

---

*LAVENANT Jordan*

*BUT Informatique - 2024 2025*

*LUCIDOR Léo*

*MALLERON Daniel*

*PIGOREAU Nathan*

# Rapport

# Explore Game

**S5.01.A Dév- avancée**

**2024 - 2025**



# SOMMAIRE

<b>SOMMAIRE.....</b>	<b>2</b>
<b>1. INTRODUCTION.....</b>	<b>5</b>
<b>2. CONTEXTE.....</b>	<b>5</b>
Projet Explore Game.....	5
Support existant.....	6
Cahier des charges client.....	7
<b>3. ORGANISATION ET OUTILS.....</b>	<b>7</b>
Répartition des tâches.....	8
Versionnage.....	9
Communication.....	9
Organisation.....	10
Schéma.....	10
Stockage.....	10
Design.....	10
<b>4. ENJEUX.....</b>	<b>11</b>
Enjeux de l'expérience utilisateur.....	11
Enjeux techniques.....	12
<b>5. ANALYSE ET CONCEPTION.....</b>	<b>13</b>
<b>6. ARCHITECTURE TECHNIQUE.....</b>	<b>18</b>
Core.....	19
Client.....	24
Type-Sharing.....	28
<b>7. ETAT DU PROJET.....</b>	<b>30</b>
Fonctionnalités et interfaces réalisées.....	31
Interfaces Utilisateur.....	31
Fonctionnalités.....	31
Fonctionnalités interactives.....	32
Images des interfaces.....	33
Client.....	33
Core.....	35
Interfaces par-défaut.....	35
Interfaces modernisées.....	38
Organisation.....	39
<b>8. AMÉLIORATIONS À VENIR.....</b>	<b>39</b>
Déploiement V1 sur les serveurs de l'IUT.....	39
Amélioration des IHM du Core.....	39
Harmonisation du micro service client.....	40

Question choix multiples.....	40
Intégrer la fonctionnalité du timer et des pénalités associées.....	40
Retirer les questions ouvertes.....	40
Bucket d'images pour enrichir l'expérience.....	41
Page des avancements des scénarios.....	41
Trophés et succès.....	41
<b>9. CONCLUSION.....</b>	<b>42</b>
<b>10. GLOSSAIRE.....</b>	<b>44</b>
<b>11. ANNEXES.....</b>	<b>48</b>
Figure 1.....	48
Figure 2.....	49
Figure 3.....	50
Figure 4.....	50
Figure 5.....	51
Figure 6.....	51
Figure 7.....	52
Figure 8.....	53
Figure 9.....	53
Figure 10.....	54
Figure 11.....	54
Figure 12.....	55
Figure 13.....	56
Figure 14.....	56
Figure 15.....	57
Figure 16.....	57
Figure 17.....	58
Figure 18.....	58
Figure 19.....	59
Figure 20.....	60
Figure 21.....	61
Figure 22.....	62
Figure 23.....	63
Figure 24.....	64
Figure 25.....	65
Figure 26.....	66
Figure 27.....	67
Figure 28.....	68
Figure 29.....	69
Figure 30.....	70

Figure 31.....	71
Figure 32.....	71
Figure 33.....	72
Figure 34.....	73
Figure 35.....	73

## **1. INTRODUCTION**

Dans le cadre de notre formation en Bachelor Universitaire de Technologie (BUT) Informatique, nous développons des compétences à la fois techniques et méthodologiques, dans des domaines tels que le développement logiciel, la gestion de projets et l'innovation numérique. Cette formation met l'accent à associer théorie et pratique à travers des projets concrets, reflétant les enjeux du monde professionnel.

Les Situations d'Apprentissage et d'Évaluation (SAE) occupent une place centrale dans cette démarche, nous confrontant à des problématiques réelles formulées par des clients ou partenaires. Ces projets majeurs nous permettent de travailler en équipe, de développer notre créativité et de répondre aux attentes spécifiques tout en respectant des contraintes techniques et organisationnelles.

Pour cette SAE, nous avons eu pour mission de moderniser un concept existant : les Explore Games de l'IUT d'Orléans, illustrant des jeux d'énigmes, initialement conçus sur papier. L'objectif était de concevoir une solution numérique sous la forme d'une application web, offrant une expérience plus immersive et interactive. Ce projet illustre l'importance de notre formation dans la création de solutions innovantes répondant à des besoins concrets.

## **2. CONTEXTE**

### Projet Explore Game

L'outil Explore Game a été conçu par l'IUT d'Orléans dans le but de valoriser ses formations et activités. Il s'adresse aussi bien aux nouveaux étudiants de première année qu'à des collégiens, lycéens, ou participants à des échanges académiques. Ces publics peuvent ainsi découvrir l'IUT d'Orléans, ses infrastructures et ses formations lors de visites ou d'événements.

Actuellement, cet outil repose sur un support papier et se structure autour de scénarios interactifs. Chaque scénario est composé de plusieurs étapes, chacune associée à un lieu spécifique de l'IUT (bâtiment Informatique, bâtiment GEA, bâtiment MT2E, parking,

cafétéria, etc...). À chaque étape, des questions sont proposées aux participants, en fonction de leurs lieux actuels.

Pour trouver les réponses, les utilisateurs doivent explorer le lieu indiqué, observer leur environnement (les bâtiments, les infrastructures) ou interagir avec des membres clés de l'IUT, tels que les enseignants, le personnel ou les étudiants.

L'objectif principal de cet outil est de permettre, le temps d'une demi-journée, aux participants de découvrir les différentes composantes de l'IUT d'Orléans. Ce jeu interactif contribue à améliorer leur intégration académique en les aidant à se familiariser avec leur futur environnement.

## Support existant

**Parcours Informatique (Violet)**  
(Suivez les indices de la couleur de votre parcours pour ne pas vous perdre)

Informations Personnelles :

Nom : \_\_\_\_\_ Prénom : \_\_\_\_\_  
Sexe : \_\_\_\_\_  
Filière : \_\_\_\_\_  
Heure de départ : \_\_\_\_\_  
Heure de retour : \_\_\_\_\_

Explore Game :

Chère étudiante, cher étudiant,

**Bienvenue à l'IUT'O !**  
Vous allez participer à cet Explore Game immersif au sein de l'IUT d'Orléans.  
Vous allez partir à la recherche de Diego, un étudiant très étourdi. Arrivé à l'IUT il y a quelques jours, il a malheureusement perdu son chemin en se promenant sur le campus. Votre mission est de retracer son parcours pour le retrouver.

Pour cela, vous allez devoir répondre à des questions et résoudre des énigmes dans chaque département ou bâtiment pour récolter des lettres. En combinant toutes les lettres obtenues, vous formerez un mot-clé final qui vous mènera au lieu où il s'est égaré. Répondez à ce questionnaire et rendez-vous avant 16h sur l'esplanade.

Et si vous souhaitez faire une pause, profitez des animations de l'esplanade !

**Êtes-vous prête, prêt à relever ce défi ?**

**LIEU NUMERO 1 : BATIMENT CAP**  
Inscription : Bienvenue dans le bâtiment CAP. Répondez aux questions suivantes avant de chercher l'indice qui indiquera votre prochaine destination.

- 1) Dans quel bâtiment se trouve la cafétéria ?
- 2) Citez au moins deux plats proposés à la cafétéria ?
- 3) Quelles sont les options de paiement disponibles ?
- 4) Qu'est-ce que Izly ?

Lettre à récolter :

**LIEU NUMERO 2 :**  
Bienvenue ici. Diego a eu cours dans ce département.  
Répondez à la question puis cherchez qui vous indiquera où se trouve votre prochaine et la lettre à récolter

- 5) Combien de radiateurs sont présents dans le hall géo-climatique ?

Lettre à récolter :

**LIEU NUMERO 3 :**  
Bienvenue, dans le département de Diego. Répondez aux questions 6 à 13 de votre questionnaire et trouvez le panneau d'affichage qui indiquera votre prochaine destination et la lettre à récolter.

- 6) Où se trouve le secrétariat de mon département ?
- 7) Quel est le nom du ou des secrétaires de mon département ?
- 8) A qui dois-je m'adresser si je souhaite prévenir d'une absence ?
- 9) Combien existe-t-il de parcours dans mon département ? (s'aider d'internet si besoin)

*Figure 2 : Explore Game existant (support papier)*

## Cahier des charges client

Introduit pour la première fois à la rentrée 2024, l'Explore Game a rencontré un vif succès. Ainsi, l'objectif est désormais de moderniser ce concept en le digitalisant. L'outil, initialement conçu sur support papier, sera transformé en une application numérique accessible depuis différents appareils : smartphones, tablettes et ordinateurs portables.

Cette version numérique vise à rendre l'expérience plus immersive et ludique. Des éléments tels que des scénarios enrichis, un temps limité pour résoudre les énigmes, un système de classement des participants et un système de niveau d'indice seront intégrés pour renforcer l'engagement des utilisateurs.

L'outil devra conserver sa vocation première : permettre aux participants de découvrir les infrastructures et formations de l'IUT d'Orléans. Grâce à cette version digitale, ils pourront facilement lancer un scénario, suivre des étapes interactives et répondre aux questions de manière intuitive tout en s'immergeant dans une expérience captivante, dans leur poche.

## 3. ORGANISATION ET OUTILS

L'organisation et l'attribution des tâches sont des éléments clés dans la réussite d'un projet informatique. Dès le début du projet, nous avons désigné Jordan LAVENANT comme chef de projet et porte-parole auprès des clients. Cette désignation a permis de centraliser la communication et d'assurer une coordination efficace entre les parties prenantes.

Ensuite, nous avons attribué des rôles distinctifs à chaque membre de l'équipe, en tenant compte des compétences et des points forts de chacun. Cette répartition claire des responsabilités nous a permis de travailler efficacement et de fixer des objectifs spécifiques pour chaque membre.

Pour structurer nos priorités, nous avons utilisé un tableau [Trello](#) lors de la phase initiale d'analyse. Cet outil nous a aidé à planifier les tâches, visualiser les différentes étapes du projet, et prioriser les actions nécessaires.

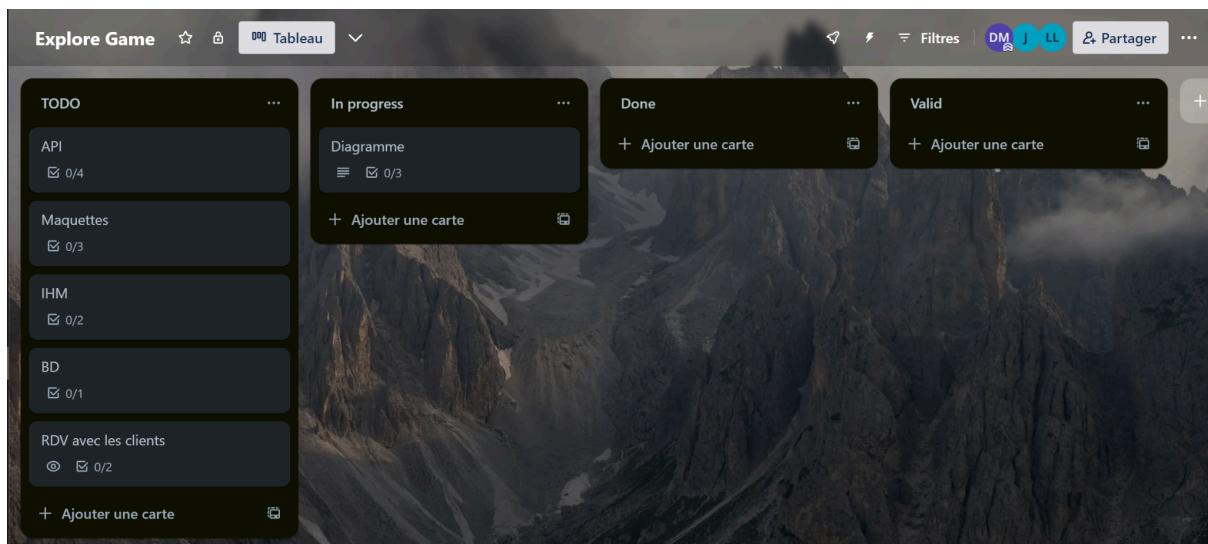


Figure 3 : Première version de notre Trello (septembre 2024)

Dès le début du projet, nous avons sollicité Mme INGARAO, directrice adjointe de l'IUT, afin d'obtenir des informations essentielles sur le fonctionnement de l'Explore Game et de clarifier nos interrogations. Une réunion de lancement a été organisée le jeudi 26 septembre, au cours de laquelle nous avons présenté notre première analyse du projet. Cette réunion a permis de recueillir les détails nécessaires pour bien comprendre les attentes liées à la digitalisation de l'Explore Game.

## Répartition des tâches

La répartition des tâches s'est faite en fonction des compétences de chacun :

LAVENANT Jordan	Chef de projet, responsable de l'architecture logicielle et développeur Full-Stack.
LUCIDOR Léo	Développeur Front-End, chargé de la réalisation des maquettes.
MALLERON Daniel	Développeur Full-Stack, chargé de la réalisation des maquettes.
PIGOREAU Nathan	Développeur Back-End

Un planning détaillé a été défini dès le départ pour respecter les délais et coordonner les différentes phases du projet, de l'analyse initiale jusqu'à la livraison finale.

Pour mener à bien ce projet, nous avons utilisé différents outils adaptés à nos besoins :

## Versionnage

[\*\*GitHub\*\*](#) a été utilisé pour le contrôle de version et la collaboration sur le code source. Grâce à ses fonctionnalités de gestion des branches et de pull requests, nous avons pu travailler simultanément tout en assurant une intégration fluide des différentes contributions.



*Github*

## Communication

[\*\*Discord\*\*](#) a été notre principal canal de communication en temps réel, facilitant les échanges rapides entre les membres de l'équipe, que ce soit par messages ou appels vocaux.



*Discord*



*Gmail*

## Organisation

**Trello** a permis de structurer nos tâches et d'organiser le projet de manière visuelle. Cet outil nous a offert une vue claire sur les étapes en cours, les actions terminées, et les échéances à venir.



*Trello*

## Schéma

[Draw.io](#) pour la conception de schémas et de diagrammes (comme les diagrammes UML ou les architectures techniques). Cet outil simple et collaboratif nous a permis de clarifier nos idées et de structurer nos documents techniques.



*Draw.io*

## Stockage

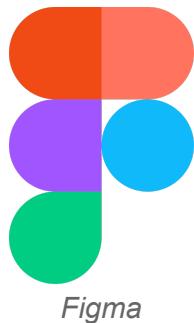
[Google Drive](#) a été notre espace de stockage centralisé pour tous les documents relatifs au projet (analyses, rapports, maquettes, etc.). Cela a assuré un accès facile et sécurisé pour l'ensemble de l'équipe.



