



DOSSIER PROJET

Julia Spriggs



Plan

Gestion du projet

Conception de la base de données

Unified Modeling Language

Maquettisation de l'application

L'environnement de travail

Installation du projet

Créer un matériel

Préparer le déploiement

Jeu d'essai

GESTION DU PROJET

- Organisation
- Type de semaine et déroulement
- Présentation de l'équipe



Organisation

The image shows a Kanban board for a project named "Oros". The board is organized into five columns representing different stages of the workflow:

- Backlog (A FAIRE):** Contains tasks like "Etoffer les tests? (pas de test front)", "Filtrage des matériels", "Réaliser le back-office", and "Panier".
- EN COURS (BRANCHE PERSO):** Contains tasks like "Authentification" and "Liste des matériels".
- PRET A DEPLOYER (DEV):** Currently empty.
- ON PROD (MAIN):** Contains tasks like "Backend: installation des dépendances", "Installation de Docker", "Front-end: installation des dépendances", "Installation Typescript", "Créer serveur Apollo avec Express", "Intégration Continue", "Tests", and "Bien mettre à jour tous les InputTypes suite aux relations qui ont été ajoutées (enlever category_id et mettre category par ex)".
- ABANDON (ARCHIVE):** Contains tasks like "Initialisation du serveur Apollo" and "Mockups Figma".

Each task card includes a progress bar, a list of assignees (represented by initials like DT, AV, JS, RL), and an "Ajouter une carte" button. The background of the board is a scenic image of a snowy mountain range under a starry night sky.

