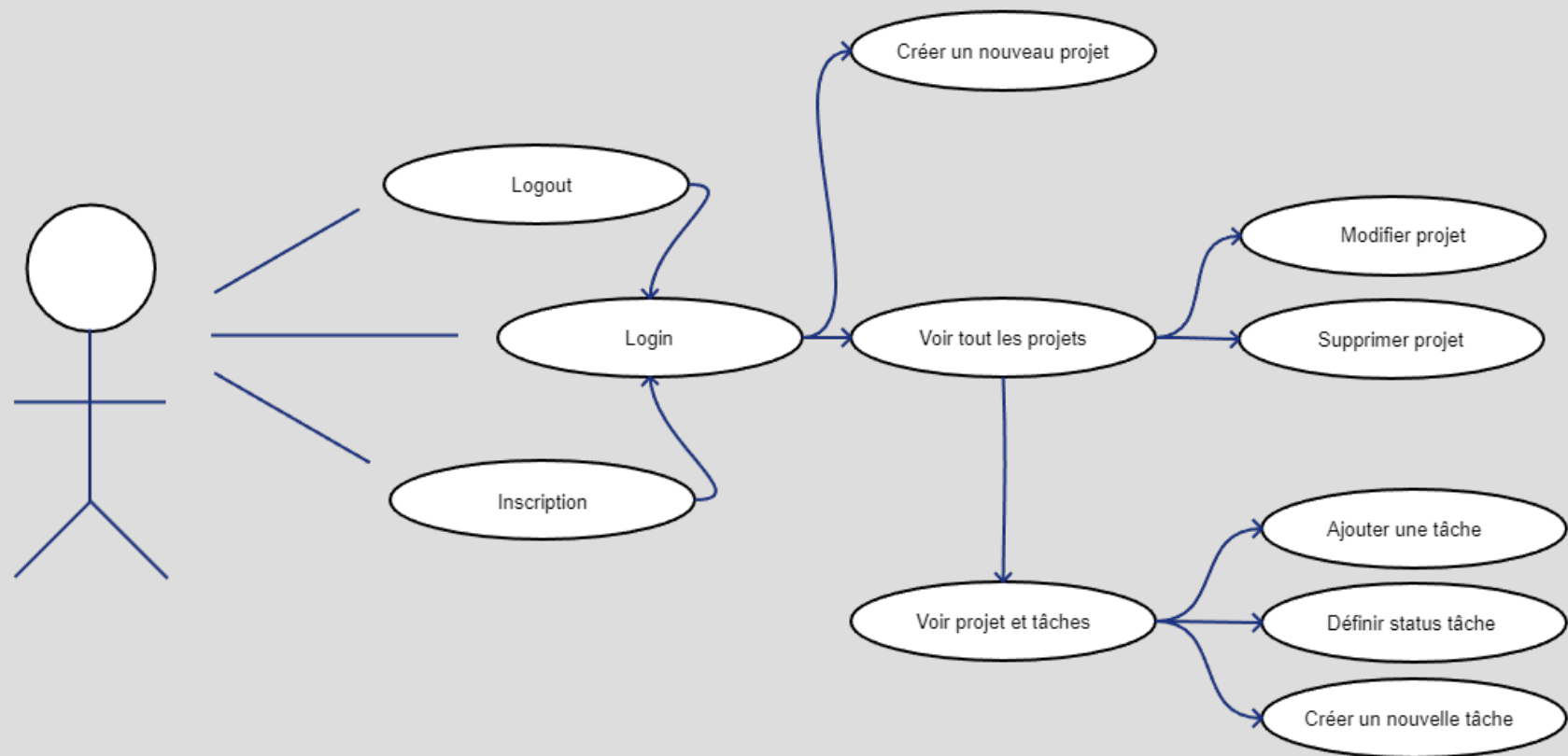


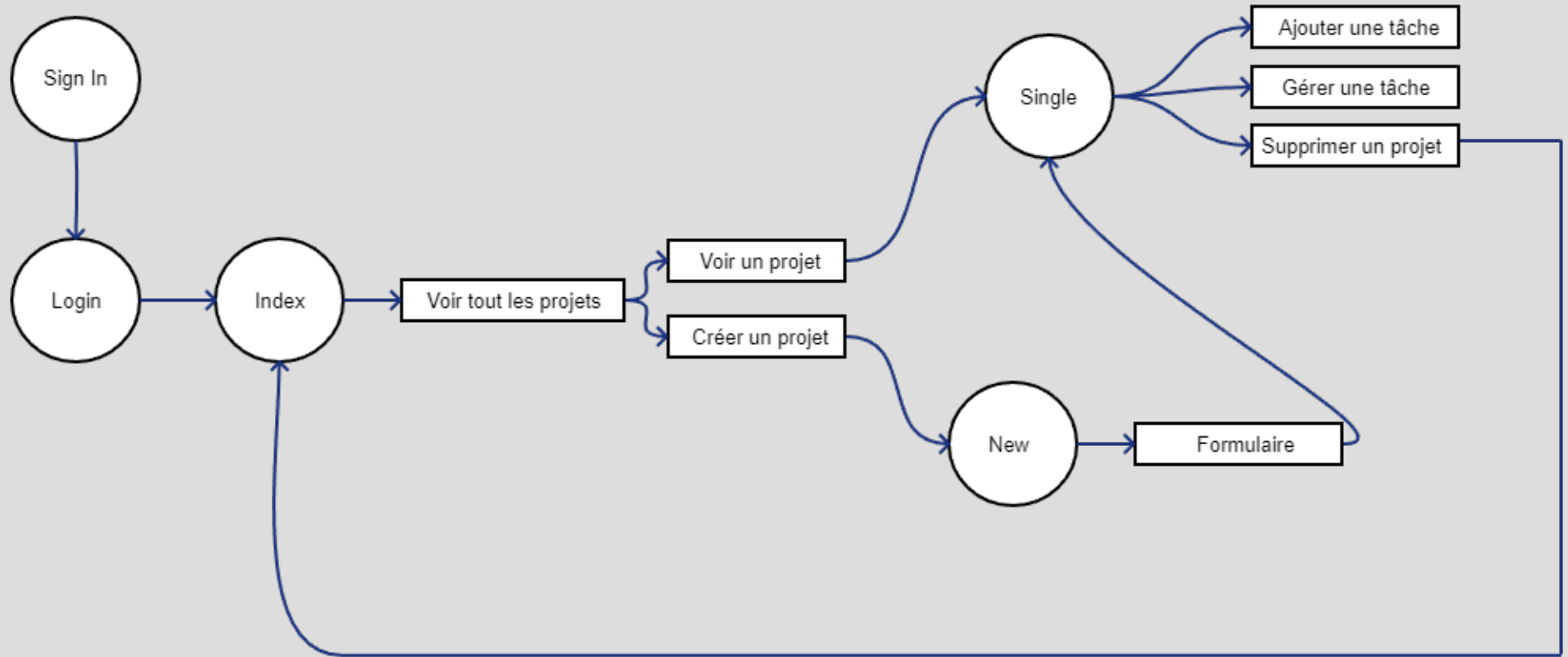
Batiment'Aleau

Afin d'optimiser son espace de travail et de rendre accessible les différents projets associés aux clients avec qui elle traite, la société « Batiment'Aleau » a commandé une application de gestion de projets. Celle-ci doit pouvoir enregistrer de nouveaux projets, avec des tâches associés, trier par ordre de date fin. Elle devra être responsive afin d'être utilisable sur une surface mobile et accessible via des identifiants de connexion de type email/mot de passe.