## Design

[Figma](#) a été utilisé pour créer des maquettes interactives et des prototypes de l'interface utilisateur, permettant d'avoir un aperçu réaliste du produit final.

[Canva](#) a servi à produire des visuels et des éléments graphiques, comme des présentations, tout en maintenant une esthétique cohérente.



Figma



Canva

Grâce à ces outils et à une organisation rigoureuse, nous avons pu coordonner efficacement nos efforts et assurer une progression fluide tout au long du projet.

## 4. ENJEUX

### Enjeux de l'expérience utilisateur

L'expérience utilisateur constitue un enjeu majeur dans notre projet, car il s'adresse à un public varié, incluant des personnes non initiées à l'informatique (collégiens, lycéens, étudiants, échanges académiques), tant du côté des utilisateurs que des administrateurs.

Notre objectif principal est de concevoir une interface intuitive et accessible, permettant à tous de naviguer facilement sans prérequis techniques. Pour les utilisateurs, nous souhaitons rendre l'expérience aussi interactive que possible, notamment grâce à des dialogues immersifs, des notifications pour rappeler les étapes ou échéances, et des éléments de gamification comme un système de classement ou l'obtention de trophées et succès, qui viendront stimuler la compétition entre joueurs.

Du côté des administrateurs et gestionnaires de l'Explore Game, l'enjeu est de proposer une interface de gestion claire et fonctionnelle pour la création de scénarios et la gestion de l'application. Cette interface doit être pensée pour des personnes sans expertise informatique, afin de garantir une prise en main rapide et intuitive. Cette approche inclusive et ergonomique est essentielle pour assurer le succès et l'adoption de notre projet par tous les utilisateurs.

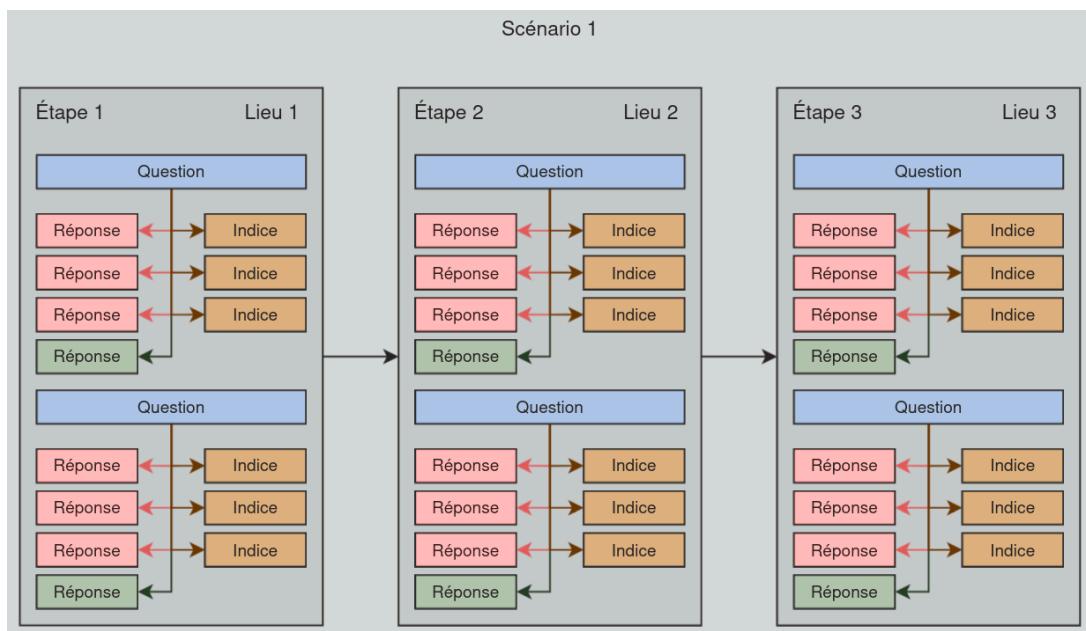
## Enjeux techniques

L'objectif était donc de concevoir une application permettant d'effectuer des Explore Game. Mais chaque énigme, chaque scénario, questions, réponses et indices associés devaient être stockés en base de données.

Ainsi, nous devions stocker ces données, afin de pouvoir les traiter (lecture, écriture, mise-à-jour et suppression). Par exemple, pouvoir sauvegarder la progression d'un utilisateur sur un scénario, vérifier ses réponses, ou encore affecter des pénalités de temps et de score par l'utilisation d'indices.

L'enjeu technique était de pouvoir concevoir une interface utilisateur viable, facilitant la gestion des différentes composantes et données relatives au bon déroulement de l'Explore Game pour les équipes de gestions du projet.

Par exemple, les équipes de gestion de l'Explore Game voudraient pouvoir créer un scénario, y affecter des questions précédemment créées, y associer des réponses, et des indices potentiels, afin de garantir un aspect de personnalisation complet, et ainsi rendre totalement modulable la gestion et l'agrémentation de données.



*Figure 5 : Modélisation de l'arborescence d'un scénario*

Le second enjeu était de judicieusement choisir nos technologies pour l'application Explore Game en elle-même, utilisée par les futurs collégiens, lycéens, étudiants et échanges académiques. En effet, ces choix ont été longuement discutés, afin de rendre la future application accessible pour tous les appareils, et tous les usagers. Ainsi, le choix d'une Web App a été retenu, afin de maximiser la compatibilité des appareils (iOS et Android), et ainsi rendre l'application accessible via le web.

Nous avons donc cloisonné l'application entre 2 microservices distincts :

- [Core](#)
- [Client](#)

Ces micro services communiquent à l'aide d'une API (pont de communication entre nos 2 applications).

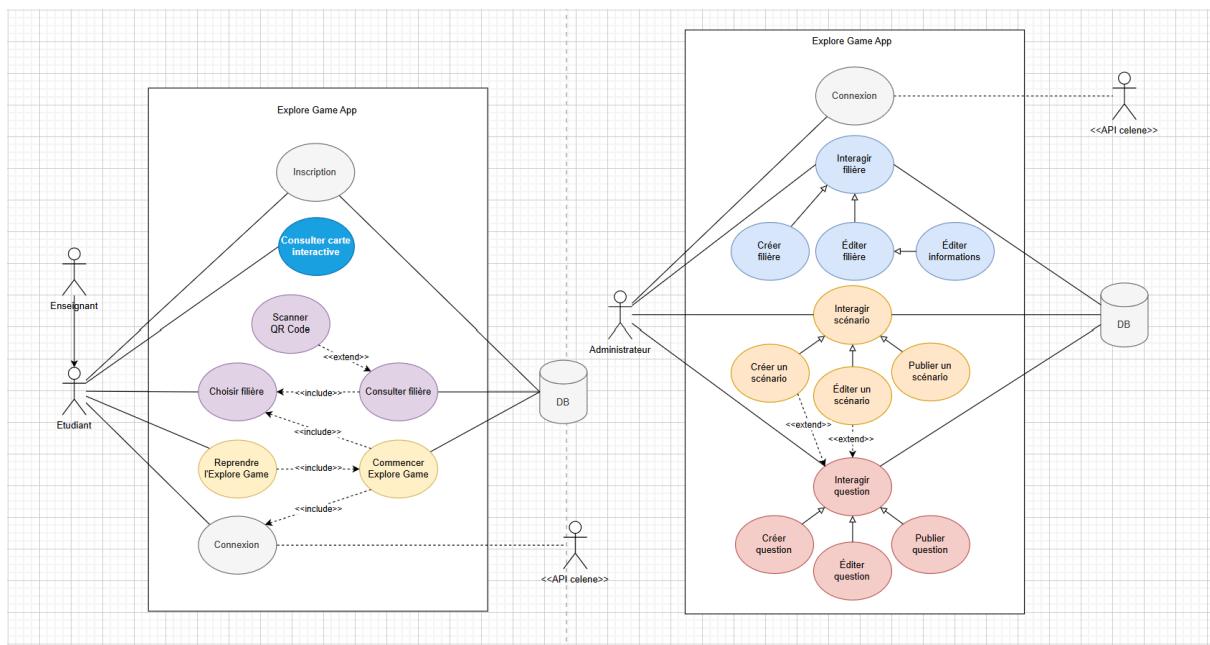
## 5. ANALYSE ET CONCEPTION

Nous avons débuté ce projet par une phase d'analyse approfondie. Au départ, notre seule source d'information était l'ancien jeu d'exploration en format papier, qui servait à la fois de référence et de cahier des charges initial. Nous avons d'abord réfléchi au format que le projet adopterait, en hésitant entre une Web App et une application mobile. Après mûre réflexion, nous avons opté pour une Web App, principalement en raison des nombreuses contraintes associées au déploiement d'une application mobile sur l'App Store d'Apple( délai trop grand pour avoir la vérification ). À ce stade, nous avons également élaboré une première version du modèle conceptuel de données (MCD) pour la base de données, puis un schéma de cas d'utilisation, ainsi qu'un schéma de l'architecture globale du projet.

Dans le cadre de notre schéma des cas d'utilisation, nous avons rassemblé toutes nos idées et notre compréhension du projet. Pour cela, nous nous sommes appuyés sur les fichiers PDF de l'ancien jeu et sur les informations fournies dans le sujet. Ce schéma nous a permis de clarifier les fonctionnalités attendues et de formaliser les actions que les utilisateurs

pourraient réaliser. Il a été un outil essentiel pour aligner l'équipe sur une vision commune du projet et pour anticiper les défis à venir.

De plus, ce schéma a grandement facilité la présentation de notre analyse aux clients. Grâce à une visualisation claire et structurée des interactions et des fonctionnalités, nous avons pu leur expliquer notre compréhension du projet de manière plus simple et efficace. Cette approche a également permis de recueillir rapidement leurs retours et d'ajuster nos idées en fonction de leurs attentes, ce qui a renforcé la pertinence de notre travail dès les premières étapes.



*Figure 6 : Premier schéma de cas d'utilisation de l'explore game*

Pour la conception du Modèle Conceptuel de Données (MCD), nous avons mis un accent particulier sur la modularité, afin de permettre une flexibilité maximale dans l'application. L'objectif était de séparer les différentes données en composants distincts, chaque composant étant un élément modifiable indépendamment des autres. Cela permet de mieux répondre aux besoins des utilisateurs tout en offrant une grande souplesse pour les évolutions futures du projet.

Nous avons donc veillé à diviser les informations en tables distinctes, chacune représentant un élément modulaire de l'application. Par exemple, les départements, les utilisateurs, et les

scénarios ont été traités comme des entités indépendantes, ce qui permet au client de facilement ajouter, modifier ou supprimer des éléments sans perturber l'ensemble du système. Cette structure facilite non seulement la gestion des données mais permet également une meilleure personnalisation selon les besoins spécifiques des utilisateurs.

Cette approche a également été pensée pour assurer une évolution fluide de l'application à mesure que de nouvelles fonctionnalités ou de nouveaux scénarios pourraient être ajoutés. En séparant les données de manière aussi précise, nous avons pu créer une base de données flexible, facile à maintenir et à adapter, tout en respectant les bonnes pratiques en matière de conception de bases de données. Ainsi, cette modularité garantit que l'application pourra évoluer en même temps que les besoins du client, tout en restant performante et cohérente à long terme.

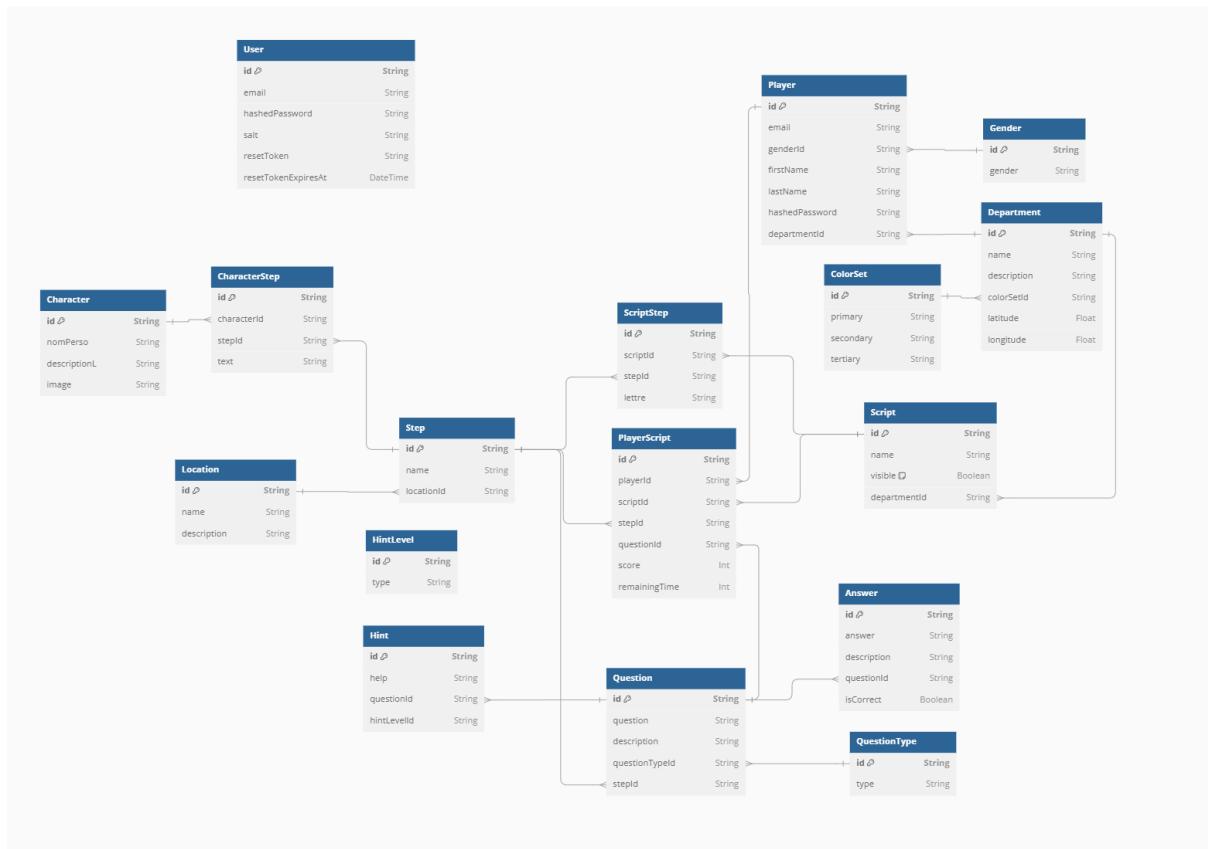


Figure 7 : Schéma de modèle conceptuel de données

Ensuite, nous avons organisé un rendez-vous avec Mme INGARAO afin de lui présenter notre première analyse du projet et les contraintes identifiées, tout en évitant d'entrer dans des détails techniques trop complexes. Lors de cette réunion, nous avons discuté de ses attentes précises ainsi que des enjeux fonctionnels de l'application. Cela nous a permis de mieux cerner les besoins spécifiques et de valider l'orientation générale du projet.