Équipe du projet



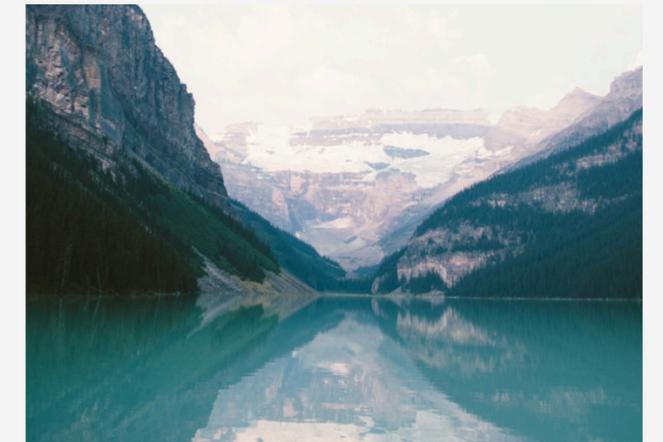
THIBAUD DELVERT



ROMARIC LAMARE



JULIA SPRIGGS



ALICIA VIVIER-MERLE

Conception de la base de données

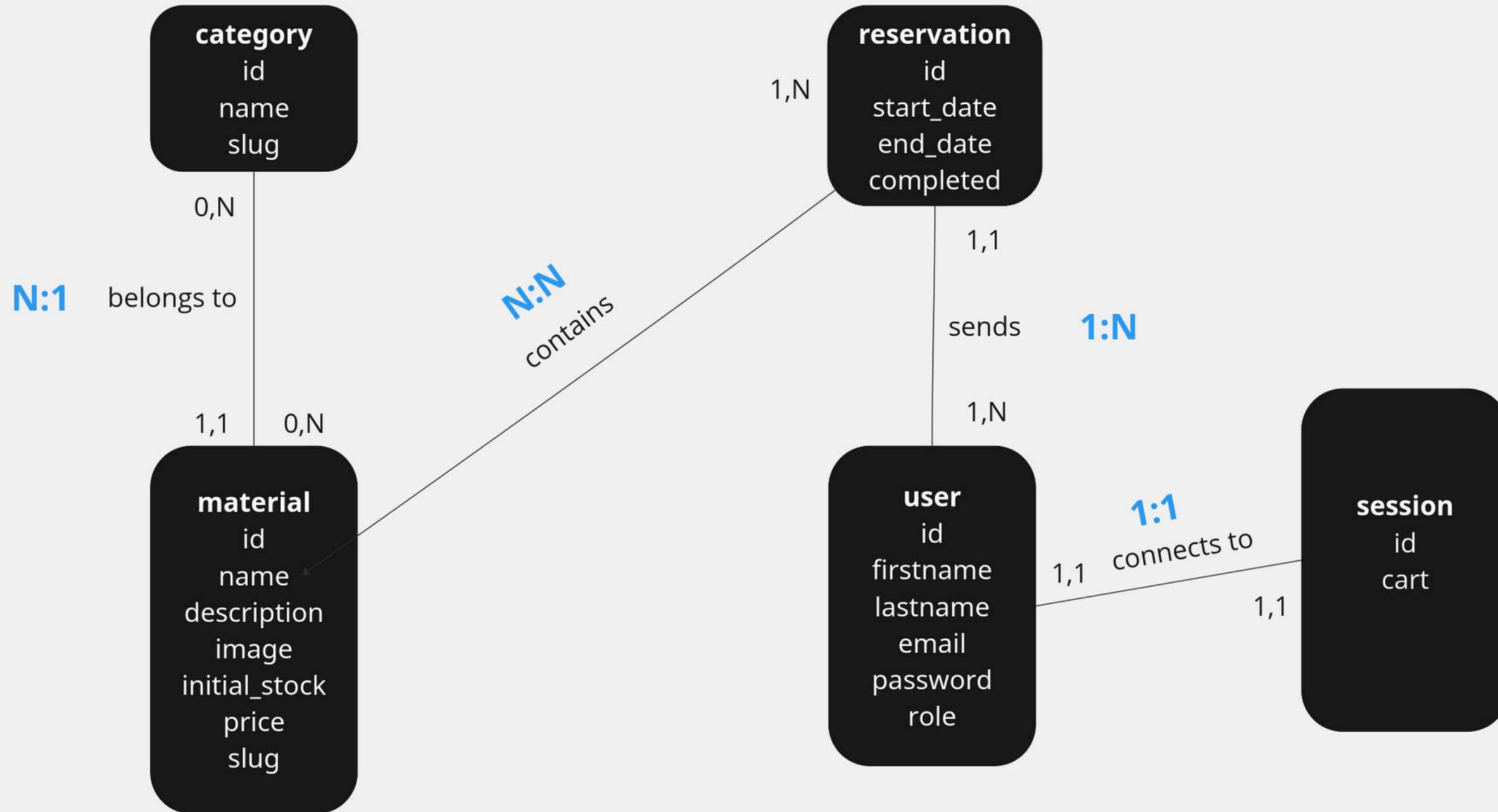
La Merise

01 MCD
Modèle Conceptuel de Données

02 MLD
Modèle Logique de Données

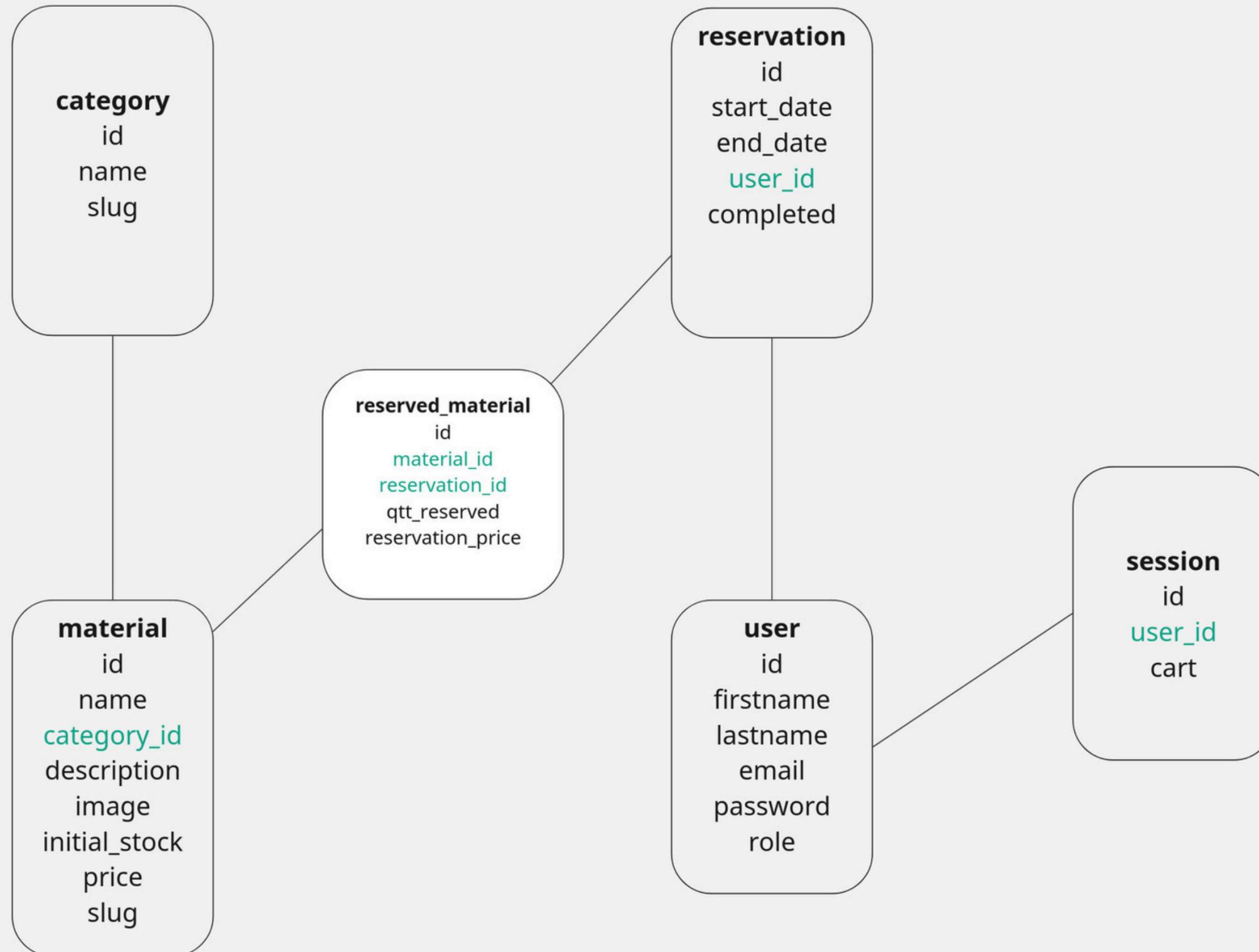
03 MPD
Modèle Physique de Données

MCD

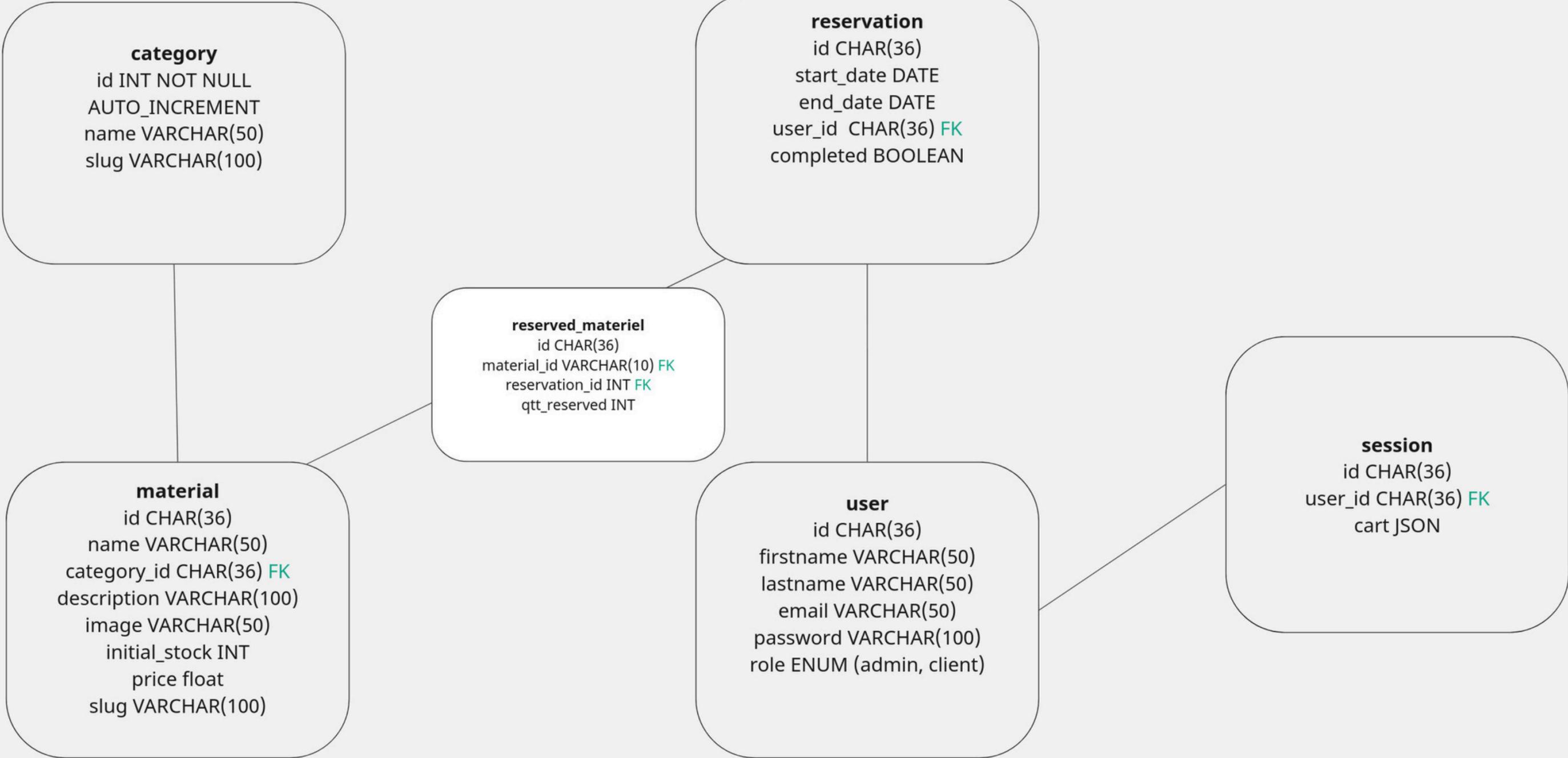


A **user** connects to a **session**, and sends a **reservation** containing **material** which belongs to a **category** and which includes the start date, end date, and if the reservation was completed.