UseCase



Arborescence Fonctionnelle



Base de données

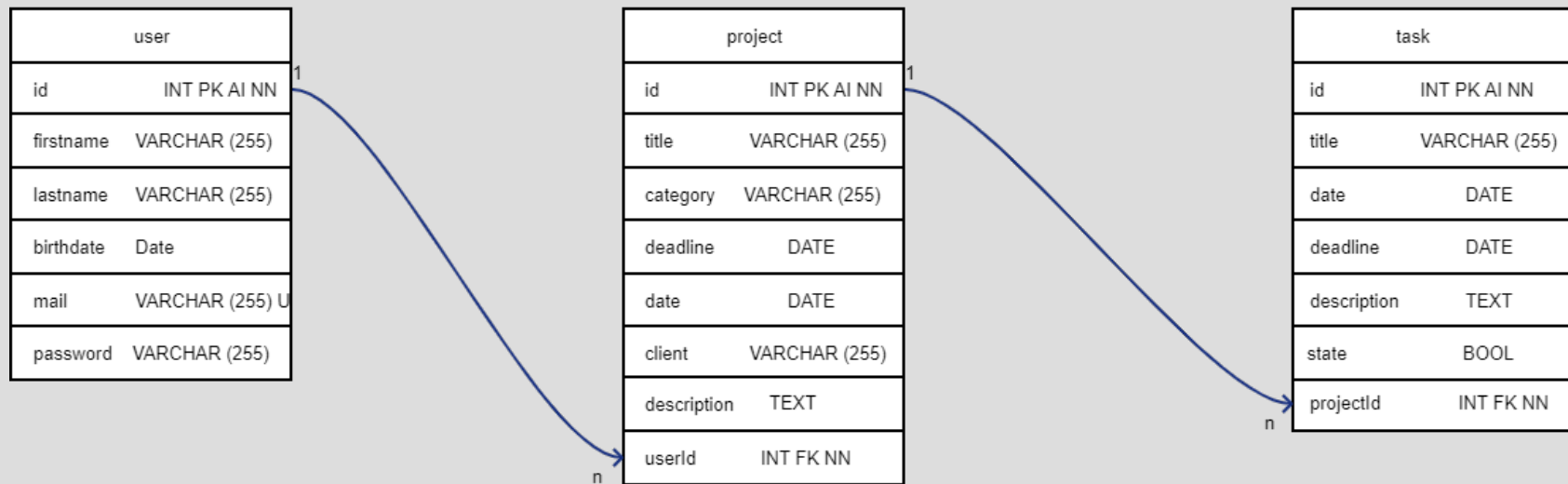