Une fois les grandes lignes confirmées par Mme INGARAO et quelques modifications sur nos schéma suite à cela, nous avons commencé à travailler sur les maquettes de l'application ainsi que sur une charte graphique adaptée.

Pour les maquettes nous avons dû relever divers défis pour respecter les différents enjeux pour l'expérience utilisateur et respecter la charte graphique actuelle de l'IUT d'Orléans.

Pour cela, nous avons choisi d'utiliser la couleur officielle de l'IUT d'Orléans comme base pour définir la charte graphique du projet. Cette décision a permis d'assurer une certaine cohérence visuelle tout en renforçant l'identité de l'IUT. Cependant, afin d'éviter que le projet ne soit visuellement monotone ou que toutes les pages aient exactement le même aspect, nous avons travaillé sur des variantes de cette couleur en jouant sur les nuances et l'opacité. Cela nous a permis de construire une palette de couleurs harmonieuse et polyvalente, parfaitement adaptée à la conception de nos maquettes.



*Figure 8 : Palette de couleur de l'Explore Game*

Une de nos idées principales était de rendre chaque département facilement identifiable dans l'"Explore Game". Pour cela, nous avons mis en place un système où, lorsqu'un utilisateur se trouve sur une page liée à un département spécifique, la page ainsi que les éléments qui le concernent s'adaptent à la couleur dédiée à ce département. Par exemple, chaque département se voit attribuer une couleur unique issue de notre palette, ce qui permet de créer une association visuelle intuitive pour les utilisateurs.



*Figure 9 : Palette de couleur du département Informatique de l'Explore Game*

Ce choix de design renforce non seulement l'identité visuelle du projet, mais améliore aussi l'expérience utilisateur en rendant la navigation plus claire et plus immersive. Cela donne un rendu dynamique et évite la monotonie tout en répondant à l'un des objectifs principaux : rendre l'interface ergonomique et facilement compréhensible.

Pour réaliser nos maquettes, nous avons utilisé l'outil Figma, une plateforme de conception collaborative qui nous a permis de créer des interfaces interactives et visuelles. L'objectif principal était de concevoir des interfaces aussi intuitives que possible. Cela signifie que les utilisateurs devaient comprendre instinctivement la fonction de chaque élément, comme les boutons, avant même de cliquer dessus. Cette approche visait à offrir une expérience utilisateur fluide et agréable.

Pour vérifier si nous avions atteint cet objectif, nous avons demandé à certains de nos camarades de tester nos maquettes interactives. Ces tests utilisateurs nous ont fourni des

retours précieux, en mettant en lumière ce qui fonctionnait bien et les points qui nécessitaient des ajustements. Ces retours ont été essentiels pour affiner nos maquettes et améliorer leur clarté et leur ergonomie.

Par la suite, les maquettes finalisées ont été présentées à M. CARLES, chargé de communication à l'IUT d'Orléans, lors d'une réunion dédiée. Cette rencontre a joué un rôle crucial dans l'avancée de notre projet. Elle nous a permis de discuter des contraintes et des exigences liées au respect des normes graphiques de l'IUT, telles que l'utilisation des couleurs, des polices, et des éléments visuels spécifiques. Pendant cet échange, M. CARLES a validé nos maquettes interactives, tout en nous donnant des conseils pertinents pour les améliorer davantage.

Cette collaboration nous a non seulement aidés à perfectionner nos maquettes, mais elle a également renforcé la cohérence visuelle de notre projet avec l'identité institutionnelle de l'IUT d'Orléans. Cela a été une étape décisive pour garantir la qualité et la pertinence de nos interfaces.

## 6. ARCHITECTURE TECHNIQUE

Notre application repose sur une séparation claire des responsabilités entre les microservices [Core](#) et [Client](#).

L'enjeu principal était de concevoir une interface utilisateur viable, facilitant la gestion des différentes composantes et données relatives au bon déroulement de l'Explore Game pour les équipes de gestion du projet. Pour répondre à cet enjeu de modularité des données – notamment la nécessité de permettre une personnalisation complète des scénarios, des étapes, des questions, des réponses, et des indices associés – nous avons choisi de distinguer deux microservices.

Le microservice [Client](#) est dédié à l'application utilisée par les usagers, en tant que consommateur des données, tandis que le micro service [Core](#) est chargé de servir ces données et d'inclure des interfaces utilisateurs spécifiques. Cette architecture nous permet de répondre efficacement à cet enjeu majeur tout en garantissant une structure modulaire, évolutive et facile à maintenir.

## Core

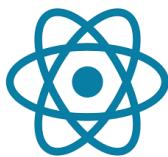
Le microservice Core a été développé avec [RedwoodJS](#), un framework JavaScript moderne conçu pour les applications web full-stack. Nous avons choisi ce framework précisément parce qu'il s'appuie sur [Node.js](#), une technologie que notre équipe de développement maîtrise pleinement, permettant ainsi une prise en main rapide et une productivité accrue.



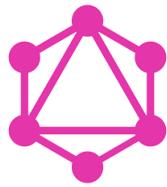
*Logo du framework JavaScript RedwoodJS*

**RedwoodJS** intègre nativement [GraphQL](#) et simplifie le développement en combinant :

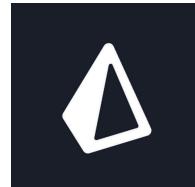
- un frontend basé sur [React](#),
- un backend utilisant une [API GraphQL](#),
- une base de données gérée avec l'ORM intégré [Prisma](#).



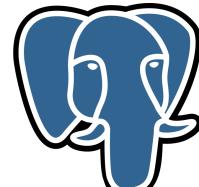
*React (UI)*



*API GraphQL*



*Prisma (ORM)*



*PostgreSQL (BDD)*

Le microservice Core prend en charge les opérations CRUD ainsi que la logique métier. Il interagit de manière efficace avec une base de données [PostgreSQL](#) pour les données relationnelles et un bucket [MinIO](#) pour la gestion des images, garantissant ainsi une gestion fiable des données.

L'utilisation de [GraphQL](#) optimise les échanges en offrant flexibilité et performances. Contrairement aux API REST, GraphQL permet de spécifier uniquement les champs et attributs nécessaires dans les requêtes, réduisant ainsi les charges inutiles.

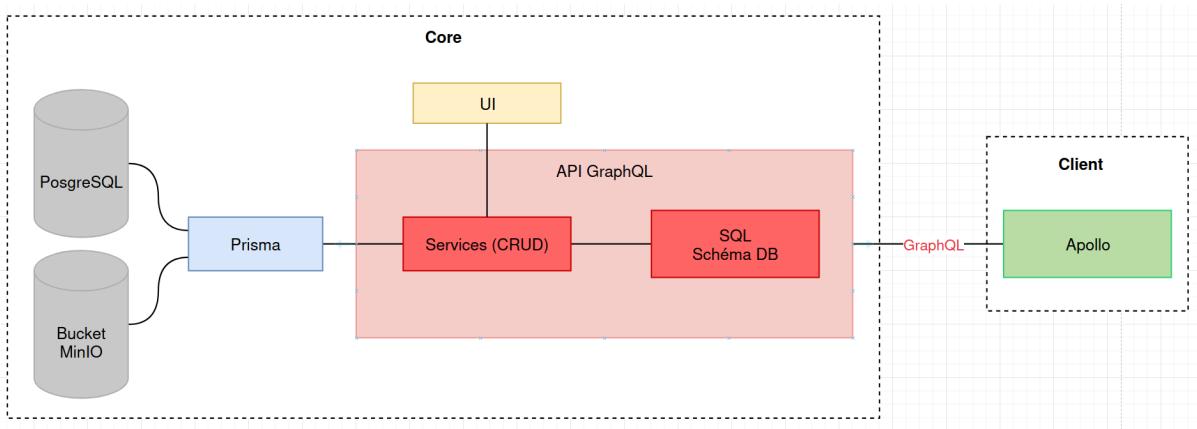


Figure 10 : Architecture des micro services (core & client)

Grâce à cette architecture et à ce framework, nous avons pu concevoir rapidement un microservice ainsi que son API dédiée, intégrée à notre client. En nous basant sur le MCD précédemment défini, nous avons décrit le schéma de la base de données et réalisé nos premières migrations.

```
model Question {
  id          String      @id @default(uuid())
  question    String
  description String
  questionTypeId String
  stepId      String
  Step         Step        @relation(fields: [stepId], references: [id])
  QuestionType QuestionType @relation(fields: [questionTypeId], references: [id])
  Answer      Answer []
  Hint         Hint []
  PlayerScript PlayerScript []
}
```

Figure 11 : Extrait du schéma (définition du modèle Question)

RedwoodJS génère automatiquement les services associés à chaque modèle défini dans le schéma, notamment les fonctions CRUD. De plus, étant développé en TypeScript, il crée également les types et interfaces correspondants pour les champs d'entrée des fonctions CRUD.

```

export const question: QueryResolvers['question'] = ({ id }) => {
  return db.question.findUnique({
    where: { id },
  })
}

export const createQuestion: MutationResolvers['createQuestion'] = ({ input, }) => [
  return db.question.create({
    data: input,
  })
]

export const updateQuestion: MutationResolvers['updateQuestion'] = ({ id, input, }) => {
  return db.question.update({
    data: input,
    where: { id },
  })
}

export const deleteQuestion: MutationResolvers['deleteQuestion'] = ({ id }) => {
  return db.question.delete({
    where: { id },
  })
}

```

*Figure 13 : Exemple du service CRUD du modèle Question*

De plus, RedwoodJS produit des routes conformes aux principes d'une API RESTful, accompagnées de formulaires standardisés permettant d'exécuter les opérations CRUD via des interfaces utilisateur.

```

<Set wrap={ScaffoldLayout} title="Questions" titleTo="questions" buttonLabel="New Question" buttonTo="newQuestion">
  <Route path="/questions/new" page={QuestionNewQuestionPage} name="newQuestion" />
  <Route path="/questions/{id}/edit" page={QuestionEditQuestionPage} name="editQuestion" />
  <Route path="/questions/{id}" page={QuestionQuestionPage} name="question" />
  <Route path="/questions" page={QuestionQuestionsPage} name="questions" />
</Set>

```

*Figure 14 : Extrait des routes créée automatiquement*

The screenshot shows a light gray modal window titled "Edit Question 1". At the top right is a green button labeled "+ NEW QUESTION". Inside the modal, there are four input fields: "Question" containing "Quel est votre nom?", "Description" containing "Entrez votre nom", "Question type id" containing "1", and "Step id" containing "1". At the bottom right of the modal is a blue "SAVE" button.



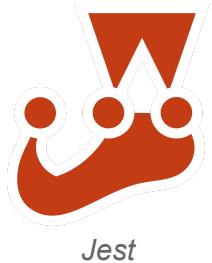
Figure 19 : Formulaire créée par défaut

The screenshot shows a dark-themed modal window titled "Question". At the top right is a green button labeled "+ NEW QUESTION". Inside the modal, there are three sections: "Intitulé" with "Quel est votre nom?", "Description" with "Entrez votre nom", and "Type de la question" with a dropdown menu showing "Texte" (which is selected) and other options like "Choix" and "Date". Below these is a section titled "Réponses associées" with a "Nouvelle réponse" button and an "Ajouter" button. A message "Pas de réponse associées" is displayed. At the bottom is a section titled "Indices associées".

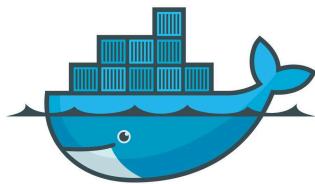
Figure 20 : Formulaire personnalisé

Par ailleurs, RedwoodJS inclut nativement des tests unitaires grâce à [Jest](#), ce qui permet de s'assurer de la stabilité et de la qualité du code à chaque étape du développement. Il intègre également [Storybook](#), un outil puissant pour le développement d'interfaces utilisateur, permettant de visualiser et tester facilement les composants indépendamment de

l'application. Cette combinaison facilite la création de tests robustes et la conception d'interfaces dynamiques, garantissant une meilleure expérience de développement et une application plus stable.



Enfin, RedwoodJS facilite grandement la mise en place de [Docker](#) en intégrant nativement les ressources et les allocations nécessaires pour se dockeriser automatiquement. Il prend en charge tous les aspects de l'architecture, y compris l'API, le frontend, la base de données et la console associée, permettant ainsi de déployer l'application et ses dépendances de manière simplifiée et cohérente dans un environnement conteneurisé.



Ainsi, RedwoodJS correspond parfaitement à nos besoins, car il inclut de nombreuses fonctionnalités natives, telles que la gestion des routes, le support GraphQL, et Prisma pour les bases de données. Ces outils intégrés nous ont permis de gagner un temps précieux dans le développement en limitant le recours à des bibliothèques tierces ou à des configurations supplémentaires.

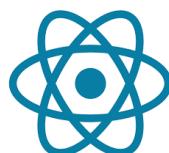
## Client

Le second enjeu était de judicieusement choisir nos technologies web pour l'application Explore Game en elle-même, utilisée par les futurs collégiens, lycéens, étudiants et échanges académiques. En effet, ces choix ont été longuement discutés, afin de rendre la future application accessible pour tous les appareils, et tous les usagers. Ainsi, le choix d'une web-app s'est fait afin de maximiser la compatibilité des appareils (iOS et Android), et ainsi rendre l'application accessible via le web.

Nous avons conçu l'application Client en adoptant une architecture **Node.js** et en utilisant **React**, grâce au serveur de développement **Vite**. Ce choix technologique simplifie grandement les gestions des dépendances et les phases de développement. Grâce à notre maîtrise de ces outils acquise au fil des années, nous avons pu accroître notre productivité et accélérer le développement des fonctionnalités de l'application.