MLD



MPD



UML

UNIFIED MODELING LANGUAGE

- Diagramme de classe
- Diagramme de séquence
- Diagramme de cas d'utilisation

class diagram

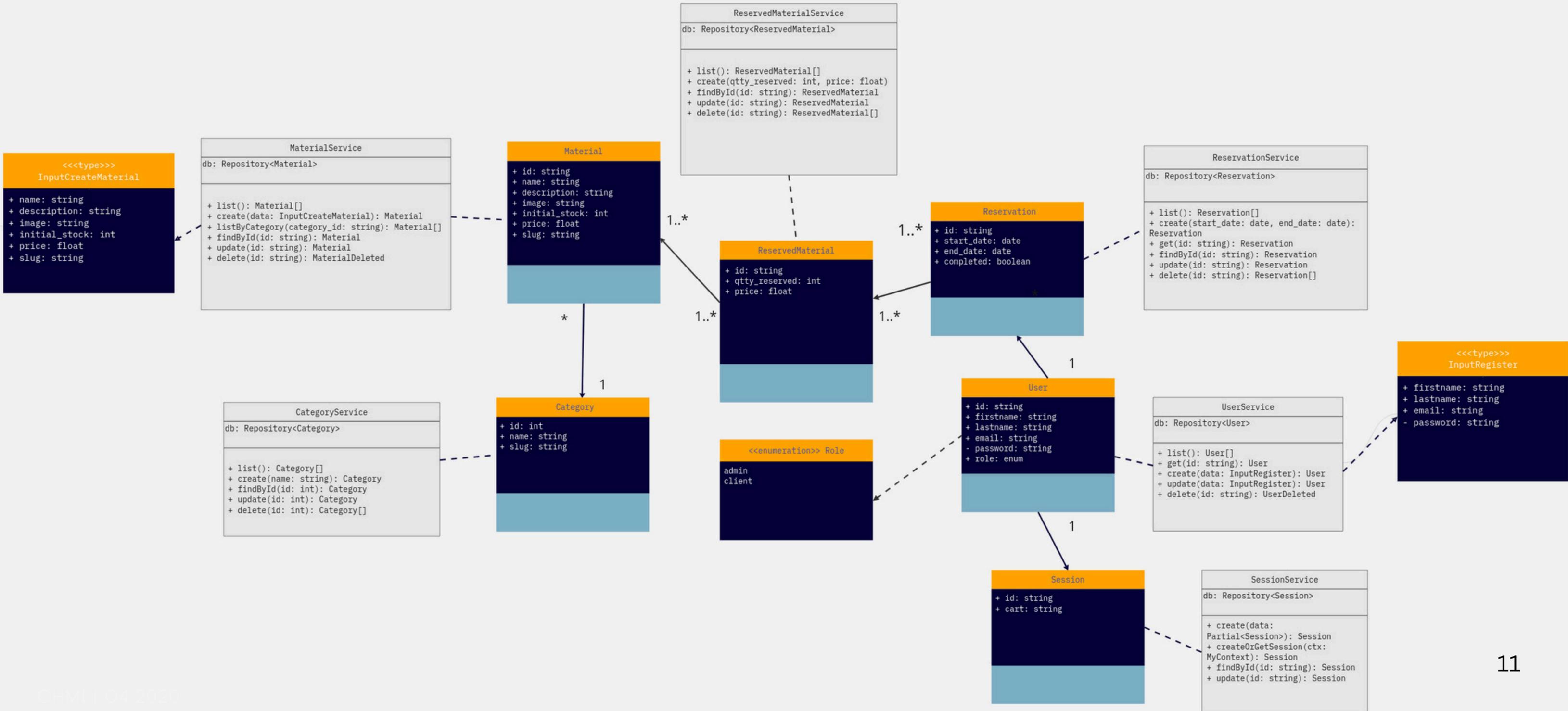


Diagramme de classe

- Multiplicités
- Portée

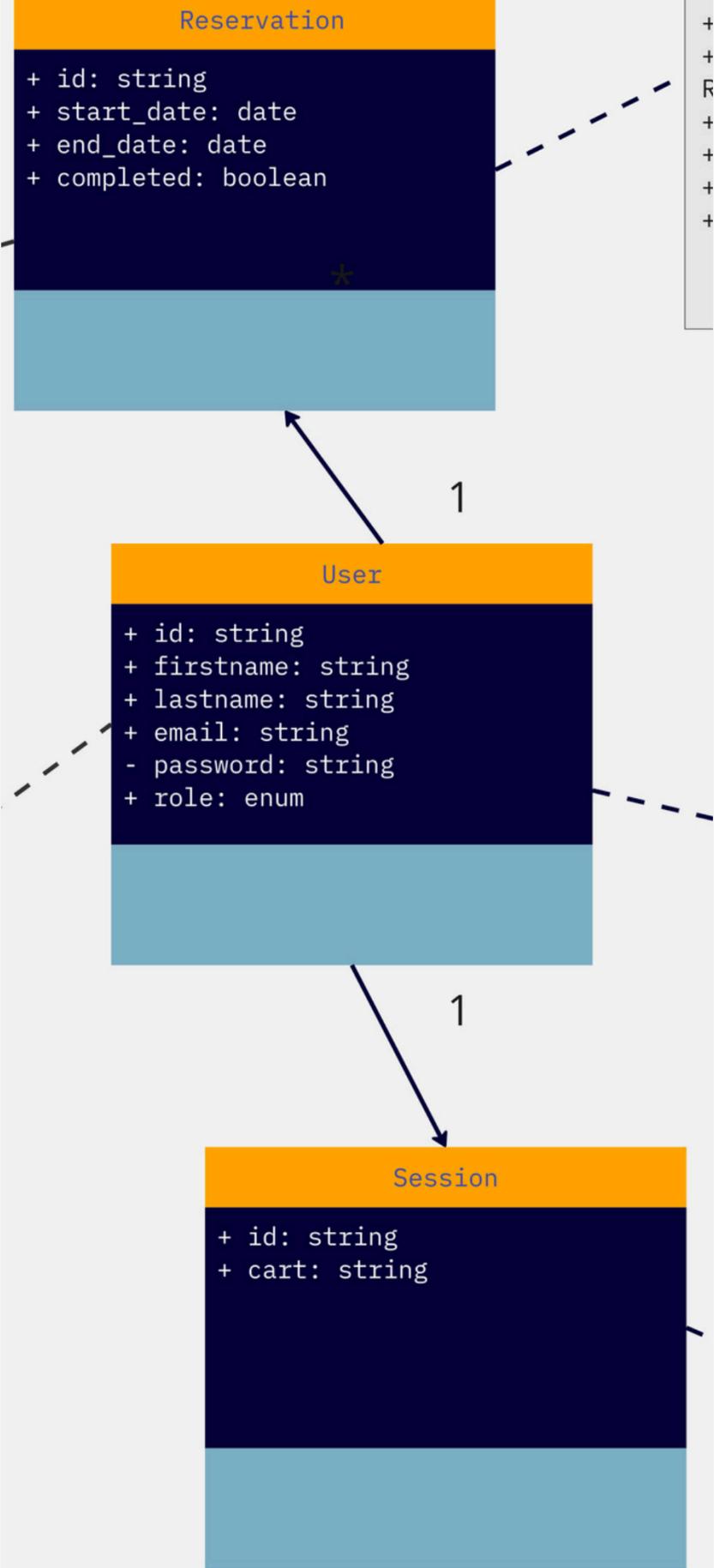
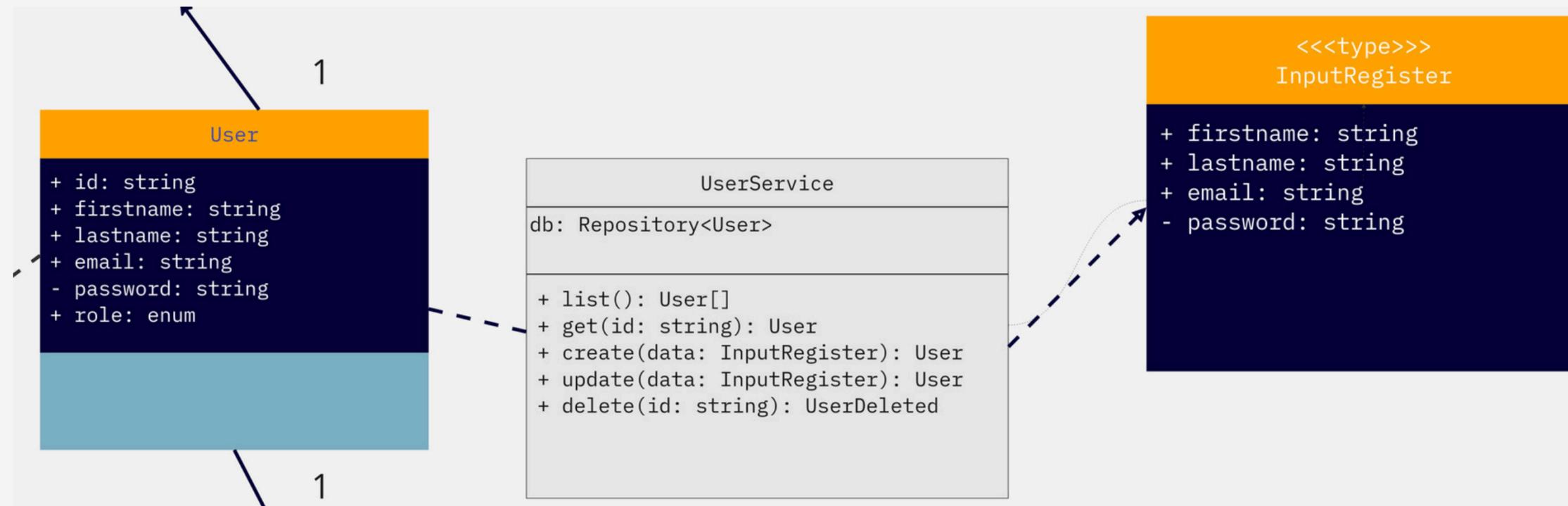


Diagramme de classe



```
@InputType()
export class InputRegister {
  @Field()
  @Column()
  firstname: string;

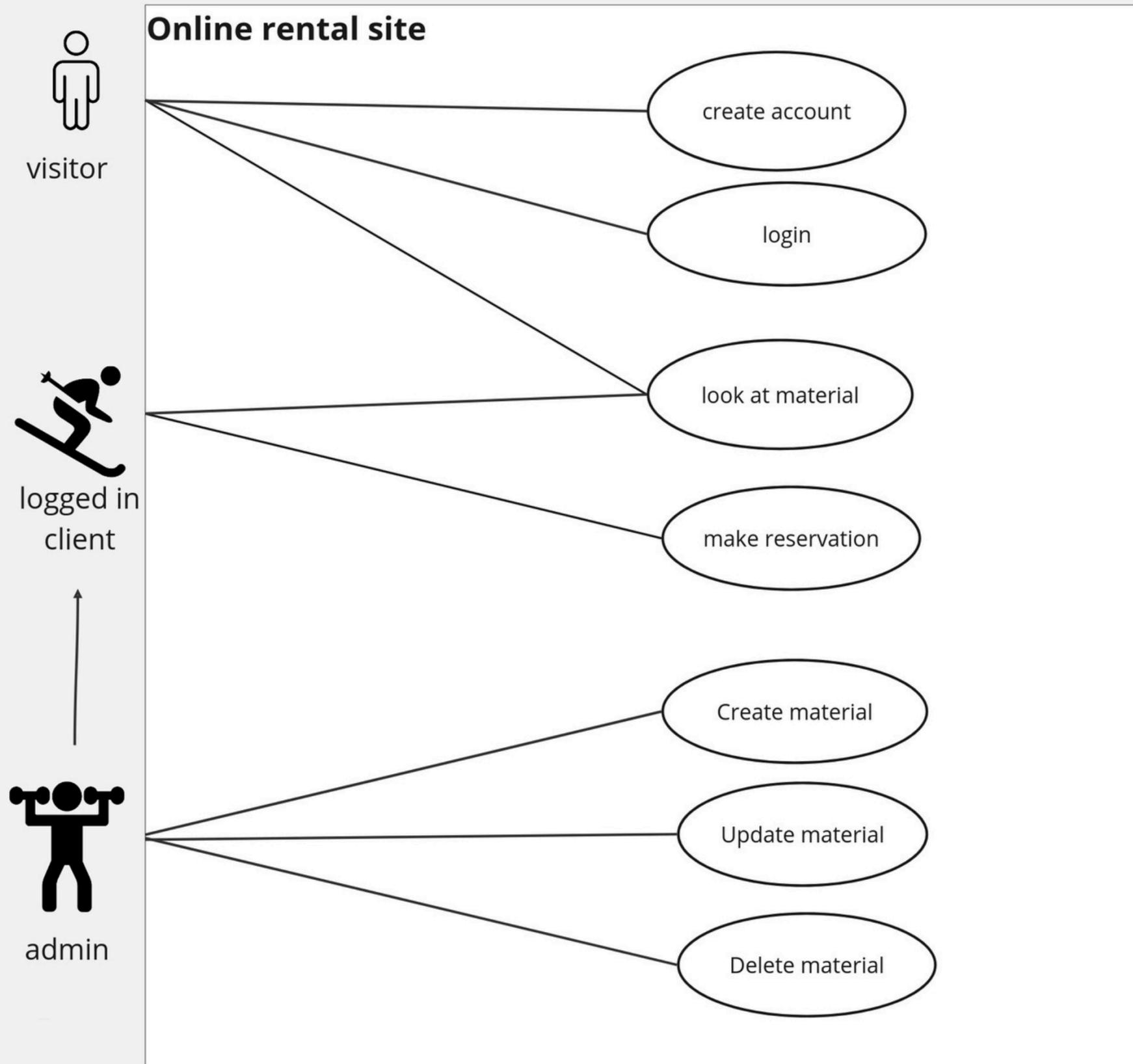
  @Field()
  @Column()
  lastname: string;

  @Field()
  @Column({ unique: true })
  email: string;

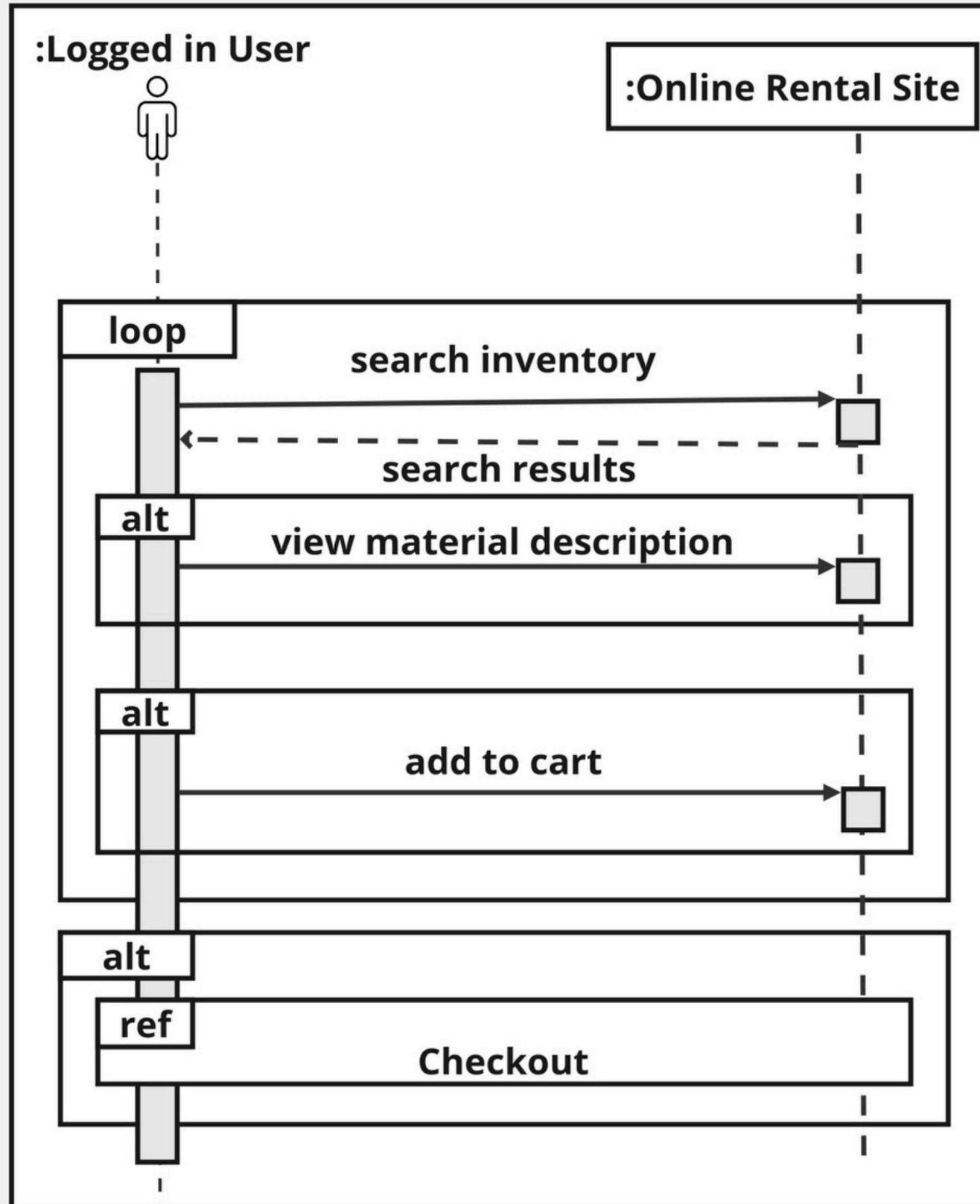
  @Field()
  @Column()
  password: string;
}
```

```
async create(data: InputRegister) {
  const newUser = await this.db.create(data);
  return await this.db.save(newUser);
}
```