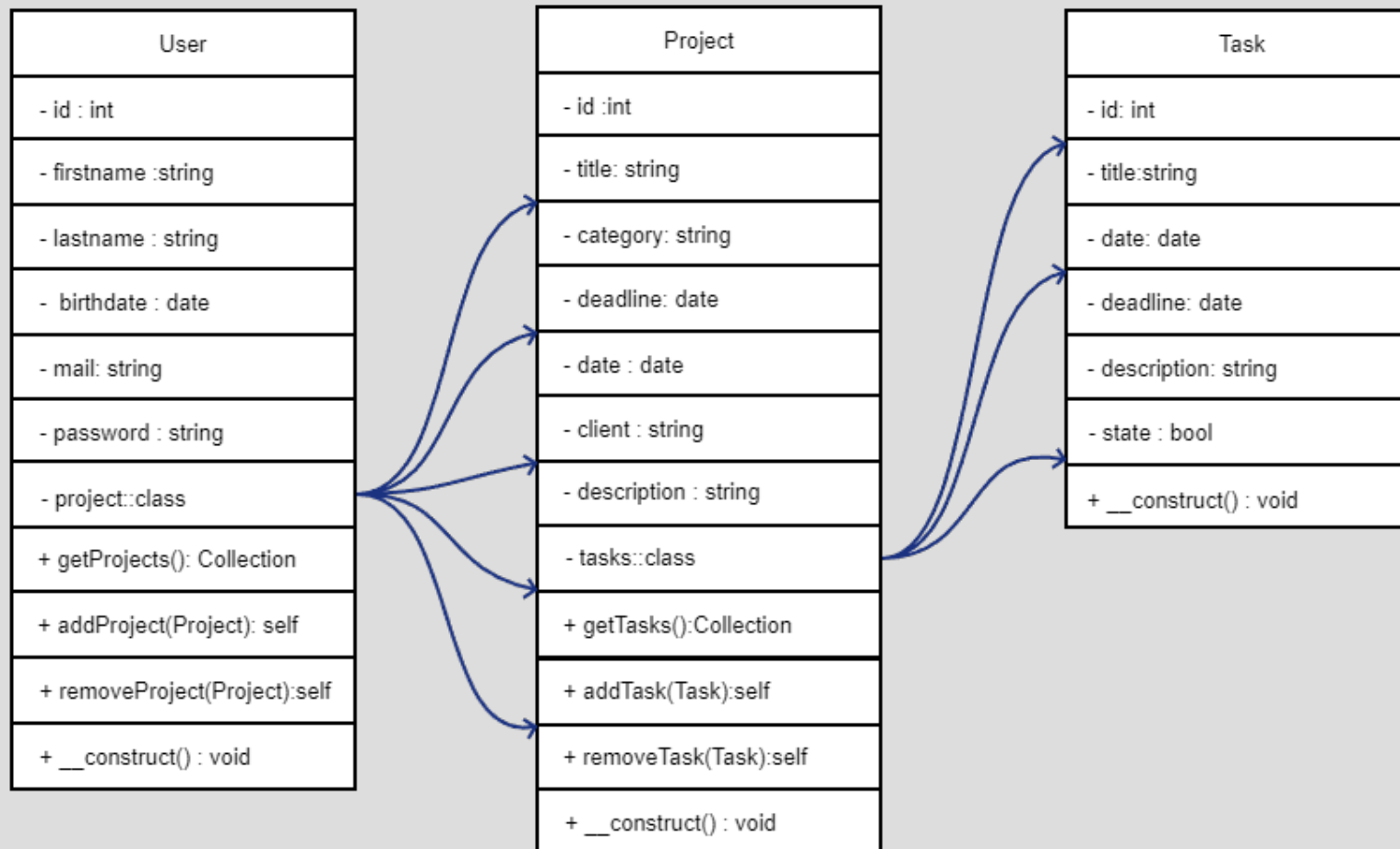
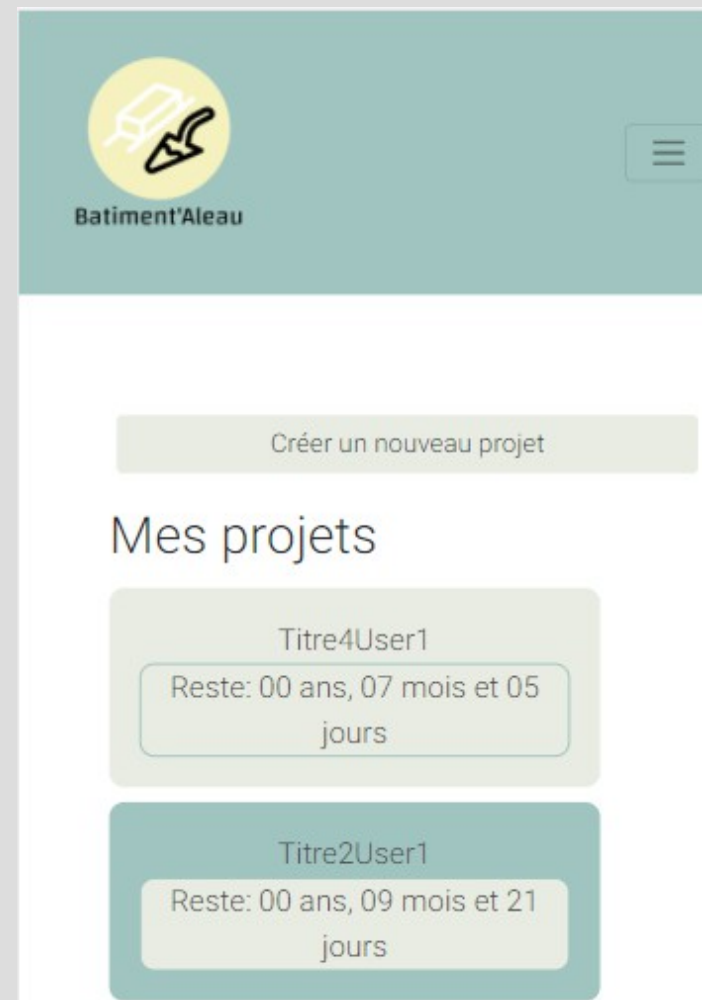
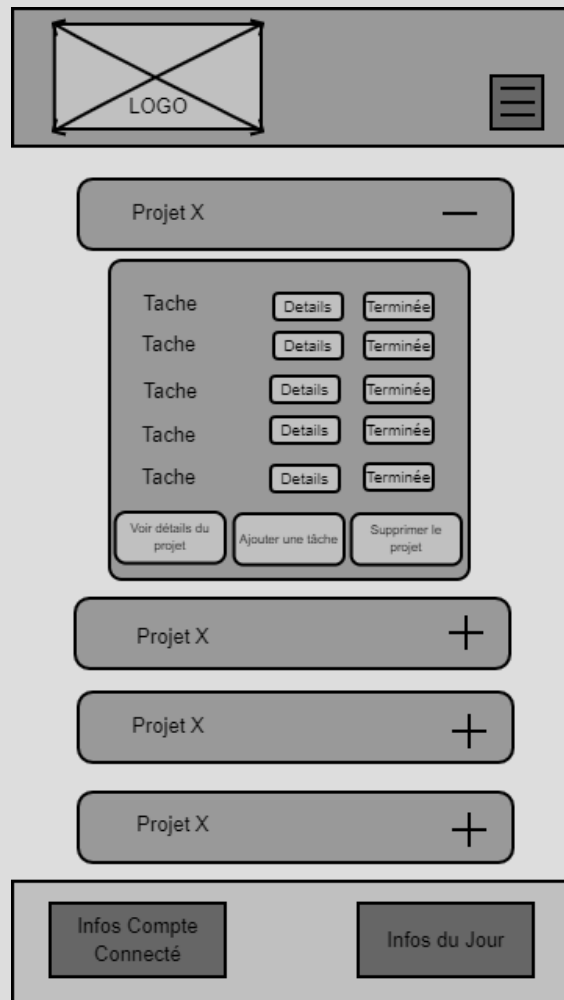