*Node.js (moteur d'environnement javascript)*



*React (UI)*



*Vite (serveur de développement)*

Afin de gagner du temps de développement, nous avons adopté **TailwindCSS**, un framework CSS open-source qui permet de définir des règles de style directement dans les éléments du DOM, comme Bootstrap. Associé à la bibliothèque de composants préfabriqués **shadcn/ui**, largement utilisée pour la conception de formulaires et d'interfaces, nous avons pu concevoir une interface épurée et fonctionnelle. Enfin, nous avons intégré la bibliothèque d'animation moderne **Framer Motion**, afin de rendre les interfaces fluides pour les utilisateurs de l'application (étudiants, collégiens, lycéens, étudiants en échanges académiques).



Tailwindcss



shadcn/ui



Framer Motion

Pour se connecter à l'API fournie par le service Core, le client utilise un client [Apollo GraphQL](#), qui lui permet de consommer les données. Ce client peut également être étendu, en y greffant un service de cache intégré, afin de sauvegarder et minimiser les requêtes au serveur distant, en utilisant la mémoire du téléphone.



Apollo GraphQL

```
const client = new ApolloClient({
  uri: import.meta.env.VITE_API_URL_GRAPHQL,
  cache: new InMemoryCache()
})

createRoot(document.getElementById('root')!).render(
  <ApolloProvider client={client}>
    <App />
  </ApolloProvider>
)
```

*Figure 15 : Définition du client Apollo provider*

La définition de ce client permet d'exécuter des requêtes personnalisées à l'API GraphQL, en interagissant directement avec la base de données distante selon les besoins spécifiques.

```
export const SCENARIO = gql`  
query FindScenarioById($id: String!) {  
  Execute Query  
  script(id: $id) {  
    id  
    name  
    ScriptStep {  
      stepId  
      Step {  
        id  
        Questions {  
          id  
        }  
      }  
      PlayerScript {  
        id  
        playerId  
        stepId  
        questionId  
        completed  
      }  
    }  
  }  
};`
```

*Figure 16 : Exemple de requête GraphQL*

Cette requête permet de récupérer certains champs de la table *Script* (scénario), tout en offrant la possibilité d'imbriquer d'autres tables dans les requêtes. Par exemple, l'objet *PlayerScript* représente la progression d'un participant au sein du scénario.

Enfin, le troisième enjeu du projet concernait l'architecture logicielle de notre implémentation. En effet, étant donné que notre projet est divisé en deux microservices distincts, qui interagissent via une API, nous devons prendre en compte certaines contraintes techniques et de réseau lorsque les requêtes simultanées augmentent. Il faut prendre en considération que cette application sera utilisée lors de journées d'intégration pour les étudiants en BUT1 à l'IUT d'Orléans, avec potentiellement plus d'une centaine de connexions simultanées, chacune envoyant plusieurs requêtes au microservice Core. Nous avons donc opté pour la minimisation des requêtes au serveur en évitant de les multiplier inutilement.

Par exemple, au lancement d'un scénario, le micro service client récupérera toutes les questions et leurs attributs associés une seule fois, avant de les stocker dans la base de données locale du téléphone (localStorage). Les requêtes au serveur ne seront réalisées uniquement lors d'actions spécifiques, telles que :

- Vérifier une réponse
- Sauvegarder la progression en cas de sortie du scénario.

Cette implémentation permet de réduire considérablement le nombre de requêtes au serveur, en privilégiant la mémoire locale du téléphone pour le suivi des scénarios.

Ainsi, le développement d'une interface utilisateur agréable et fluide a été un enjeu clé dans la conception de l'application client, visant à améliorer l'expérience utilisateur dans un contexte où le site est utilisé par un grand nombre de personnes. Par ailleurs, la performance et la minimisation des requêtes vers le serveur distant (Core) sont rapidement devenues des préoccupations essentielles, que nous avons su approchées et solutionnées par l'utilisation de stockage local intermédiaire.

## Type-Sharing

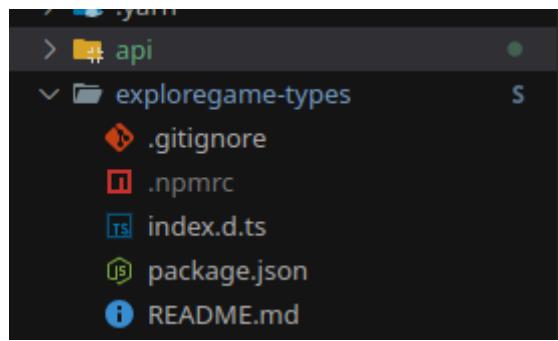
Nos deux microservices utilisent principalement le langage TypeScript, qui se distingue de JavaScript par son système de typage strict. Ce choix a été fait dès le début du développement afin d'assurer une qualité de code rigoureuse et claire, en s'appuyant sur des types spécifiques et réutilisables. Dans cette optique, nous avons décidé de réutiliser les types générés automatiquement par notre microservice **Core** au sein de notre micro service **Client**.

Ces types sont générés automatiquement à partir des attributs et des types définis dans le schéma de base de données Prisma.

```
export type Question = {
    __typename?: 'Question';
    Answer: Array<Maybe<Answer>>;
    Hint: Array<Maybe<Hint>>;
    PlayerScript: Array<Maybe<PlayerScript>>;
    QuestionType: QuestionType;
    Step: Step;
    description: Scalars['String'];
    id: Scalars['String'];
    question: Scalars['String'];
    questionTypeId: Scalars['String'];
    stepId: Scalars['String'];
};
```

*Figure 12 : Exemple de type généré automatiquement dans le micro service Core*

Pour éviter de réécrire manuellement les types dans notre micro service **Client** et automatiser leur partage entre différents services, nous avons choisi d'implémenter un submodule GitHub (un dépôt GitHub intégré à un dépôt existant) basé sur notre microservice **Core**.



*Figure 17 : Submodule de types-sharing*

Ce submodule est un projet npm indépendant dont le point d'entrée est un fichier `index.d.ts`, qui centralise tous les types définis dans le schéma de base de données. Une commande dédiée, `yarn run push:types`, permet de mettre à jour le submodule et le fichier de types dans le micro service **Core**. Une fois cette étape réalisée, nous publions le package sur le registre officiel npm.

*Registre officiel npm : <https://www.npmjs.com/>*

**@explore-game/types** ts

1.1.1 • Public • Published 4 days ago

**Explore Game - Types**

Librairie de **types-sharing** pour le projet *Explore Game* de l'IUT d'Orléans.

Ce paquet permet de partager les types entre les différents microservices du projet.

**Microservices**

- **core** : Microservice principal, définit le schéma de la base de donnée, distribue l'API et gère les requêtes.  
*node.js - redwoodjs - react - graphql - prisma - apollo*
- **client** : Microservice de l'interface des clients, connectés via l'API du microservice core. Cette interface permet d'utiliser l'application *Explore Game* en tant qu'étudiant / lycéen / collégien.  
*node.js - vite - react - apollo*

**Authors**

4 étudiants de BUT Informatique de l'IUT d'Orléans :

- Jordan LAVUANT

**Install**

```
> npm i @explore-game/types
```

**Weekly Downloads**

169

Version	License
1.1.1	All rights reserved

Unpacked Size Total Files  
90 kB 3

Last publish  
4 days ago

Collaborators

*Figure 18 : Paquet npm de types-sharing*

*(<https://www.npmjs.com/package/@explore-game/types>)*

Ensuite, ce package peut être installé dans notre micro service **Client** ou dans tout autre microservice futur nécessitant l'intégration de ces types, par exemple pour un nouveau fournisseur tiers (Third Party Provider). Cette approche facilite la mise à jour des types à

travers nos microservices en publant simplement une nouvelle version (majeure, mineure ou patch) du package npm. Les consommateurs des types, comme le **Client**, peuvent alors mettre à jour leurs dépendances. Cette méthode est rapide, simple et efficace.

## 7. ETAT DU PROJET

À ce jour, le projet Explore Game a bien progressé dans sa phase de digitalisation. L'application a été structurée avec un ensemble d'interfaces et de fonctionnalités clés qui visent à améliorer l'expérience utilisateur tout en respectant l'objectif de découverte immersive de l'IUT d'Orléans. Nous avons mis en place les bases nécessaires à une version numérique qui permettra aux participants d'explorer l'IUT de manière interactive, en répondant à des questions et en découvrant des éléments du campus, tout en suivant les étapes du scénario proposé.

L'interface utilisateur a été définie de manière à faciliter la navigation, avec des pages permettant aux utilisateurs de s'identifier, de suivre leurs progrès, de découvrir les filières de l'IUT, et d'interagir avec la carte interactive pour se déplacer à travers les différents lieux du campus. De plus, une logique de leaderboard et un système de gestion des scores ont été intégrés, afin de renforcer l'aspect compétitif et ludique de l'expérience.

Concernant les fonctionnalités du jeu, plusieurs éléments ont été définis pour assurer une progression fluide dans les scénarios. Les utilisateurs peuvent démarrer, reprendre ou clôturer des scénarios, répondre aux questions associées aux différents lieux, utiliser des indices, et découvrir des lettres dissimulées pour résoudre des énigmes. Le système prend également en compte des pénalités liées au temps, avec un minuteur qui ajoute un défi supplémentaire pour les participants.

L'intégration de ces fonctionnalités interactives et l'objectif de rendre l'expérience plus immersive et accessible via une plateforme numérique sont en bonne voie, dans le respect de la vision du projet : digitaliser l'existant pour offrir une expérience enrichie, dynamique et ludique. Il reste toutefois à finaliser certaines parties de la mise en œuvre technique afin de rendre un livrable utilisable, notamment pour les critères de la réussite, et il sera aussi

important de tester l'application en amont afin de ne pas découvrir des problèmes de dernières minutes.

## Fonctionnalités et interfaces réalisées

### Interfaces Utilisateur

- **Identification / Inscription** : Permet aux utilisateurs de créer un compte ou de se connecter à l'application.
- **Page Profil** : Affiche les informations personnelles de l'utilisateur, telles que le score et l'avancement dans le jeu.
- **Page Avancements** : Affiche l'état d'avancement dans les différents scénarios, avec un suivi détaillé des étapes terminées et restantes.
- **Page d'Accueil** : Point de départ de l'application, offrant l'accès aux scénarios et à d'autres fonctionnalités importantes.
- **Page de département**: Présente les différentes formations proposées par l'IUT d'Orléans que les participants peuvent découvrir pendant le jeu.
- **Carte** : Carte interactive de l'IUT, permettant aux utilisateurs de visualiser et de se déplacer dans les différents bâtiments et lieux associés aux étapes du jeu.
- **Page des Questions** : Affiche les questions à répondre en fonction des lieux explorés, permettant la progression dans le scénario.
- **Page de Découverte de Lettres entre les Étapes** : Permet de découvrir des lettres cachées entre les étapes pour résoudre des énigmes.
- **Leaderboard** : Affiche le classement des participants en fonction de leurs scores et performances.

### Fonctionnalités

- **Démarrer / Clôturer / Mettre en pause/ Reprendre un scénario** : Permet aux utilisateurs de commencer un scénario, de le clôturer une fois terminé ou de le reprendre après l'avoir interrompu.

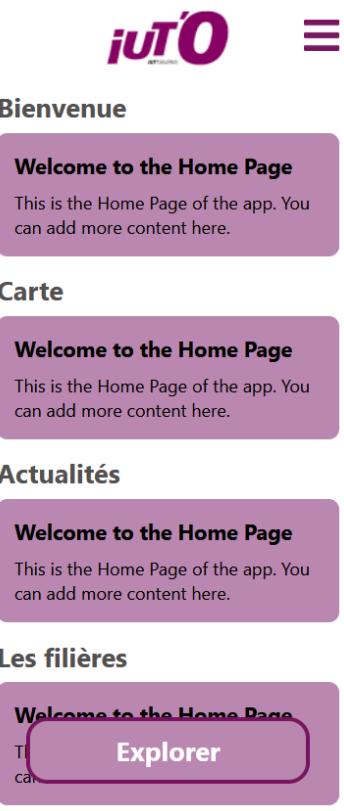
- **Répondre aux questions** : Permet aux utilisateurs de répondre aux questions posées à chaque étape du jeu. Seulement les questions ouvertes et les questions à choix unique sont fonctionnelles.
- **Utiliser les indices de question** : Les utilisateurs peuvent utiliser des indices avec un système de niveau permettant d'être plus ou moins aidé. Cela a une répercussion sur le temps et le score final.
- **Progresser dans le scénario** : Les utilisateurs avancent dans le jeu en fonction de leurs réponses et de l'exploration des lieux.
- **Découvrir des lettres** : Les lettres cachées à travers les étapes sont découvertes pour résoudre des énigmes.
- **Jeu de données initiales** : Un jeu de données nécessaires à l'exécution du jeu basé sur l'Explore Game papier de cette année(scénarios, étapes, questions, etc.).
- **Pénalité score / minuteur** : Le système intègre un minuteur, et les utilisateurs sont pénalisés en fonction du temps qu'ils mettent à répondre aux questions.
- **Logique du leaderboard** : Calcul et affichage du classement des joueurs en fonction des scores et du temps écoulé.

#### Fonctionnalités interactives

- **Carte interactive** : La carte dynamique permet une exploration du campus et de ses différents lieux associés aux étapes du jeu.
- **Dialogue interactif** : Fonction permettant aux utilisateurs d'interagir avec des personnages clés du jeu, tels que des enseignants ou des étudiants, pour obtenir des indices et des informations.

# Images des interfaces

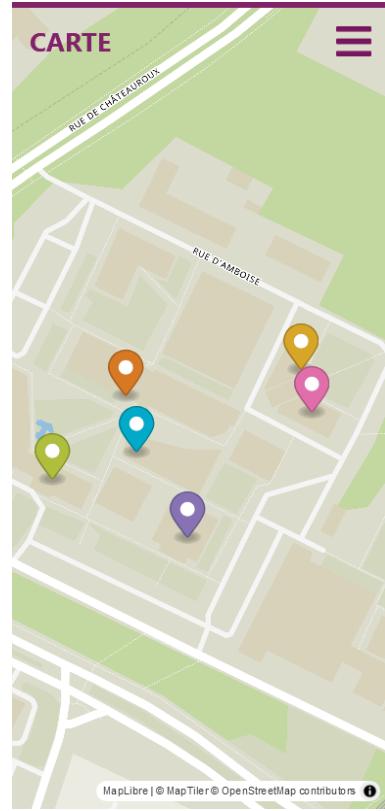
## Client



[Figure 21 : Page d'accueil](#)



[Figure 22 : Menu Burger](#)



[Figure 23 : Carte interactive](#)



## Bienvenue

### Welcome to the Home Page

This is the Home Page of the app. You can add more content here.