use case diagram



sequence diagram



A person wearing a black jacket and a backpack with orange straps stands on a mountain ridge, looking out over a valley. The valley features terraced fields and a small village. The surrounding mountains are covered in snow. The sky is clear and blue.

MAQUETTISATION DE L'APPLICATION

Accueil et recherche



Création de compte / connexion

Accueil - sign up modal

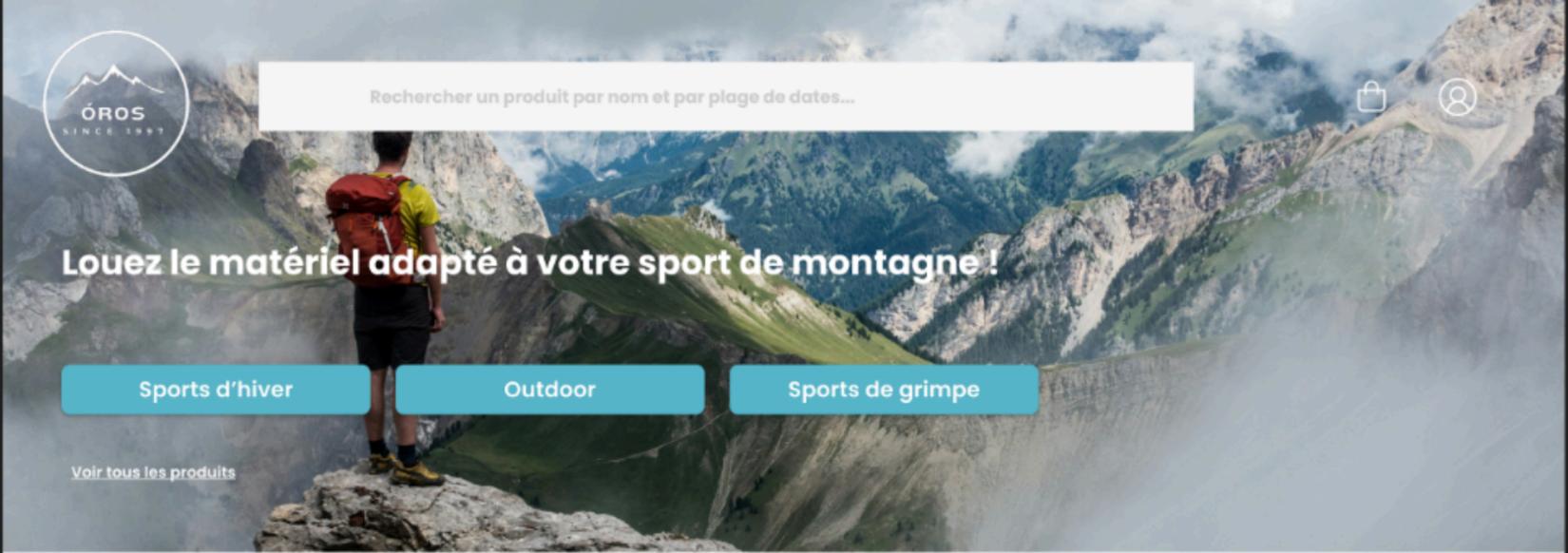
The screenshot shows the home page of the 'ÓROS' website. At the top left is the logo. A search bar contains the placeholder text 'Rechercher un produit, un sport, une référence...'. To the right are icons for a shopping bag and a user profile. Below the search bar is the heading 'Trouver le matériel adapté à votre sport de montagne !' followed by three buttons: 'Sports d'hiver', 'Outdoor', and 'Sports de grimpe'. The main content area is titled 'Top ventes' and contains several product cards. A white modal is centered on the screen with the title 'Créer un compte'. It contains three input fields labeled 'Nom', 'Email', and 'Mot de passe', and a 'Créer un compte' button at the bottom.

Accueil - sign in modal

The screenshot shows the same home page as the previous one, but with a sign-in modal open. The modal is titled 'Se connecter à votre compte' and contains two input fields labeled 'Email' and 'Mot de passe', followed by a 'Se connecter' button. Below the button is a link that says 'Vous ne possédez pas de compte ? Inscrivez-vous'. The background content of the website is dimmed.

Mockup de l'accueil

Accueil



Logo: ÓROS SINCE 1997

Rechercher un produit par nom et par plage de dates...

Louez le matériel adapté à votre sport de montagne !

Sports d'hiver Outdoor Sports de grimpe

Voir tous les produits

Top ventes

 <p>HEAD MONSTER JUNIOR SKI ALPIN 87 CM Head</p> <p>160,00€ Ajouter</p>	 <p>HEAD MONSTER JUNIOR SKI ALPIN 87 CM Head</p> <p>160,00€ Ajouter</p>	 <p>HEAD MONSTER JUNIOR SKI ALPIN 87 CM Head</p> <p>160,00€ Ajouter</p>
---	---	---

