utilisateur	se déconnecter	fermer sa session	
Utilisateur	Supprimer mon compte	de supprimer mes accès	
Utilisateur	créer mon profil	ajouter photo/pseudo/mail...	
Utilisateur	modifier son profil	changer photo/pseudo/mail...	

Utilisateur	m'inscrire à une activité	participer à une activité	
Utilisateur	consulter les profils des utilisateur-ice-s	connaître les autres utilisateur-ice-s pour une activité	
Utilisateur	modifier une activité	mettre à jour l'activité	
Utilisateur	supprimer une activité	annuler une activité	
Utilisateur	visualiser les acceptations des organisateurs	savoir si je peux participer à une activité	
Utilisateur	prévenir de la suppression d'activité	d'alerter tous les participants	
Utilisateur	accès à ma liste de camarade	voir s' ils sont connectés, discuter	
Utilisateur	communiquer avec mes camarades	d'échanger sur n'importe quel sujet	
Utilisateur	signaler utilisateur	d'alerter des mauvais comportements	
Utilisateur	Accepté les demandes de participations	Ajouter qui l'on souhaite dans l'activité	
Utilisateur	Bloquer les messages d'un utilisateur	ne plus être importuné	V2
Utilisateur	communiquer avec les participants	d'échanger sur l'activité	V2
Utilisateur	ajouter photos dans l'activité	garder des souvenirs	V2
Administrateur	accès aux messages de réclamations	régler les incidents rencontrés	



Administrateur	supprimer des activités	modérer les activités inappropriées	
Administrateur	envoyer un mail	contacter un-e user	V2
Administrateur	visualiser les stats du site	connaître l'activité du réseau social	V2
Administrateur	bannir les utilisateurs	bannir les mauvais joueurs	V2