## Carte

### Welcome to the Home Page

This is the Home Page of the app. You can add more content here.

## Actualités



## Bienvenue

### Welcome to the Home Page

This is the Home Page of the app. You can add more content here.

## Carte

### Welcome to the Home Page

This is the Home Page of the app. You can add more content here.

## Actualités



## Bienvenue

### Welcome to the Home Page

This is the Home Page of the app. You can add more content here.

## Carte

### Welcome to the Home Page

This is the Home Page of the app. You can add more content here.

## Actualités

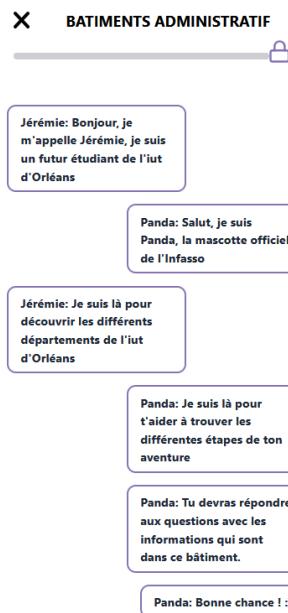


Figure 27 : Page de dialogue



Figure 28 : Page de question

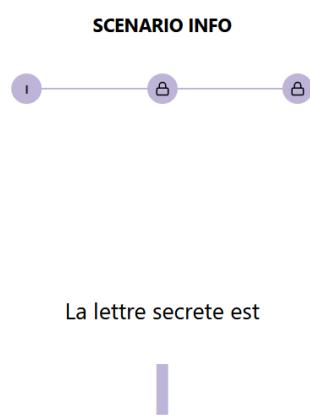
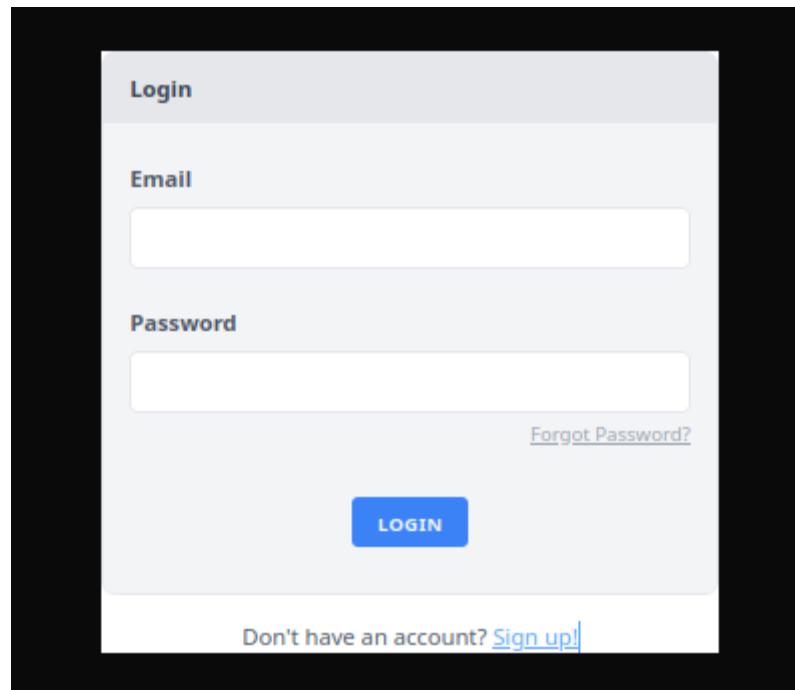


Figure 29 : Découverte de lettre

## Core

### Interfaces par-défaut

Ces interfaces, bien que fonctionnelles par défaut, restent rudimentaires et ne respectent pas les normes d'expérience utilisateur établies.



*Figure 30 : Interface provisoire d'authentification*

Departments						<a href="#">+ NEW DEPARTMENT</a>
Id	Name	Description	Color set id	Latitude	Longitude	
1	INFO	BUT Informatique	1	47.8433084	1.9267556	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
2	GEA	BUT Gestion des Entreprises et des Administrations	2	47.8437317	1.9263861	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
3	GMP	BUT Génie Mécanique et Productique	3	47.8440098	1.9263063	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
4	QLIO	BUT Qualité, Logistique Industrielle et Organisation	4	47.8441395	1.927583	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
5	Chimie	BUT Chimie	6	47.8439269	1.927661	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
6	MT2E	BUT Métiers de la transition et de l'efficacité énergétiques	5	47.8435985	1.9257699	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>

*Figure 31 : Interface provisoire de listing des occurrences d'une entité (ici département)*

Departments		<a href="#">+ NEW DEPARTMENT</a>
<b>Department 1 Detail</b>		
	<b>Id</b>	1
	<b>Name</b>	INFO
	<b>Description</b>	BUT Informatique
	<b>Color set id</b>	1
	<b>Latitude</b>	47.8433084
	<b>Longitude</b>	1.9267556
<a href="#">EDIT</a> <a href="#">DELETE</a>		

*Figure 32 : Interface provisoire du détail d'une occurrence (ici département)*

**Departments**

+ NEW DEPARTMENT

Edit Department 1

Name	INFO
Description	BUT Informatique
Color set id	1
Latitude	47.8433084
Longitude	1.9267556

**SAVE**



Figure 33 : Figure Interface provisoire d'édition d'une occurrence (ici département)

**Departments**

+ NEW DEPARTMENT

ID	Name	Description	Color set id	Latitude	Longitude	SHOW	EDIT	DELETE
1	INFO	BUT Informatique	1	47.8433084	1.9267556	SHOW	EDIT	DELETE
2	GEA	BUT Gestion des Entreprises et des Administrations	2	47.8437317	1.9263861	SHOW	EDIT	DELETE
3	GMP	BUT Génie Mécanique et Productique	3	47.8440098	1.9263063	SHOW	EDIT	DELETE
4	QLIO	BUT Qualité, Logistique Industrielle et Organisation	4	47.8441395	1.927583	SHOW	EDIT	DELETE
5	Chimie	BUT Chimie	6	47.8439269	1.927661	SHOW	EDIT	DELETE
6	MT2E	BUT Métiers de la transition et de l'efficacité	85	1.9257699		SHOW	EDIT	DELETE

localhost:8910

Are you sure you want to delete department 1?

Cancel OK

Figure 34 : Exemple de suppression d'une occurrence (ici département)

## Interfaces modernisées

Nous avons engagé à la date de ce document une phase de modernisation de ces interfaces afin de les rendre intuitives, simples et rapides à utiliser. Vous trouverez ci-dessous un exemple :

Questions

+ NEW QUESTION

**Question**

Intitulé  
Dans quel bâtiment se trouve la cafétéria ?

Description

Type de la question Choix

Etape associée Step 1

**Réponses associées**

Nouvelle réponse Ajouter

Description

Bâtiment CAP

Bâtiment GEO

Bâtiment A

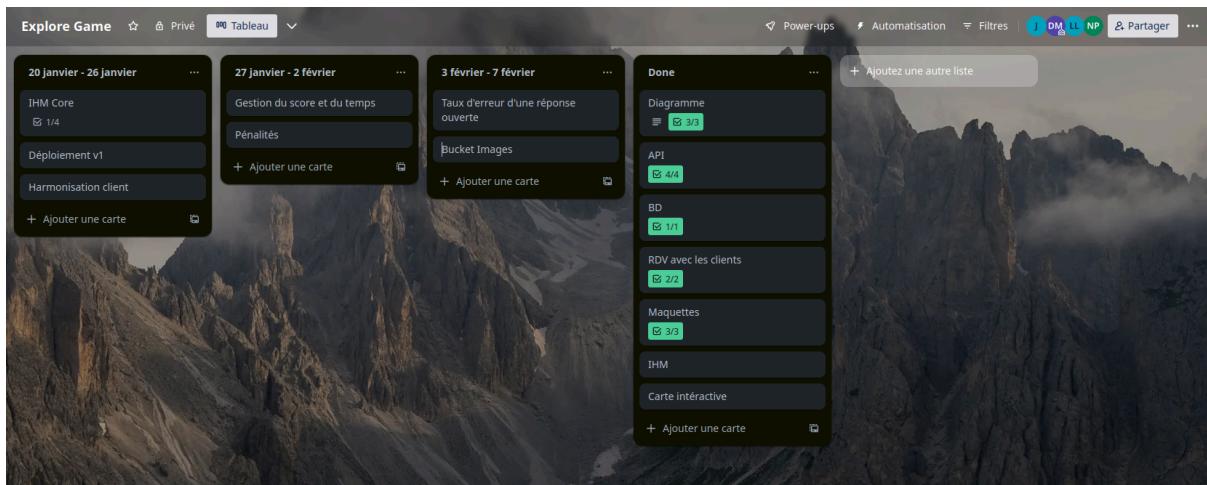
Bâtiment B

**Indices associés**

Annuler Sauvegarder

*Figure 35 : Interface provisoire d'édition d'une occurrence (ici question)*

## Organisation



*Figure 4 : Version de notre Trello (16 janvier 2025)*

## 8. AMÉLIORATIONS À VENIR

### Déploiement V1 sur les serveurs de l'IUT

En collaboration avec M. Louis NOËL, nous avons convenu de déployer l'application sur une machine virtuelle hébergée par les serveurs de l'IUT d'Orléans. Une première version sera mise en place pour tester son fonctionnement, suivie de déploiements des futures mises à jour.

### Amélioration des IHM du Core

Les interfaces de listing, de détail, de modification et de création pour chaque occurrence d'entités seront modernisées et réorganisées afin d'être plus intuitives, claires et pratiques pour les gestionnaires et administrateurs du projet *Explore Game*. Ces interfaces répondront à l'objectif principal de cette application Core : offrir une personnalisation rapide, claire et précise des données nécessaires au déroulement des scénarios d'*Explore Game*, ainsi que des contenus textuels associés.

Cette modernisation des interfaces est actuellement en cours de développement à la date de rédaction de ce document.

## Harmonisation du micro service client

Nous allons refaire une passe complète sur les composants et pages du services client afin que tout soit parfait, que ca soit pour l'interface utilisateur, l'expérience utilisateur, ou encore la modularité du projet.

## Question choix multiples

L'intégration de questions à choix multiples est actuellement en développement. L'affichage de ces questions sera similaire à celui des questions à choix unique, mais un label précisera que plusieurs réponses sont attendues

## Intégrer la fonctionnalité du timer et des pénalités associées

Le minuteur est un élément clé de la progression dans un scénario. Pour offrir un challenge supplémentaire, chaque scénario sera accompagné d'un minuteur d'une heure. Cela encourage les utilisateurs à terminer le scénario dans le temps imparti tout en ajoutant une dimension compétitive. En effet, les performances des étudiants seront classées en fonction du temps de résolution dans leur département.

Les pénalités, quant à elles, s'appliquent lorsque vous utilisez un indice. Le niveau de pénalité sera proportionnel à l'indice consommé et se traduira par une réduction du temps restant sur le minuteur (quelques secondes ou minutes en moins).

## Retirer les questions ouvertes

En effet les questions ouvertes posaient trop de problèmes au niveau des différentes réponses possibles de l'utilisateur, ainsi que le fait de pouvoir donner une réponse correcte à un utilisateur qui avait donné la bonne réponse mais pas avec l'orthographe qui convient.

## Bucket d'images pour enrichir l'expérience

Nous ajouterons la possibilité d'afficher des images importées via l'application de gestion de l'Explore Game (Core). Ces images pourront illustrer des questions ou réponses, offrant ainsi une expérience utilisateur plus immersive.

## Page des avancements des scénarios

Cette page a pour mission de répertorier et d'afficher vos progressions sur les différents scénarios disponibles. Vous pourrez également reprendre un scénario en cours si vous ne l'avez pas terminé.

## Page BDE : Actualités et événements

La page « BDE », accessible depuis le menu Burger, sera dédiée aux actualités des BDE des départements de l'IUT d'Orléans. Elle fournira des informations sur les événements, soirées et activités organisés au sein des formations. Les parties prenantes pourront l'alimenter via l'interface Core.

## Édition du profil : Simplification des données personnelles

La page d'édition de profil permettra aux utilisateurs de modifier leur mot de passe, bien que cette fonctionnalité ne soit pas encore disponible. Par souci de conformité au RGPD, toutes les données personnelles seront supprimées. Les champs de nom, prénom, sexe et adresse email seront remplacés par un nom d'utilisateur et un mot de passe uniquement.

## Trophés et succès

La page des trophées et succès sera consacrée à l'affichage des scénarios terminés et des autres réussites obtenues. Par exemple :

- Terminer un scénario en moins de 20 minutes.
- Obtenir 100 % de bonnes réponses sur un scénario.
- etc...

Cette page permettra aux utilisateurs de suivre leurs accomplissements et de se fixer de nouveaux objectifs.

## 9. CONCLUSION

Le projet Explore Game a été une expérience enrichissante, à la fois techniquement et humainement. Il nous a offert l'opportunité de développer nos compétences professionnelles dans un cadre réel avec un client, tout en appliquant des notions théoriques académiques sur l'organisation d'équipe, la qualité du développement et les méthodologies avancées de conception logicielle.

Ce projet a également mis en lumière des enjeux techniques majeurs, notamment lors de la réflexion préliminaire sur l'architecture logicielle. Il a été essentiel de concevoir une structure modulable pour anticiper la personnalisation des données du jeu (questions, réponses, indices, scénarios, étapes, lieux, etc...). Parallèlement, la conception d'interfaces utilisateur viables a renforcé notre compréhension des enjeux d'une expérience utilisateur optimale.

Points positifs :

- Renforcement de nos compétences techniques grâce à des technologies modernes telles que **RedwoodJS**, **Prisma**, **Apollo GraphQL** et **Docker**.
- Expérience concrète en gestion de projet et coordination d'équipe, en complément de nos expériences en alternance.
- Réflexion approfondie sur l'expérience utilisateur et la conception d'un outil pratique et intuitif.

Cependant, des axes d'amélioration ont été identifiés :

- Un manque de communication et de mise à jour régulière envers le client concernant l'avancement du projet.
- Une fréquence insuffisante des réunions pour ajuster les priorités et renforcer la cohésion de l'équipe.

En conclusion, le projet Explore Game a été une étape clé de notre parcours. Il nous a permis de développer des compétences transversales et techniques tout en nous confrontant à des problématiques réelles. Ces apprentissages constituent une base solide pour aborder avec confiance de futurs projets professionnels.

## 10. GLOSSAIRE

API : Application Programming Interface, ou interface logicielle, permet de connecter un logiciel ou un service à un autre, afin d'échanger des données et des fonctionnalités.