AFFICHER PLUS

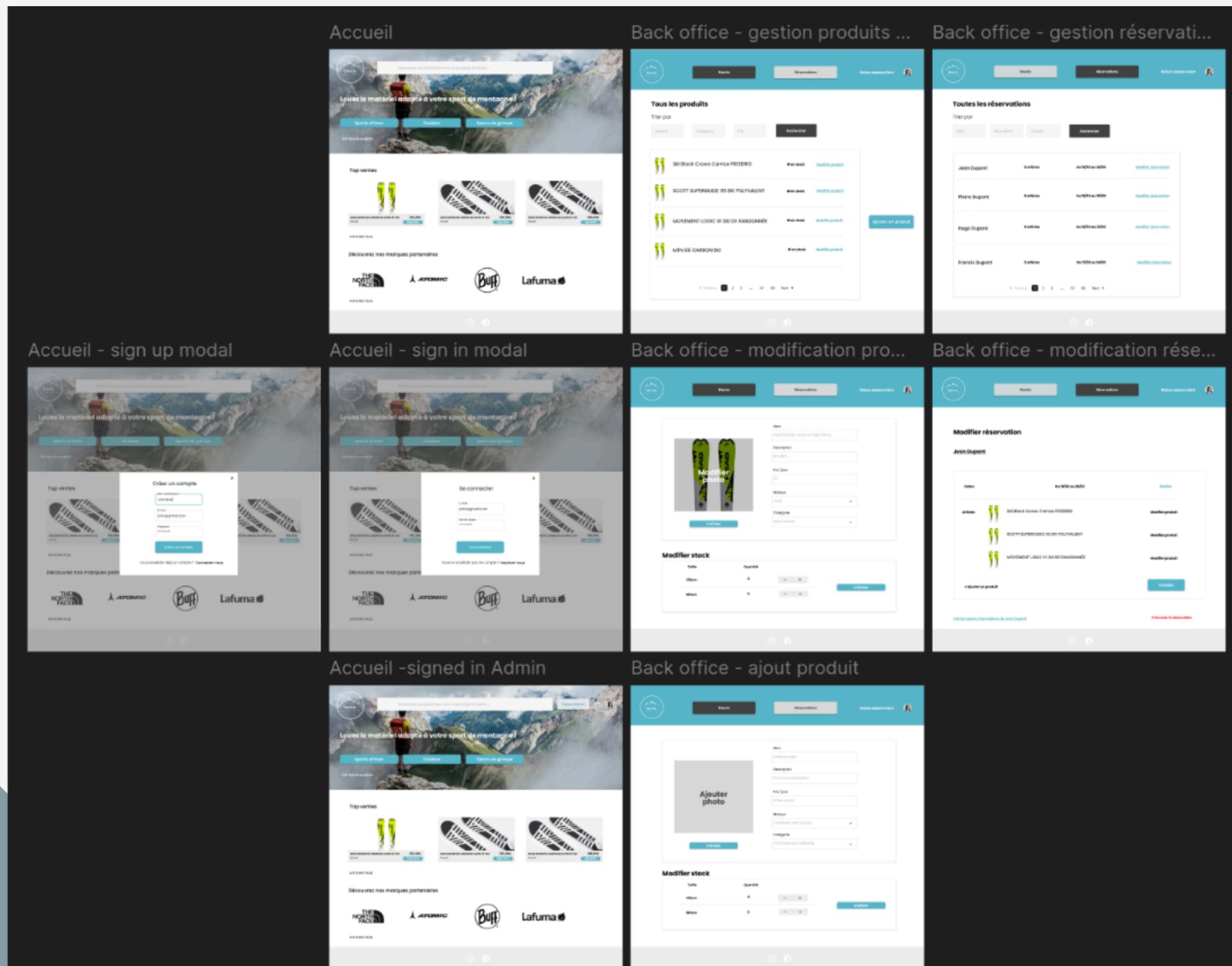
Découvrez nos marques partenaires

THE NORTH FACE ATOMIC Buff Lafuma

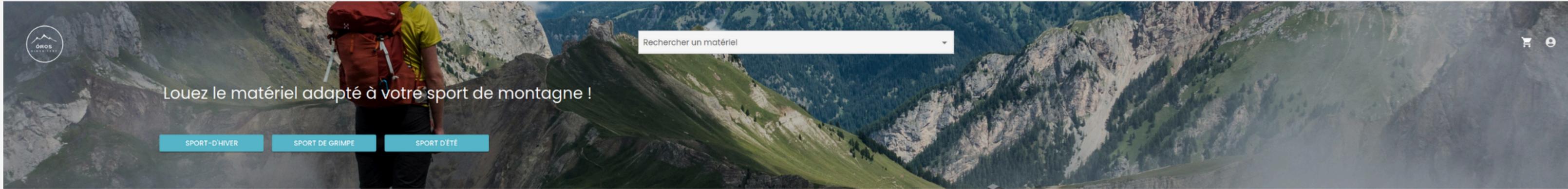
AFFICHER PLUS

Instagram Facebook

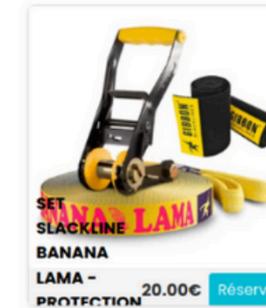
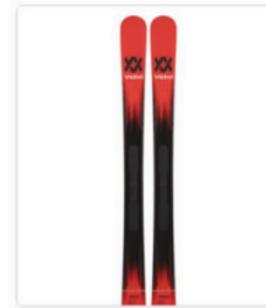
Mockup de l'espace admin