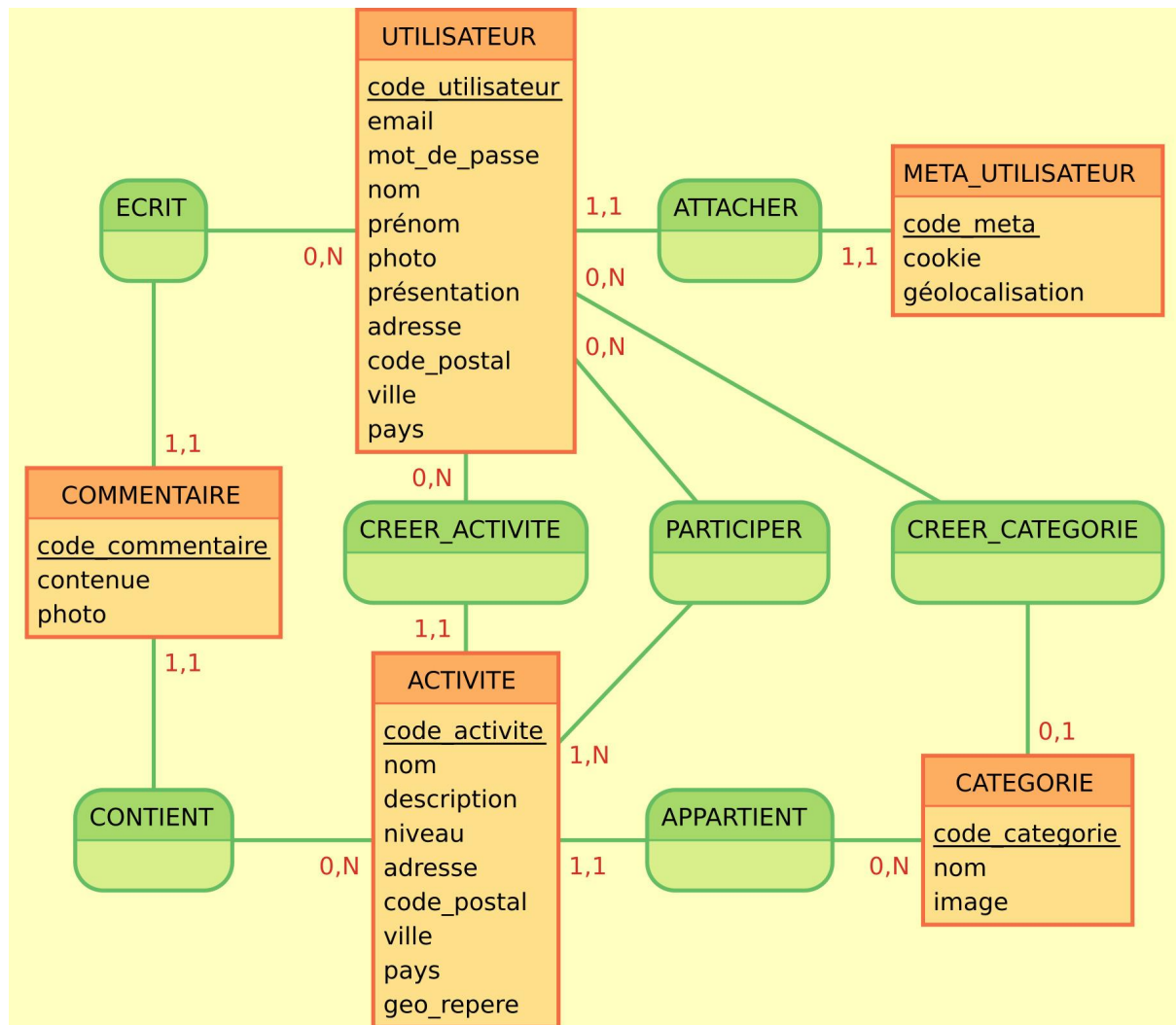
## Équipe

- **Product Owner** : Anas
- **Scrum Master** : Benoit
- **Référents Techniques** : Kenny
- **Git Master** : Damien
- **Lead dev front** : Xavier
- **Lead dev back** : Kenny

## Documents base de données

MCD :

- ECRIT, 11 COMMENTAIRE, 0N UTILISATEUR
- UTILISATEUR: code\_utilisateur, email, mot\_de\_passe, nom, prénom, photo, présentation, adresse, code\_postal, ville, pays
- ATTACHER, 11 UTILISATEUR, 11 META\_UTILISATEUR
- META\_UTILISATEUR: code\_meta, cookie, géolocalisation
- COMMENTAIRE: code\_commentaire, contenu, photo
- CREER\_ACTIVITE, 0N UTILISATEUR, 11 ACTIVITE
- PARTICIPER, 0N UTILISATEUR, 1N ACTIVITE
- CREER\_CATEGORIE, 0N UTILISATEUR, 01 CATEGORIE
- CONTIENT, 11 COMMENTAIRE, 0N ACTIVITE
- ACTIVITE: code\_activite, nom, description, niveau, adresse, code\_postal, ville, pays, geo\_repere
- APPARTIENT, 11 ACTIVITE, 0N CATEGORIE
- CATEGORIE: code\_categorie, nom, image



## MLD :

- UTILISATEUR: (code\_utilisateur, email, mot\_de\_passe, nom, prénom, photo, présentation, adresse, code\_postal, ville, pays, #code\_meta, created\_at, updated\_at)
- ACTIVITE: (code\_activite, nom, description, niveau, adresse, code\_postal, ville, pays, geo\_repere, #code\_utilisateur, #code\_categorie, created\_at, updated\_at)
- COMMENTAIRE: (code\_commentaire, contenue, photo, #code\_utilisateur, #code\_activite, created\_at, updated\_at)
- META\_UTILISATEUR: (code\_meta, cookie, géolocalisation, #code\_utilisateur, created\_at, updated\_at)
- CATEGORIE: (code\_categorie, nom, image, #code\_utilisateur, created\_at, updated\_at)
- PARTICIPER: (code\_participer, #code\_utilisateur, #code\_activite, created\_at, updated\_at)