API REST : Une API REST (Representational State Transfer) est un style d'architecture logicielle qui définit un ensemble de contraintes pour la création de services web. Les API REST utilisent les principes fondamentaux du protocole HTTP (Hypertext Transfer Protocol) pour permettre aux systèmes informatiques de communiquer et d'échanger des données de manière simple et efficace.

API RESTful : Interface de programmation qui suit les principes REST (Representational State Transfer) pour permettre des échanges de données entre systèmes via HTTP, en utilisant des méthodes standard comme GET, POST, PUT et DELETE.

Back-End : Le terme back-end désigne la partie non visible d'une application informatique ou d'un site web, qui est responsable du traitement des données, de la logique métier, et de l'interaction avec la base de données.

BUT : (Brevet de Technicien Supérieur Universitaire) Diplôme national de l'enseignement supérieur français qui se prépare en trois ans dans un Institut Universitaire de Technologie (IUT).

Client : Micro service désignant l'application utilisée par les usagers finaux, faisant les Explore Game.

Conteneurisation (Docker) : Technologie qui permet de regrouper une application et ses dépendances dans un conteneur, facilitant ainsi le déploiement, la portabilité et la scalabilité des logiciels.

Core : Micro service désignant l'application de gestion et de personnalisation du projet Explore Game, destiné aux administrateurs et gestionnaires de l'outil.

CRUD : Acronyme pour Create, Read, Update, Delete. Ce sont les quatre opérations de base utilisées dans les bases de données et les systèmes de gestion d'informations pour manipuler les données.

CSS : Acronyme pour Cascading Style Sheets. Langage utilisé pour définir la mise en page et le style des documents HTML en ajoutant des couleurs, des polices, des marges, des paddings, etc.

DOM : (Document Object Model) Structure d'un document HTML représentée sous forme d'objets que l'on peut manipuler via du JavaScript, permettant l'interaction avec les éléments du document, leur modification, suppression, ou ajout.

Explore Game : Un jeu centré sur la découverte et l'exploration d'environnements, où le joueur peut interagir avec divers éléments, résoudre des énigmes, rencontrer des personnages, et découvrir des secrets ou des lieux inexplorés.

Framework : Cadre de développement facilitant le développement d'applications web et mobiles.

Front-End : Le terme front-end désigne la partie visible et interactive d'une application informatique ou d'un site web. Il représente l'interface utilisateur avec laquelle les utilisateurs interagissent directement.

Full-Stack : Un développeur qui maîtrise à la fois le développement front-end (interface utilisateur) et le back-end (logiciel, base de données, serveur). Il est capable de travailler sur toutes les couches d'une application web, du design jusqu'au déploiement.

IDE : Integrated Development Environment, ou environnement de développement intégré, est une interface graphique permettant d'éditer, compiler, déboguer et coder des programmes logiciels.

IHM : Acronyme pour Interface Homme-Machine. C'est l'interface qui permet la communication entre un utilisateur et un système informatique, généralement composée de composants graphiques tels que les boutons, les champs de saisie, les menus, etc.

MCD : Acronyme pour Modèle Conceptuel de Données. Représentation graphique qui permet de définir la structure des données d'une base en montrant les entités, leurs relations, les attributs, et les contraintes entre elles.

Mobile First : Concept Web Design qui consiste à concevoir une application web en se focalisant sur sa version mobile, pour ensuite l'adapter à de plus grands formats (ordinateur, tablette, etc...)

ORM : est un traducteur entre les langages hauts niveaux et le langage SQL d'une base de données, permettant de manipuler des données en base tel des objets hauts niveaux.

Responsive : est un terme utilisé en conception web pour décrire une approche de conception et de développement qui permet aux sites web et aux applications de s'adapter et de s'afficher correctement sur une variété de dispositifs et de tailles d'écran, tels que les ordinateurs de bureau, les tablettes et les smartphones.

SAE (Situation d'Apprentissage et d'Évaluation) : En BUT, c'est une approche pédagogique qui combine des activités d'apprentissage avec des évaluations concrètes. Elle met l'accent sur l'application des savoirs, compétences et savoir-faire dans des projets pratiques, souvent en lien avec le monde professionnel.

UI : User Interface, ou interface utilisateur, désigne l'ensemble des éléments graphiques et interactifs avec lesquels un utilisateur interagit dans une application, un site web ou tout autre système numérique.

UML : Acronyme pour Unified Modeling Language. Langage de modélisation utilisé pour visualiser, spécifier, construire et documenter des systèmes logiciels, en créant des

diagrammes comme les diagrammes de cas d'utilisation, de classe, de séquence, de composant, etc.

UX : User Experience, ou expérience utilisateur, englobe tous les aspects de l'interaction d'un utilisateur avec une entreprise, ses services et ses produits. L'objectif principal de l'UX est d'améliorer la satisfaction de l'utilisateur en rendant ses interactions avec le système aussi simples, agréables et efficaces que possible.

Web App : Application web accessible via un navigateur, qui fonctionne en ligne et ne nécessite pas de téléchargement. Elle peut offrir des services comme la gestion de contenu, la saisie de données, ou des interactions utilisateurs, souvent basée sur des technologies comme HTML, CSS, JavaScript et des frameworks backend.

## 11. ANNEXES

Figure 1



*Page de garde d'un parcours*

Figure 2

  
**Parcours Informatique (Violet)**  
(Suivez les indices de la couleur de votre parcours pour ne pas vous perdre)

Informations Personnelles :

Nom : \_\_\_\_\_ Prénom : \_\_\_\_\_  
Sexe : \_\_\_\_\_  
Filière : \_\_\_\_\_  
Heure de départ : \_\_\_\_\_  
Heure de retour : \_\_\_\_\_

Explore Game :

Chère étudiante, cher étudiant,

**Bienvenue à l'IUT' O !**  
Vous allez participer à cet Explore Game immersif au sein de l'IUT d'Orléans.  
Vous allez partir à la recherche de Diego, un étudiant très étourdi. Arrivé à l'IUT il y a quelques jours, il a malheureusement perdu son chemin en se promenant sur le campus. Votre mission est de retracer son parcours pour le retrouver.

Pour cela, vous allez devoir répondre à des questions et résoudre des énigmes dans chaque département ou bâtiment pour récolter des lettres. En combinant toutes les lettres obtenues, vous formerez un mot-clé final qui vous mènera au lieu où il s'est égaré. Répondez à ce questionnaire et rendez-vous avant 16h sur l'esplanade.

Et si vous souhaitez faire une pause, profitez des animations de l'esplanade !

**Êtes-vous prête, prêt à relever ce défi ?**

**LIEU NUMERO 1 : BATIMENT CAP**  
*Inscription : Bienvenue dans le bâtiment CAP. Répondez aux questions suivantes avant de chercher l'indice qui indiquera votre prochaine destination.*

- 1) Dans quel bâtiment se trouve la cafétéria ?
- 2) Citez au moins deux plats proposés à la cafétéria ?
- 3) Quelles sont les options de paiement disponibles ?
- 4) Qu'est-ce que Izly ?

Lettre à récolter :

**LIEU NUMERO 2 :**  
*Bienvenue ici. Diego è eu cours dans ce département. Répondez à la question puis chercher qui vous indiquera où se trouve votre prochaine et la lettre à récolter*

- 5) Combien de radiateurs sont présents dans le hall géo-climatique ?

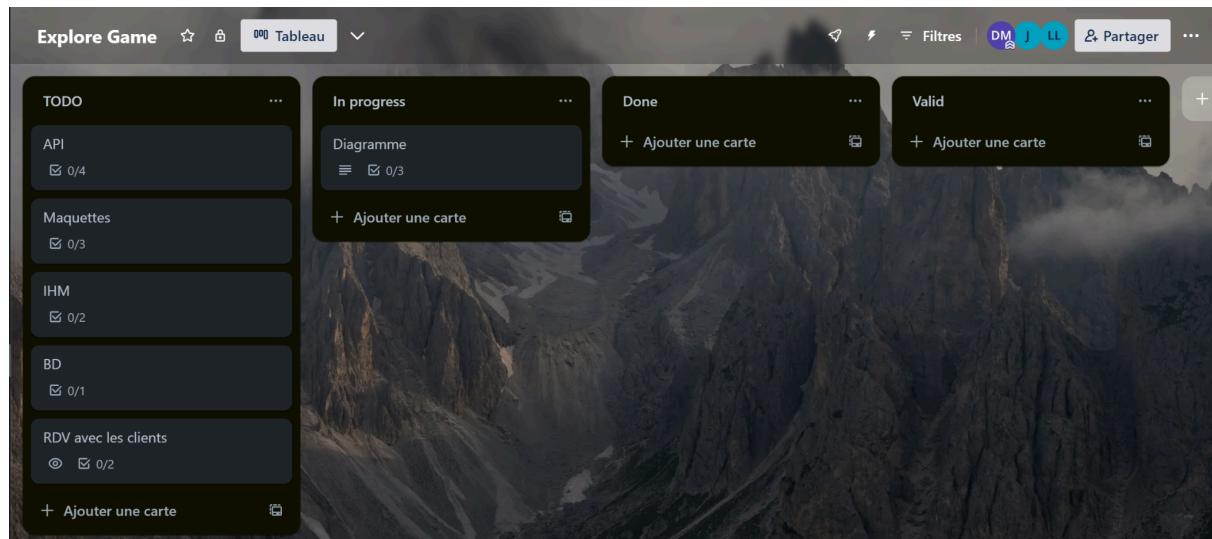
Lettre à récolter :

**LIEU NUMERO 3 :**  
*Bienvenue, dans le département de Diego. Répondez aux questions 6 à 13 de votre questionnaire et trouvez le panneau d'affichage qui indiquera votre prochaine destination et la lettre à récolter.*

- 6) Où se trouve le secrétariat de mon département ?
- 7) Quel est le nom du ou des secrétaires de mon département ?
- 8) A qui dois-je m'adresser si je souhaite prévenir d'une absence ?
- 9) Combien existe-t-il de parcours dans mon département ? (s'aider d'internet si besoin)

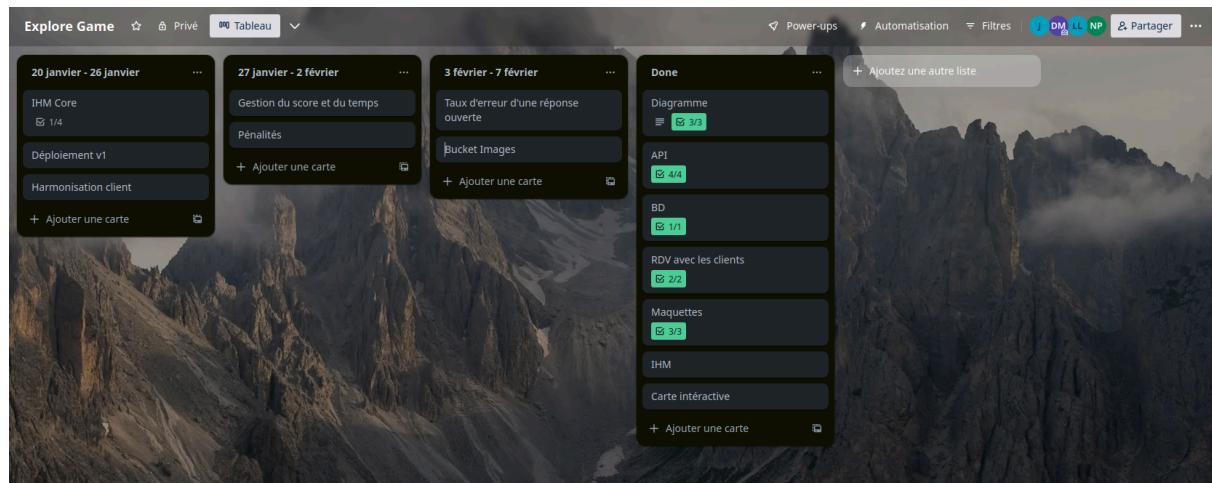
Figure 1 : Extrait de suite de questions d'un parcours

Figure 3



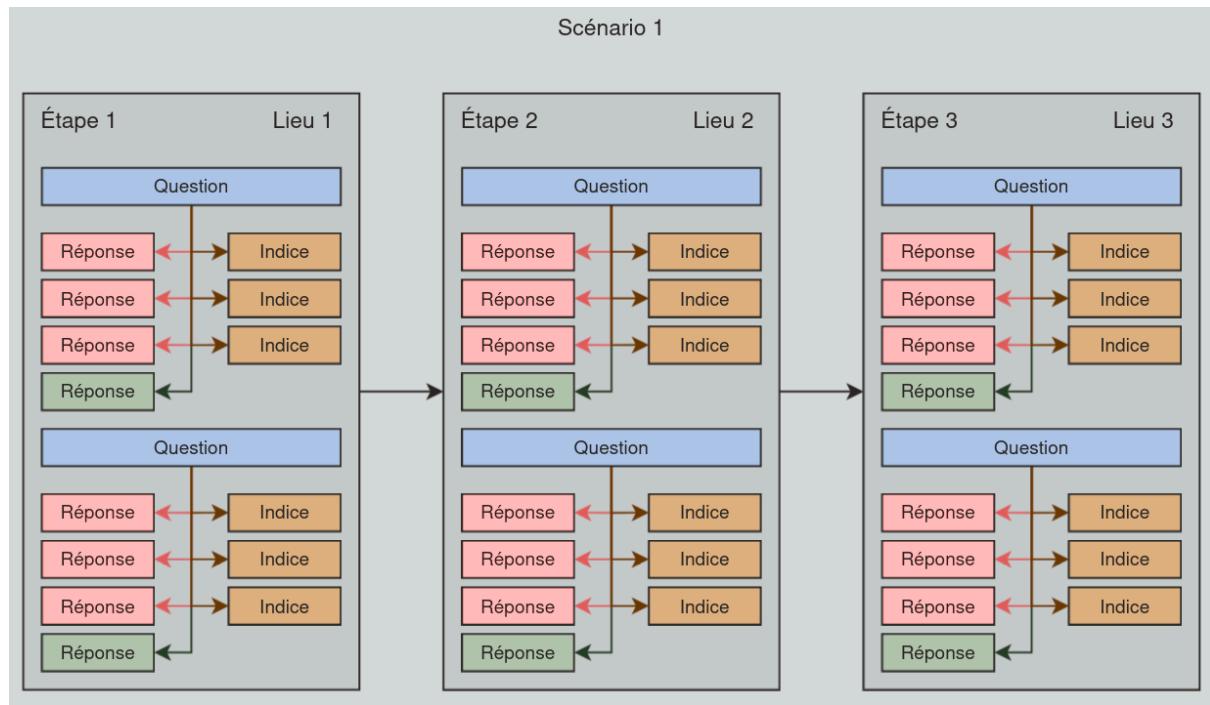
Première version de notre Trello (septembre 2024)