Page d'accueil



À la Une



Découvrez nos marques partenaires



À Propos

Qui sommes-nous ?
Notre histoire
Contactez-nous

Services

Louez du matériel
Réservations
Assistance

Suivez-nous



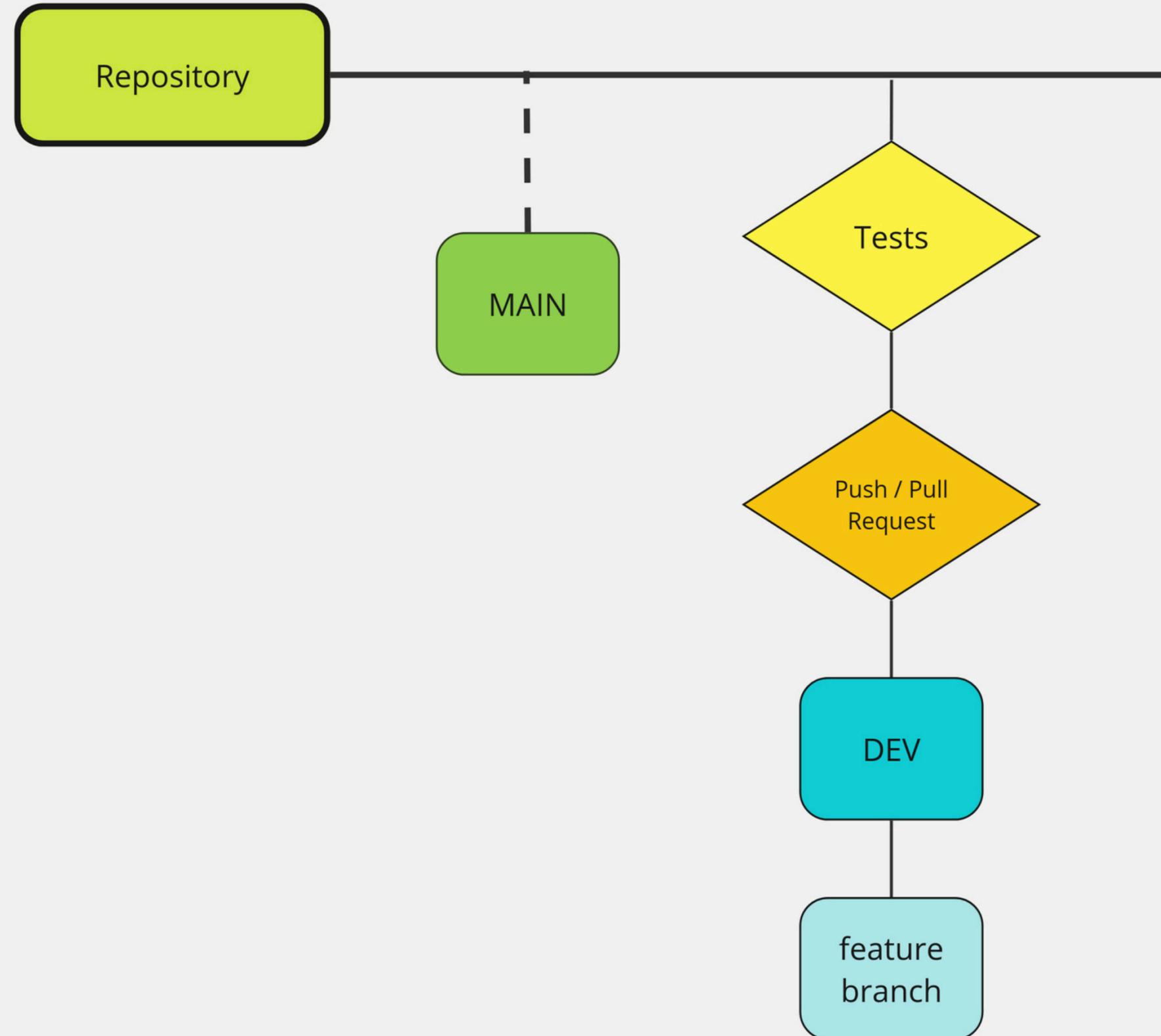


L'environnement de travail

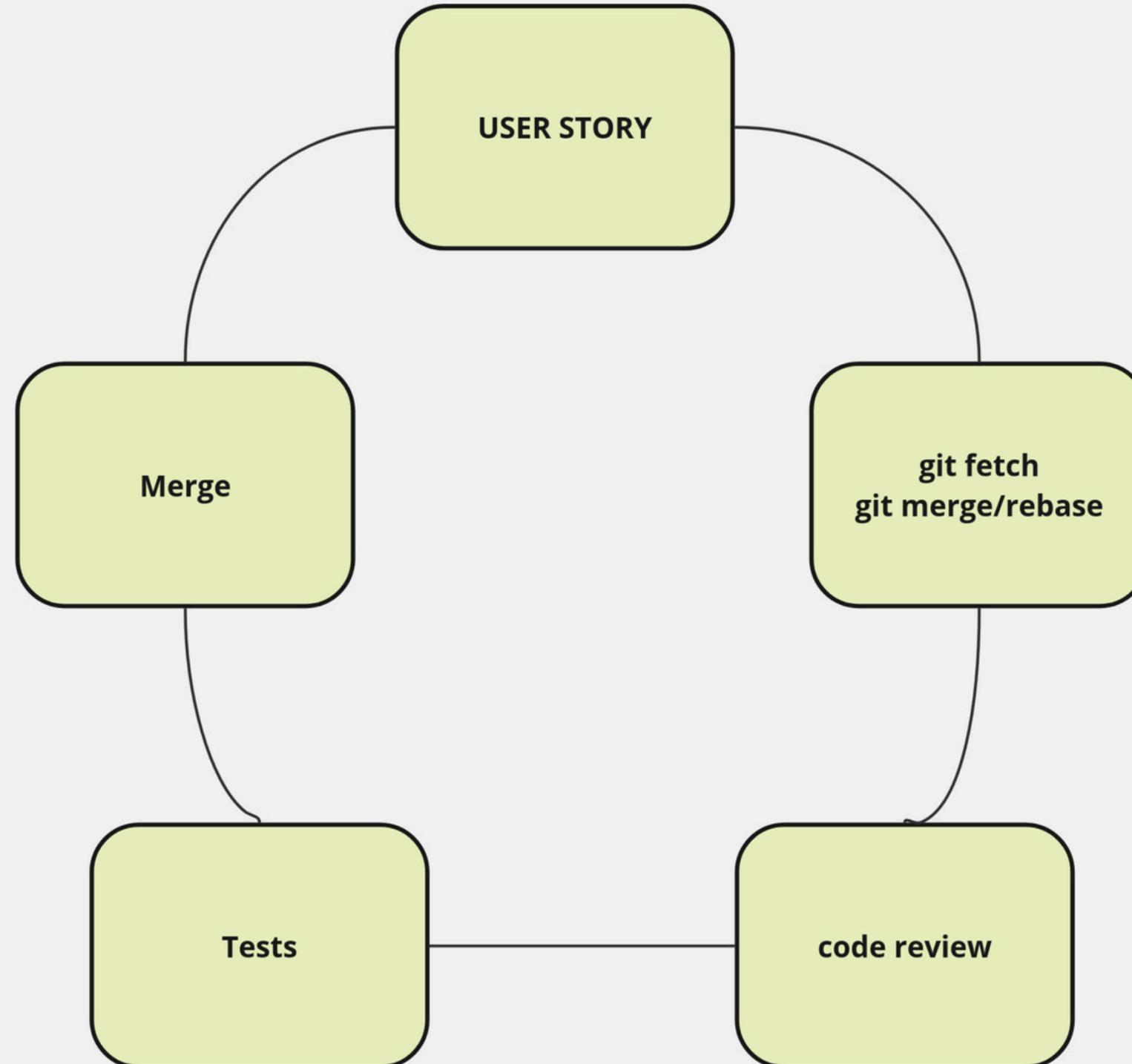
- Dépôt github
- Processus de développement



Dépôt github



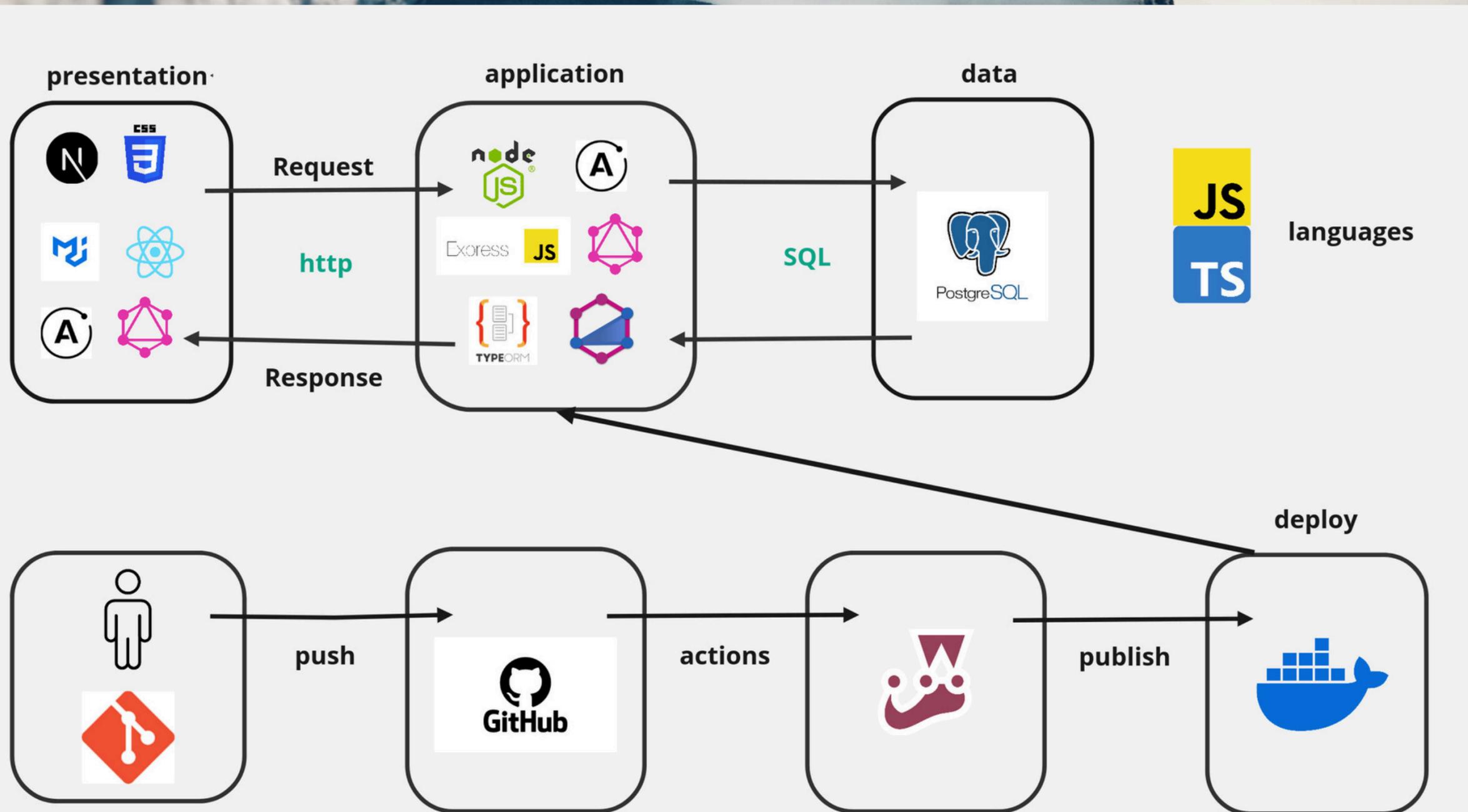
Processus de développement



Installation du projet

- Architecture
- Les technologies
- Dockerisation

SCHÉMA DE L'APPLICATION



Dockerisation

Dockerfile X

2311-wns-jaune-oros-front > Dockerfile

```
1 FROM node:lts-alpine
2
3 RUN mkdir /app
4 WORKDIR /app
5
6 COPY package.json package.json
7
8 RUN npm i
9 RUN apk --update --no-cache add curl
10
11
12 COPY src src
13 COPY public public
14
15 COPY tsconfig.json tsconfig.json
16
17 COPY codegen.ts codegen.ts
18 COPY next.config.js next.config.js
19
20 CMD npm run dev && npm run generate
```

Dockerfile X

2311-wns-jaune-oros-back > Dockerfile

```
1 FROM node:lts-alpine
2
3 RUN mkdir /app
4 WORKDIR /app
5 COPY package.json .
6 RUN npm i
7
8 RUN apk --update --no-cache add curl
9
10 COPY src src
11 COPY tsconfig.json .
12
13 CMD npm run start
```

Dockerisation

```
docker-compose.yml M X
2311-wns-jaune-oros-front > docker-compose.yml
1  services:
2    backend:
3      build: ../2311-wns-jaune-oros-back
4      ports:
5        - 4005:4000
6      volumes:
7        - "../2311-wns-jaune-oros-back/src:/app/src"
8      env_file:
9        - ../2311-wns-jaune-oros-back/db.env
10       - ../2311-wns-jaune-oros-back/.env
11     healthcheck:
12       test:
13         - CMD-SHELL
14         - >
15           curl
16           -f http://backend:4000/graphql?query=%7B__typename%7D
17           -H 'Apollo-Require-Preflight: true'
18           || exit 1
19       interval: 10s
20       timeout: 30s
21       retries: 5
22     depends_on:
23       db:
24         condition: service_healthy
```

Dockerisation

```
26 frontend:
27   build: ./
28   ports:
29     - 3000:3000
30   volumes:
31     - "./src:/app/src"
32   depends_on:
33     backend:
34       condition: service_healthy
35   env_file:
36     - ./env
37   develop:
38     watch:
39       - action: sync+restart
40         path: ./src
41         target: /app/src
```

```
43 db:
44   image: postgres
45   restart: always
46   env_file:
47     - ../2311-wns-jaune-oros-back/db.env
48   ports:
49     - 5437:5432
50   healthcheck:
51     test: ["CMD-SHELL", "pg_isready -d oros -U utilisateur"]
52     interval: 10s
53     timeout: 5s
54     retries: 5
55   volumes:
56     - orosdatabase:/var/lib/postgresql/data
57
58   adminer:
59     image: adminer
60     restart: always
61     ports:
62       - 8087:8080
63
64   volumes:
65     orosdatabase:
```