## Dictionnaire de données :

CHAMP	TYPE	SPÉCIFICITÉS	DESCRIPTION
TABLE UTILISATEUR ( <b>users</b> )			
<b>id</b>	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	identifiant unique utilisateur
<b>email</b>	TEXT -> DOMAIN EMAIL / REGEX	NOT NULL UNIQUE	email utilisateur
<b>password</b>	TEXT	NOT NULL	hash mot de passe
<b>firstname</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	prenom
<b>lastname</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	nom
<b>picture</b>	TEXT	UNIQUE	liens vers la photo de profil
<b>about</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)		texte de présentation
<b>address</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)		rue et numéro de rue
<b>zip_code</b>	INT ou TEXT -> DOMAIN ZIP_CODE / REGEX	NOT NULL	code postale
<b>city</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	ville
<b>country</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	pays
<b>meta_id</b>	INT	NOT NULL REFERENCES user("id")	clé étrangère meta utilisateur
<b>created_at</b>	TIMESTAMPZ	NOT NULL DEFAULT NOW()	date de création
<b>updated_at</b>	TIMESTAMPZ		date de dernière mise à jour

TABLE ACTIVITE (activity)			
<b>id</b>	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	identifiant unique activité
<b>name</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	Nom de l'activité
<b>description</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	présentation de l'activité
<b>level</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	Niveau de l'activité
<b>address</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	adresse activité
<b>zip_code</b>	INT ou TEXT -> DOMAIN ZIP_CODE / REGEX	NOT NULL	code postal
<b>city</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	ville
<b>country</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	pays
<b>landmark</b>	TEXT		géo repères
<b>id_user</b>	INT	NOT NULL REFERENCES user("id")	identifiant unique utilisateur
<b>id_category</b>	INT	NOT NULL REFERENCES category("id")	identifiant unique catégorie
<b>created_at</b>	TIMESTAMPTZ	NOT NULL DEFAULT NOW()	date de création
<b>updated_at</b>	TIMESTAMPTZ		date de dernière mise à jour

TABLE COMMENTAIRES ( <b>comment</b> )			
<b>id</b>	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	identifiant unique commentaire
<b>content</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL	corps du commentaire
<b>picture</b>	TEXT	NOT NULL	liens vers l'image
<b>id_user</b>	INT	NOT NULL REFERENCES user("id")	clé étrangère créateur du commentaire
<b>id_activity</b>	INT	NOT NULL REFERENCES activity("id")	clé étrangère de l'activité
<b>created_at</b>	TIMESTAMPTZ	NOT NULL DEFAULT NOW	date de création
<b>updated_at</b>	TIMESTAMPTZ		date de dernière mise à jour

TABLE META_UTILISATEUR( <b>meta</b> )			
<b>id</b>	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	identifiant unique meta
<b>cookie</b>	BOOLEAN	NOT NULL	cookies
<b>landmark</b>	BOOLEAN	NOT NULL	géolocalisation
<b>created_at</b>	TIMESTAMPTZ	NOT NULL DEFAULT NOW()	date de création
<b>updated_at</b>	TIMESTAMPTZ		date de dernière mise à jour
<b>id_user</b>	INT	NOT NULL REFERENCES user("id")	clé étrangère utilisateur



TABLE CATEGORIE(category)			
<b>id</b>	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	identifiant unique catégorie
<b>name</b>	TEXT -> DOMAIN TEXT + / REGEX (pas de symbole ou autre)	NOT NULL UNIQUE	nom de la catégorie
<b>picto</b>	TEXT		balise react picto
<b>id_user</b>	INT	REFERENCES user("id")	clé étrangère utilisateur / créateur
<b>created_at</b>	TIMESTAMPTZ	NOT NULL DEFAULT NOW()	date de création
<b>updated_at</b>	TIMESTAMPTZ		date de dernière mise à jour

TABLE PARTICIPER (user_activity)			
<b>id</b>	INT	GENERATED ALWAYS AS IDENTITY PRIMARY KEY	identifiant unique participé
<b>id_user</b>	INT	NOT NULL REFERENCES user ("id")	identifiant unique utilisateur
<b>id_activity</b>	INT	NOT NULL REFERENCES activity ("id")	identifiant unique activité
<b>created_at</b>	TIMESTAMPTZ	NOT NULL DEFAULT NOW()	date de création
<b>updated_at</b>	TIMESTAMPTZ		date de dernière mise à jour

## **Wireframes**

Landing page :