Figure 4



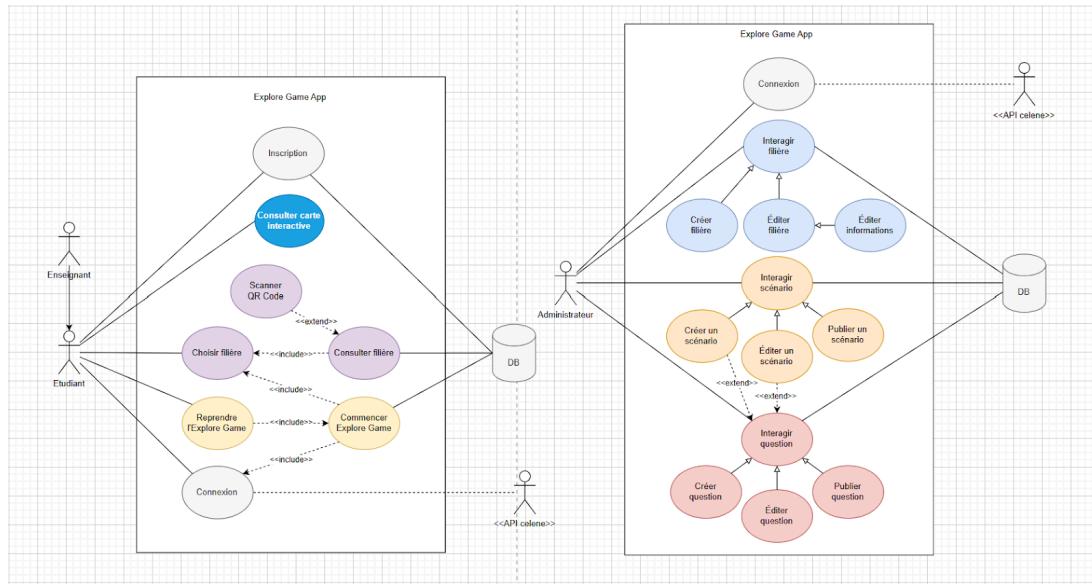
Version de notre Trello (16 janvier 2024)

Figure 5



Modélisation de l'arborescence d'un scénario

Figure 6



Premier schéma de cas d'utilisation de l'explore game

Figure 7

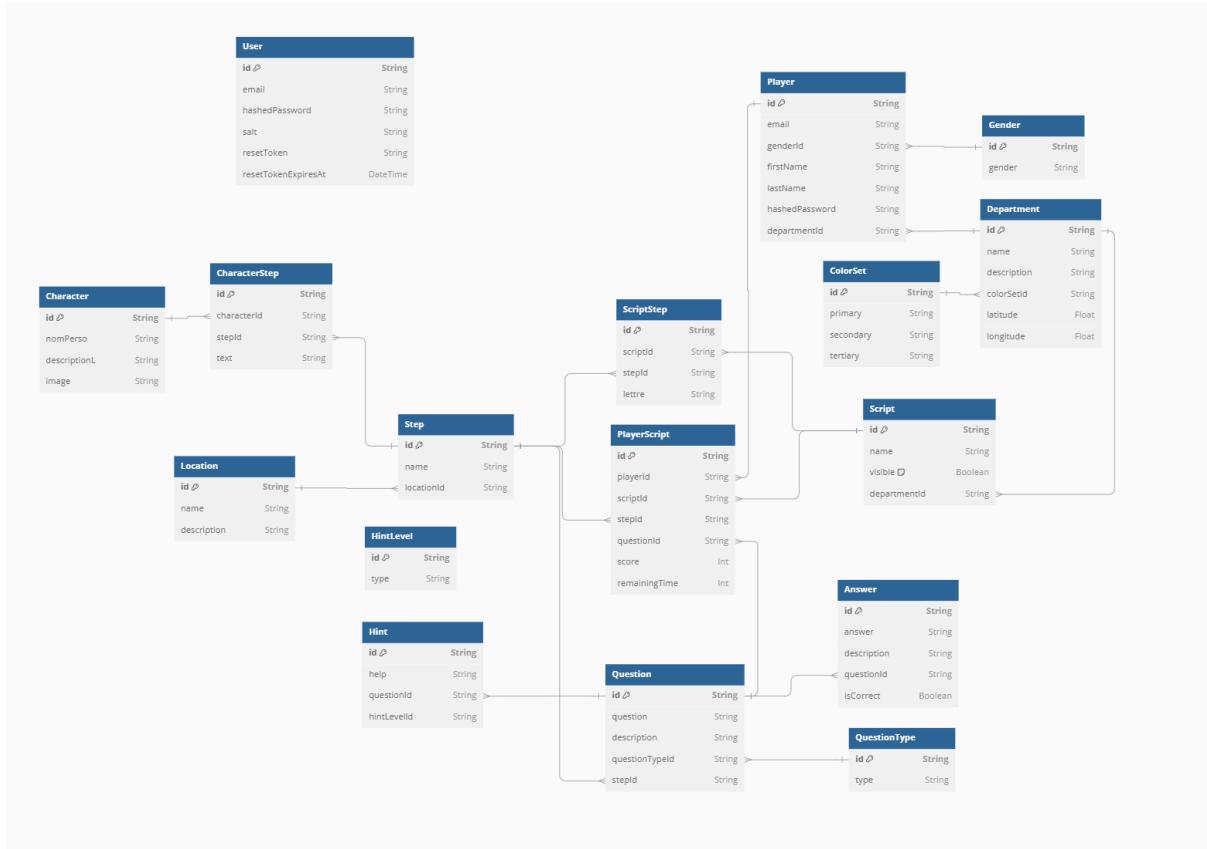


Schéma de modèle conceptuel de données

Figure 8



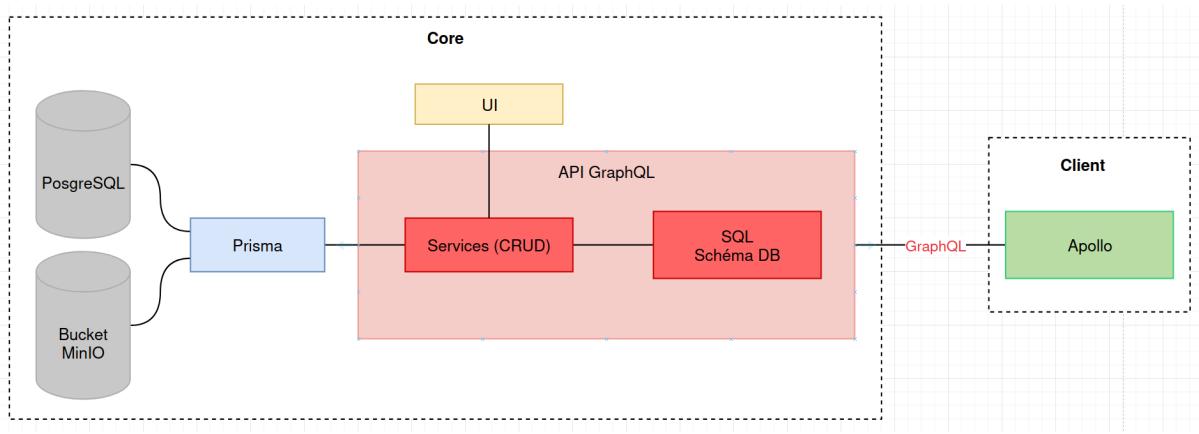
*Palette de couleur de l'Explore Game*

Figure 9



*Palette de couleur du département Informatique de l'Explore Game*

Figure 10



*Architecture des micro services (core & client)*

Figure 11

```
model Question {
    id          String      @id @default(uuid())
    question    String
    description String
    questionTypeId String
    stepId      String
    Step         Step        @relation(fields: [stepId], references: [id])
    QuestionType QuestionType @relation(fields: [questionTypeId], references: [id])
    Answer      Answer[]
    Hint         Hint[]
    PlayerScript PlayerScript[]
}
```

*Extrait du schéma (définition du modèle Question)*

Figure 12

```
export type Question = {
    __typename?: 'Question';
    Answer: Array<Maybe<Answer>>;
    Hint: Array<Maybe<Hint>>;
    PlayerScript: Array<Maybe<PlayerScript>>;
    QuestionType: QuestionType;
    Step: Step;
    description: Scalars['String'];
    id: Scalars['String'];
    question: Scalars['String'];
    questionTypeId: Scalars['String'];
    stepId: Scalars['String'];
};
```

*Exemple de type généré automatiquement dans le micro serviceCore*

Figure 13

```
export const question: QueryResolvers['question'] = ({ id }) => {
  return db.question.findUnique({
    where: { id },
  })
}

export const createQuestion: MutationResolvers['createQuestion'] = ({ input,
}) => [
  return db.question.create({
    data: input,
  })
]

export const updateQuestion: MutationResolvers['updateQuestion'] = ({ id,
  input,
}) => {
  return db.question.update({
    data: input,
    where: { id },
  })
}

export const deleteQuestion: MutationResolvers['deleteQuestion'] = ({ id }) => {
  return db.question.delete({
    where: { id },
  })
}
```

*Exemple du service CRUD du modèle Question*

Figure 14

```
<Set wrap={ScaffoldLayout} title="Questions" titleTo="questions" buttonLabel="New Question" buttonTo="newQuestion">
  <Route path="/questions/new" page={QuestionNewQuestionPage} name="newQuestion" />
  <Route path="/questions/{id}/edit" page={QuestionEditQuestionPage} name="editQuestion" />
  <Route path="/questions/{id}" page={QuestionQuestionPage} name="question" />
  <Route path="/questions" page={QuestionQuestionsPage} name="questions" />
</Set>
```

*Extrait des routes créée automatiquement*

Figure 15

```
const client = new ApolloClient({
  uri: import.meta.env.VITE_API_URL_GRAPHQL,
  cache: new InMemoryCache()
})

createRoot(document.getElementById('root')!).render(
  <ApolloProvider client={client}>
    <App />
  </ApolloProvider>
)
```

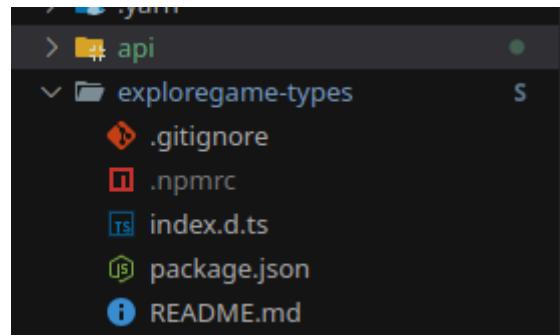
Définition du client Apollo provider (micro service **Client**)

Figure 16

```
export const SCENARIO = gql`query FindScenarioById($id: String!) {
  Execute Query
  script(id: $id) {
    id
    name
    ScriptStep {
      stepId
      Step {
        id
        Questions {
          id
        }
      }
    }
    PlayerScript {
      id
      playerId
      stepId
      questionId
      completed
    }
  }
};`
```

Exemple de requête GraphQL

Figure 17



*Submodule de types-sharing*

Figure 18

**@exploregame/types** ts

1.1.1 • Public • Published 4 days ago

[Readme](#) [Code](#) Beta [0 Dependencies](#) [0 Dependents](#) [10 Versions](#) [Settings](#)

## Explore Game - Types

Librairie de **types-sharing** pour le projet `Explore Game` de l'IUT d'Orléans.

Ce paquet permet de partager les types entre les différents microservices du projet.

### Microservices

- **core** : Microservice principal, défini le schéma de la base de donnée, distribue l'API et gère les requêtes.  
*node.js - redwoodjs - react - graphql - prisma - apollo*
- **client** : Microservice de l'interface des clients, connectés via l'API du microservice core. Cette interface permet d'utiliser l'application *Explore Game* en tant qu'étudiant / lycéen / collégien.  
*node.js - vite - react - apollo*

### Authors

4 étudiants de BUT Informatique de l'IUT d'Orléans :

- Jordan LAVENANT

### Install

```
> npm i @exploregame/types
```

### Weekly Downloads

169

### Version

1.1.1

### License

All rights reserved

### Unpacked Size

90 kB

### Total Files

3

### Last publish

4 days ago

### Collaborators

*Paquet npm de types-sharing (<https://www.npmjs.com/package/@exploregame/types>)*

Figure 19

Questions + NEW QUESTION

**Edit Question 1**

**Question**

**Description**

**Question type id**

**Step id**

SAVE



*Formulaire créée par défaut*

Figure 20

Questions + NEW QUESTION

**Question**

Intitulé  
Quel est votre nom ?

Description  
Entrez votre nom

Type de la question

Texte

Etat ✓ Texte  
Choix  
Date

**Réponses associées**

Nouvelle réponse Ajouter

Description

Pas de réponse associées

**Indices associées**

*Formulaire personnalisé*

Figure 21

The screenshot shows a mobile application interface. At the top is a purple header bar with the text "iUT'0" in white. To the right of the logo is a vertical menu icon consisting of three horizontal lines. Below the header, the screen is divided into four main sections, each enclosed in a rounded rectangular box with a light purple background and a dark purple border.