Créer un matériel

- Utilisateur authentifié
- Requête
- Création d'un matériel

Créer un utilisateur

```
TS authChecker.ts X
2311-wns-jaune-oros-back > src > lib > TS authChecker.ts > customAuthChecker
1 import { AuthChecker } from "type-graphql";
2 import { MyContext } from "..";
3
4 export const customAuthChecker: AuthChecker<MyContext> = (
5   { context },
6   roles
7 ) => {
8   if (context.user) {
9     //si l'utilisateur est connecté
10    //vérifier que le user a le rôle demandé si le tableau de rôles a une longueur > 1
11    if (roles.length > 0) { // si un rôle est indiqué au décorateur
12      if (roles.includes(context.user.role)) { //et que le user a le rôle parmi ce tableau
13        return true; //on laisse passer
14      } else { //sinon
15        return false; //on bloque
16      }
17    }
18    return true; //si le user est connecté et qu'on a pas spécifié de rôle, on laisse passer
19  }
20  return false; //si le user n'est pas connecté quand on utilise le décorateur, on bloque
21 };
```

```
TS user.resolver.ts X
2311-wns-jaune-oros-back > src > resolvers > TS user.resolver.ts > UserResolver > logout
30 @Query(() => Message)
31 async login(@Arg('infos') infos: InputLogin, @Ctx() ctx: MyContext) {
32   const user = await new UserServices().findUserByEmail(infos.email);
33   if (!user) {
34     throw new Error('Vérify your login information');
35   }
36
37   const isValid = await argon2.verify(user.password, infos.password);
38   const m = new Message();
39   if (isValid) {
40     const token = await new SignJWT({ email: user.email, role: user.role })
41       .setProtectedHeader({ alg: 'HS256', typ: 'jwt' })
42       .setExpirationTime('2h')
43       .sign(new TextEncoder().encode(`${process.env.SECRET_KEY}`));
44
45     const cookies = new Cookies(ctx.req, ctx.res);
46     cookies.set('token', token, { httpOnly: true });
47
48     // Créer ou récupérer la session
49     const sessionServices = new SessionServices();
50     const session = await sessionServices.createOrGetSession(ctx, user);
51
52     console.log(token);
53     console.log(cookies);
54     console.log('babakens:', session.cart);
55
56     m.message = 'Welcome!';
57     m.success = true;
58   } else {
59     m.message = 'Verify your login information';
60     m.success = false;
61     throw new Error('Vérify your login information');
62   }
63   return m;
64 }
```

Créer un matériel

```
@InputType()
export class InputCreateMaterial {
  @Field()
  name: string;

  @Field()
  description: string;

  @Field()
  image: string;

  @Field()
  initial_stock: number;

  @Field()
  price: number;

  @Field({ nullable: true })
  category: PartialCategoryInput;

  @Field()
  slug: string;
}
```

```
@Authorized('ADMIN')
@Mutation(() => Material)
async createMaterial(@Arg('infos') infos: InputCreateMaterial) {
  const result: Material = await new MaterialServices().create(infos);
  console.log('RESULT', result);
  return result;
}
```

```
async create(data: InputCreateMaterial) {
  const category = await new CategoryServices().findById(data.category.id);
  const newMaterial = await this.db.create({ ...data, category });
  return await this.db.save(newMaterial);
}
```

Test de création de matériel

The screenshot displays the GraphQL Playground interface. At the top, the URL is `http://localhost:4005/graphq` and the environment is set to `SANDBOX`. The active tab is `CreateMaterial`. The left sidebar shows the `Documentation` pane with a tree view for `Root > Query` and a list of available queries such as `findCategoryById`, `findMaterialById`, and `listCategories`.

The central editor shows the following GraphQL operation:

```
1 mutation CreateMaterial($infos: InputCreateMaterial!) {
2   createMaterial(infos: $infos) {
3     id
4     image
5     initial_stock
6     name
7     price
8     description
9     category {
10    | id
11    | }
12    slug
13  }
14 }
```

Below the editor, the `Variables` tab is active, showing the input variables in JSON:

```
1 {
2   "infos": {
3     "name": "AR05 Race - Casque de cyclisme",
4     "image": "https://www.bfgcdn.com/600_600_90/851-1421-0811/oakley-aro5-race-casque-de-cyclisme.jpg",
5     "category": {
6       "id": 3
7     },
8     "description": "L'AR05 Race a été conçu pour des performances optimales et une respirabilité maximale sur la route. Fabriqué dans un souci de résultats, l'AR05 Race présente un design aérodynamique et ventilé qui augmente considérablement le flux d'air à travers 19 orifices de ventilation. Plus léger que son prédécesseur, l'AR05, l'AR05 Race vous fournit l'avantage nécessaire pour dépasser le peloton. Alliant des avancées techniques supplémentaires sur le plan de l'intégration optique, du confort et de la protection, l'AR05 Race est prêt à aller vite.",
9     "initial_stock": 20,
10    "price": 15
11  }
12 }
```

The right-hand pane shows the response status: `200` with a response time of `72.0ms` and a size of `873B`. The response is displayed in JSON:

```
{
  "data": {
    "createMaterial": {
      "id": "83ab49a0-cf65-47da-b011-c391b57a8524",
      "image": "https://www.bfgcdn.com/600_600_90/851-1421-0811/oakley-aro5-race-casque-de-cyclisme.jpg",
      "initial_stock": 20,
      "name": "AR05 Race - Casque de cyclisme",
      "price": 15,
      "description": "L'AR05 Race a été conçu pour des performances optimales et une respirabilité maximale sur la route. Fabriqué dans un souci de résultats, l'AR05 Race présente un design aérodynamique et ventilé qui augmente considérablement le flux d'air à travers 19 orifices de ventilation. Plus léger que son prédécesseur, l'AR05, l'AR05 Race vous fournit l'avantage nécessaire pour dépasser le peloton. Alliant des avancées techniques supplémentaires sur le plan de l'intégration optique, du confort et de la protection, l'AR05 Race est prêt à aller vite.",
      "category": {
        "id": "3"
      },
      "slug": "casque"
    }
  }
}
```

Gestion du formulaire

```
TS materials.mutations.ts X
2311-wns-jaune-oros-front > src > requests > mutations > TS materials.mutations.ts > ...
18 `;
19
20 export const CREATE_MATERIAL = gql`
21   mutation CreateMaterial($infos: InputCreateMaterial!)
22     createMaterial(infos: $infos) {
23     id
24     name
25     category {
26       id
27     }
28     description
29     image
30     initial_stock
31     price
32     slug
33   }
34 }
35 `;
```

```
TS newProduct.tsx X
2311-wns-jaune-oros-front > src > pages > admin > stocks > TS newProduct.tsx > ...
1 import { useState } from "react";
2 import { TextField, Button, Typography } from "@mui/material";
3 import { useMutation } from "@apollo/client";
4 import { CREATE_MATERIAL } from "@requests/mutations/materials.mutations";
5 import { useRouter } from "next/router";
6 import styles from "../index.module.css";
7 import adminProtection from "@hoc/adminProtection";
8
9 const NewProduct = () => {
10   const router = useRouter();
11
12   const [formData, setFormData] = useState({
13     name: "",
14     description: "",
15     price: "",
16     initial_stock: "",
17     slug: "",
18     category: "",
19     image: "",
20   });
21
22   const [createMaterial, { loading, error }] = useMutation(CREATE_MATERIAL, {
23     onCompleted: () => {
24       router.push("/admin/stocks");
25     },
26     onError: (err) => {
27       console.error("Erreur lors de la création du produit", err);
28     },
29   });
```

Création de matériel par un admin



STOCKS

RÉSERVATIONS

[Retour espace client](#)



Ajouter un produit

Nom
SET SLACKLINE BANANA LAMA - PROTECTION D'ARBRE - GIBBON - 25M

Description
La slack BANANA LAMA est le compromis idéal entre CLASSIC et FLOWLINE – une slack de 3,5cm de large. Elle combine les avantages des deux types de slack. Débutants ou expérimentés, tous y trouveront satisfaction et du plaisir. L'impression spécifique sur la ligne offre une excellente adhérence pour débiter et s'améliorer. Durable et Fiable : Conçu pour une utilisation répétée dans divers environnements extérieurs, des jardins personnels aux terrains d'entraînement scolaires.

Prix
20

Stock initial
8

Slug
set-slackline

ID de la catégorie
1

URL de l'image
https://www.bfgcdn.com/600_600_90/330

[AJOUTER](#)

Tous les produits

Trier par

Prix

Rechercher un matériel



SET SLACKLINE BANANA LAMA -
PROTECTION D'ARBRE - GIBBON -
25M

Stocks: 8

Prix: 20 €

[Modifier produit](#)