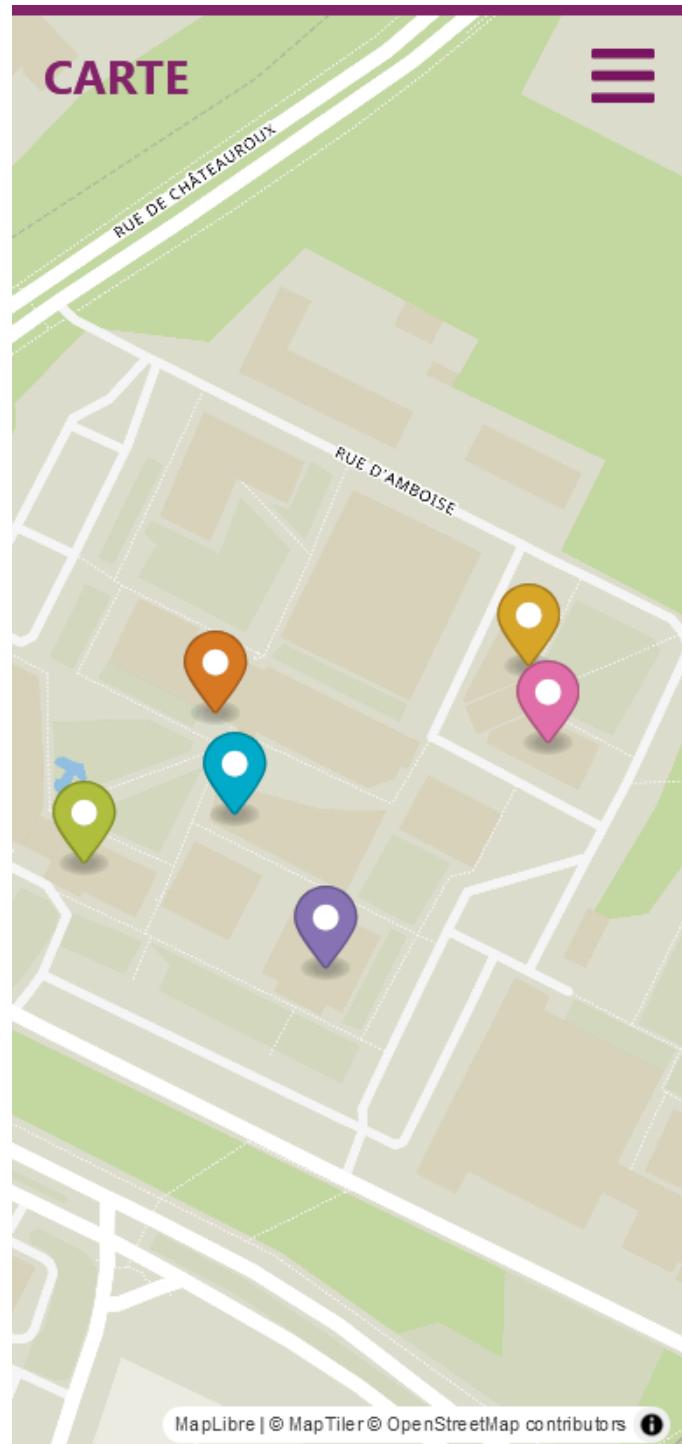
- Bienvenue**: Contains the text "Welcome to the Home Page" and "This is the Home Page of the app. You can add more content here."
- Carte**: Contains the text "Welcome to the Home Page" and "This is the Home Page of the app. You can add more content here."
- Actualités**: Contains the text "Welcome to the Home Page" and "This is the Home Page of the app. You can add more content here."
- Les filières**: Contains the text "Welcome to the Home Page" and a button labeled "Explorer".

Figure 22



*Menu Burger*

Figure 23



*Carte interactive*

Figure 24



**Explorer**

## Bienvenue

**Welcome to the Home Page**

This is the Home Page of the app. You can add more content here.

## Carte

**Welcome to the Home Page**

This is the Home Page of the app. You can add more content here.

## Actualités



*Page de département (département informatique)*

Figure 25



## Bienvenue

### Welcome to the Home Page

This is the Home Page of the app. You can add more content here.

## Carte

### Welcome to the Home Page

This is the Home Page of the app. You can add more content here.

## Actualités

*Page de département (département GEA)*

Figure 26



*Page de département (département Chimie)*

Figure 27

## X BATIMENTS ADMINISTRATIF



Jérémie: Bonjour, je m'appelle Jérémie, je suis un futur étudiant de l'iut d'Orléans

Panda: Salut, je suis Panda, la mascotte officiel de l'Infasso

Jérémie: Je suis là pour découvrir les différents départements de l'iut d'Orléans

Panda: Je suis là pour t'aider à trouver les différentes étapes de ton aventure

Panda: Tu devras répondre aux questions avec les informations qui sont dans ce bâtiment.

Panda: Bonne chance ! :)

Page de dialogue

Figure 28

**X BATIMENTS ADMINISTRATIF**



Dans quel bâtiment se trouve la cafétéria ?

Bâtiment CAP

Bâtiment GEO

Bâtiment A

Bâtiment B

 **Correcte !** 

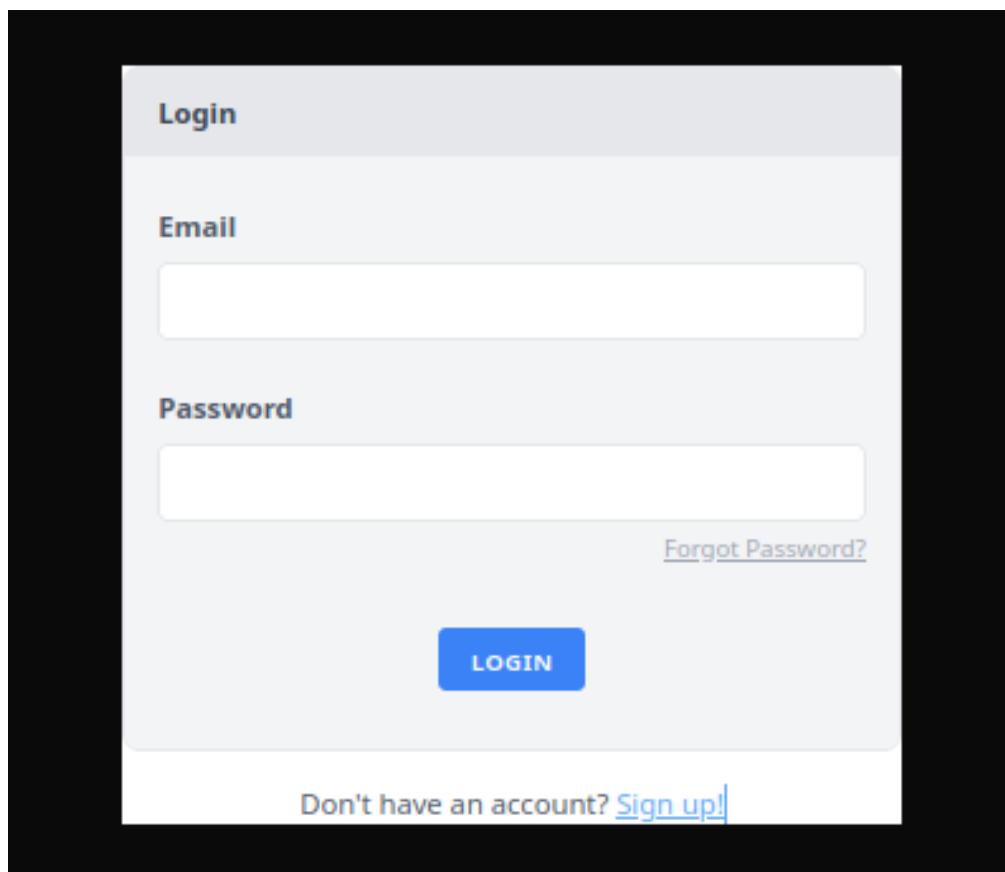
**Continuer**

*Page de question*

Figure 29



Figure 30



*Interface provisoire d'authentification*

## Figure 31

Departments						+ NEW DEPARTMENT
Id	Name	Description	Color set id	Latitude	Longitude	
1	INFO	BUT Informatique	1	47.8433084	1.9267556	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
2	GEA	BUT Gestion des Entreprises et des Administrations	2	47.8437317	1.9263861	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
3	GMP	BUT Génie Mécanique et Productique	3	47.8440098	1.9263063	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
4	QLIO	BUT Qualité, Logistique Industrielle et Organisation	4	47.8441395	1.927583	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
5	Chimie	BUT Chimie	6	47.8439269	1.927661	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>
6	MT2E	BUT Métiers de la transition et de l'efficacité énergétiques	5	47.8435985	1.9257699	<a href="#">SHOW</a> <a href="#">EDIT</a> <a href="#">DELETE</a>

*Interface provisoire de listing des occurrences d'une entité (ici département)*

## Figure 32

Department 1 Detail		+ NEW DEPARTMENT
<b>Id</b>	1	
<b>Name</b>	INFO	
<b>Description</b>	BUT Informatique	
<b>Color set id</b>	1	
<b>Latitude</b>	47.8433084	
<b>Longitude</b>	1.9267556	

*Interface provisoire du détail d'une occurrence (ici département)*

Figure 33

Departments

+ NEW DEPARTMENT

Edit Department 1

Name  
INFO

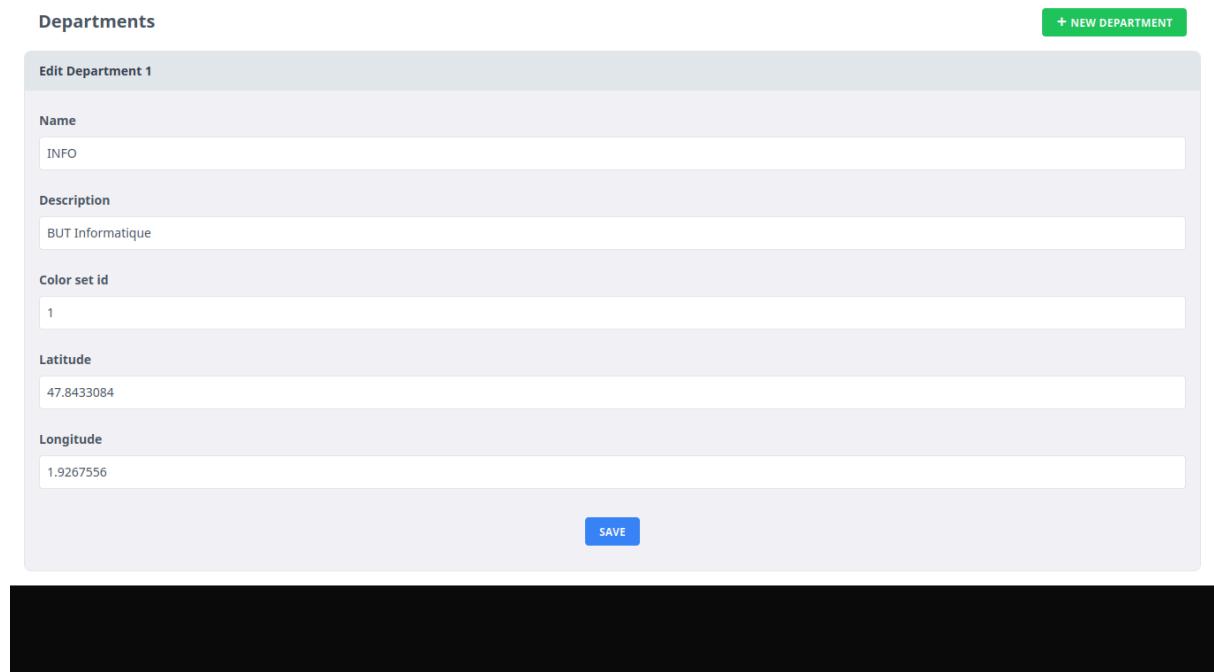
Description  
BUT Informatique

Color set id  
1

Latitude  
47.8433084

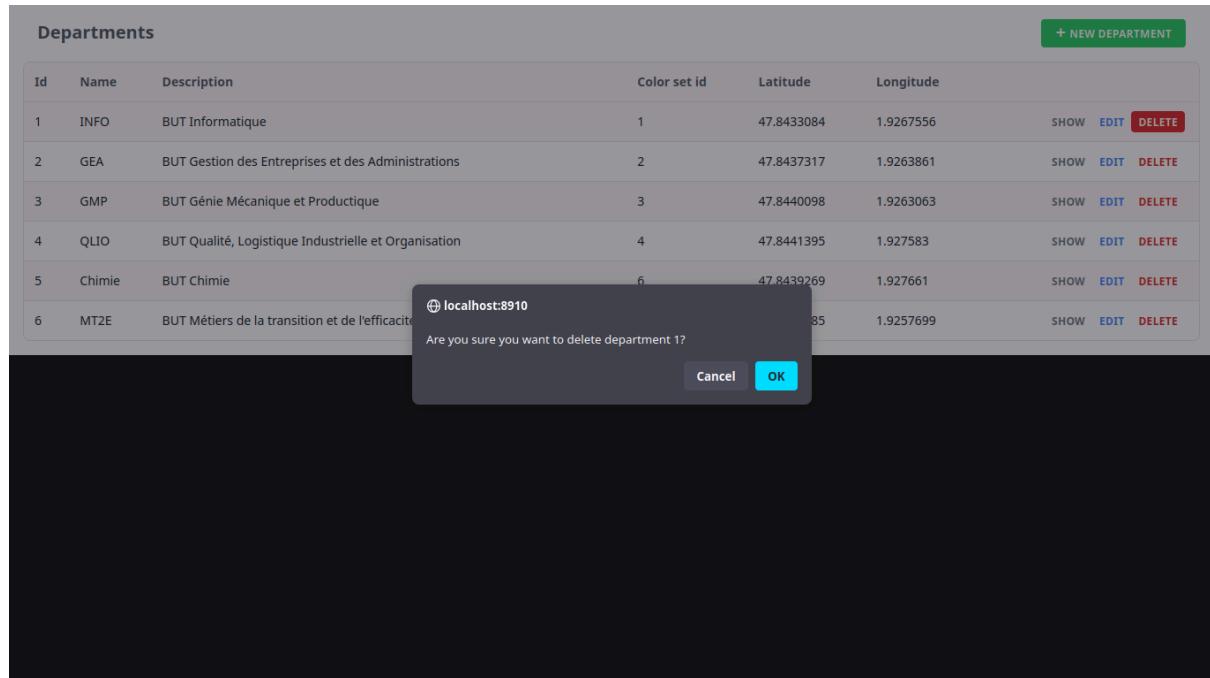
Longitude  
1.9267556

SAVE



*Interface provisoire d'édition d'une occurrence (ici département)*

Figure 34



*Exemple de suppression d'une occurrence (ici département)*

Figure 35

The screenshot shows a form for editing a question. It is divided into three main sections: "Question", "Réponses associées", and "Indices associés".

- Question:** Contains fields for "Intitulé" (with the value "Dans quel bâtiment se trouve la cafétéria ?"), "Description" (empty), "Type de la question" (set to "Choix"), and "Etape associée" (set to "Step 1").
- Réponses associées:** Contains a "Nouvelle réponse" input field and an "Ajouter" button. Below this are four items: "Bâtiment CAP" (selected, indicated by a blue toggle switch), "Bâtiment GEO" (not selected), "Bâtiment A" (not selected), and "Bâtiment B" (not selected).
- Indices associés:** This section is currently empty.

At the bottom of the form are two buttons: "Annuler" (Cancel) and "Sauvegarder" (Save).

*Interface provisoire d'édition d'une occurrence (ici question)*