< 1 2 >

Préparer le déploiement

- Environnement de tests
- Automatisation des tests
- Script de déploiement

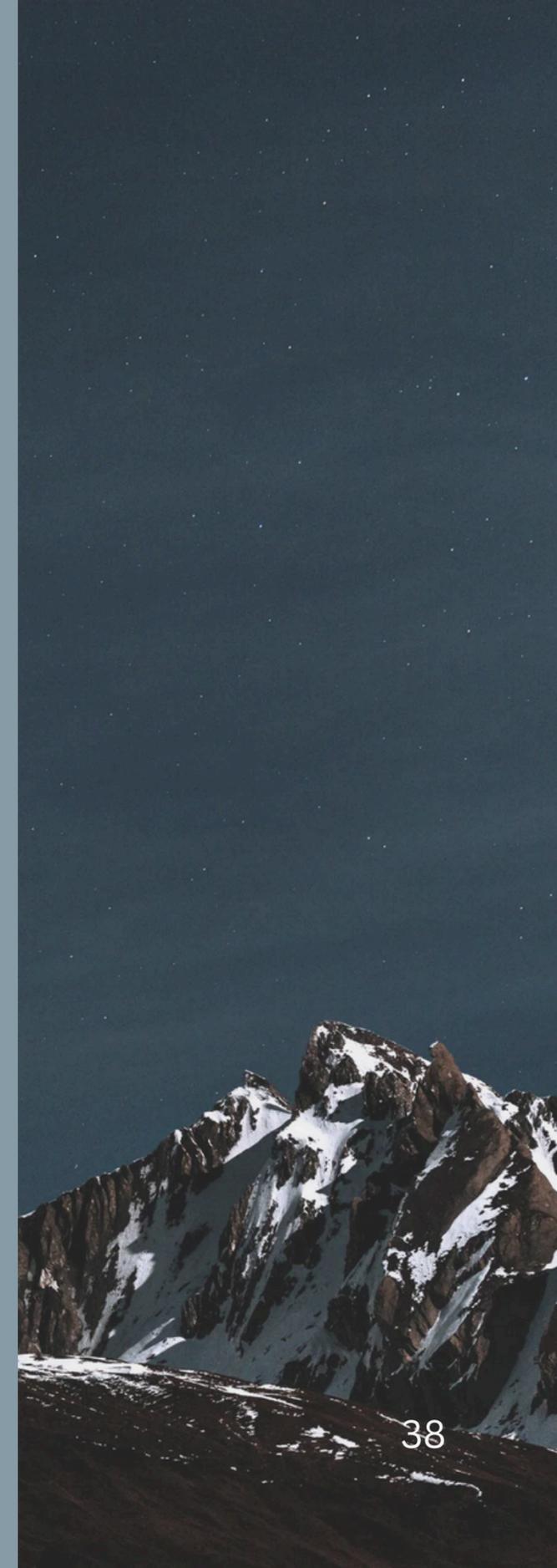
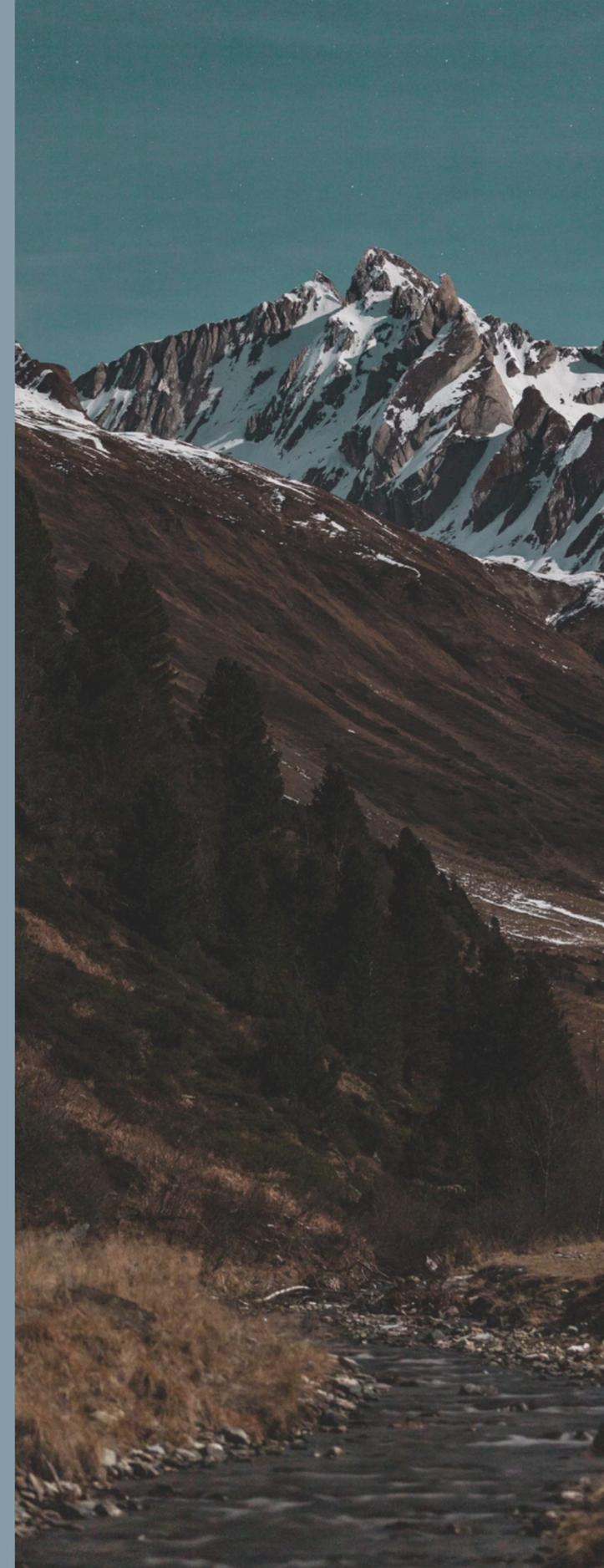
Intégration

```
14 const materialsData: Material[] = [
15   {
16     id: '1',
17     name: 'My Material 1',
18     category: {
19       id: 1,
20       name: 'Category 1',
21       materials: [],
22       slug: 'category-1',
23     },
24     description: 'Description 1',
25     image: 'image1.jpg',
26     initial_stock: 10,
27     price: 10.5,
28     slug: 'my-material-1',
29   },
30   {
31     id: '2',
32     name: 'My Material 2',
33     category: {
34       id: 2,
35       name: 'Category 2',
36       materials: [],
37       slug: 'category-2',
38     },
39     description: 'Description 2',
40     image: 'image2.jpg',
41     initial_stock: 20,
42     price: 20.5,
43     slug: 'my-material-2',
44   },
45 ];
46
47 export const LIST_MATERIALS = `#graphql
48 query ListMaterials {
49   listMaterials {
50     id
51   }
52 }
53 `;
```

```
54 type ResponseData = {
55   material: Material[];
56 };
57
58 let server: ApolloServer;
59
60 const baseSchema = buildSchemaSync({
61   resolvers: [MaterialResolver],
62   authChecker: () => true,
63 });
64
65 const schemaString = printSchema(baseSchema);
66 const schema = makeExecutableSchema({ typeDefs: schemaString });
67
68 const resolvers = (store: IMockStore) => ({
69   //resolvers est une fonction qui reçoit le store en argument!
70   Query: {
71     listMaterials() {
72       return store.get('Query', 'ROOT', 'listMaterials');
73     },
74   },
75 });
76
77 beforeAll(async () => {
78   const store = createMockStore({ schema });
79   server = new ApolloServer({
80     schema: addMocksToSchema({
81       schema: baseSchema,
82       store,
83       resolvers,
84     }),
85   });
86
87   store.set('Query', 'ROOT', 'listMaterials', materialsData);
88 });
89
90 describe('Test sur les matériels', () => {
91   it('Récupération des matériels depuis le store', async () => {
92     const response = await server.executeOperation<ResponseData>({
93       query: LIST_MATERIALS,
94     });
95   });
96 });
```

Exécution des tests

```
! back-tests.yml M X
2311-wns-jaune-oros-back > .github > workflows > ! back-tests.yml
1 name: jest-and-docker-ci
2
3 on: push
4
5 jobs:
6   test-back:
7     runs-on: ubuntu-latest
8
9     steps:
10      - name: Check out code
11        uses: actions/checkout@v2
12      - name: run tests
13        run: npm i && npm run test-ci
14
15   docker:
16     needs: test-back
17     if: github.ref == 'refs/heads/main'
18     runs-on: ubuntu-latest
19     steps:
20      - name: Set up QEMU
21        uses: docker/setup-qemu-action@v2
22      - name: Set up Docker Buildx
23        uses: docker/setup-buildx-action@v2
24      - name: Login to Docker Hub
25        uses: docker/login-action@v2
26      with:
27        username: ${{ secrets.DOCKERHUB_USERNAME }}
28        password: ${{ secrets.DOCKERHUB_TOKEN }}
29      - name: Build and push
30        uses: docker/build-push-action@v4
31      with:
32        push: true
33        context: '{{defaultContext}}'
34        tags: ${{ secrets.DOCKERHUB_USERNAME }}/wns-jaune-oros-back:latest
```



Déploiement

```
// wns_student@wns-project-server:~/apps/oros/staging$  
  
services:  
  backend:  
    image: orosgroup/wns-jaune-oros-back  
    expose:  
      - 4000  
    env_file:  
      - ./db.env  
    healthcheck:  
      test:  
        - CMD-SHELL  
        - >  
          curl -f http://backend:4000/graphql?query=%7B__typename%7D  
            -H 'Apollo-Require-Preflight: true'  
            || exit 1  
      interval: 10s  
      timeout: 30s  
      retries: 5  
    depends_on:  
      db:  
        condition: service_healthy  
  
  frontend:  
    image: orosgroup/wns-jaune-oros-front  
    expose:  
      - 3000  
    depends_on:  
      backend:
```

Déploiement

```
    condition: service_healthy
env_file:
  - ./env

db:
image: postgres
restart: always
env_file:
  - ./db.env
expose:
  - 5432
healthcheck:
  test: ["CMD-SHELL", "pg_isready -d oros -U utilisateur"]
  interval: 10s
  timeout: 5s
  retries: 5
volumes:
  - orosdatabase:/var/lib/postgresql/data

nginx:
image: nginx:1.21.3
depends_on:
  - backend
  - frontend
restart: always
ports:
  - ${GATEWAY_PORT:-8001}:80
volumes:
  - ./nginx.conf:/etc/nginx/nginx.conf
  - ./logs:/var/log/nginx

adminer:
image: adminer
restart: always
ports:
  - 8087:8080
```

Diagramme CI/CD