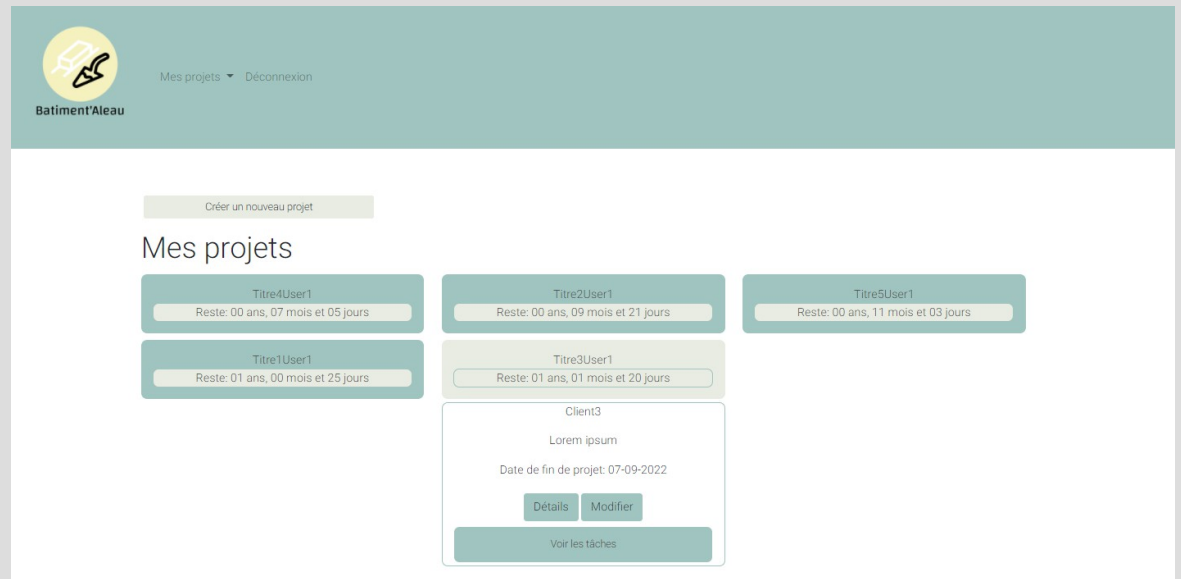
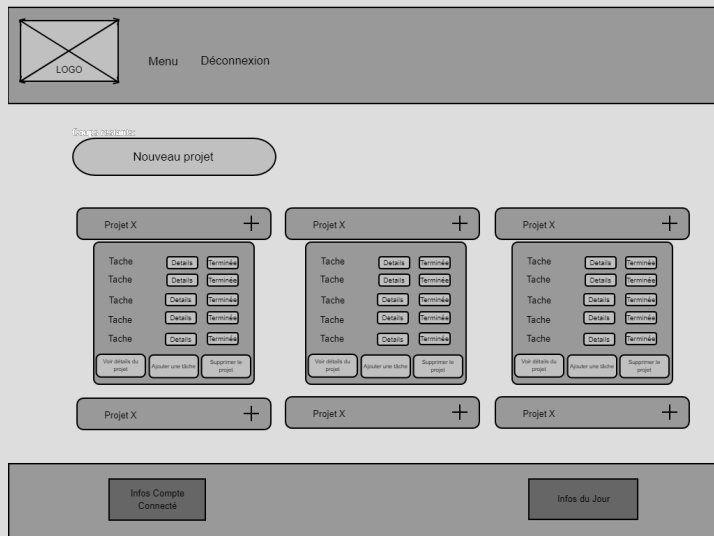
Diagramme de classes



Version Mobile



Version Ordinateur



Affichage des projet sur l'index

```
public function findProjects(User $user){  
    return $this->createQueryBuilder('p')  
        ->innerJoin('p.user', 'u')  
        ->addSelect('u')  
        ->andWhere('p.user= :val')  
        ->orderBy('p.deadline', 'DESC')  
        ->setParameter('val', $user)  
        ->getQuery()  
        ->getResult()  
    ;  
}
```

```
#[Route('/project')]  
class ProjectController extends AbstractController  
{  
    #[Route('/', name: 'project_index', methods: ['GET'])]  
    public function index(ProjectRepository $projectRepository): Response  
    {  
        $user=$this->getUser();  
        $projects = $projectRepository->findProjects($user);  
        return $this->render('project/index.html.twig', [  
            'projects' => $projects,  
        ]);  
    }  
}
```


Mises en formes et Animations

JavaScript et CSS

```
let acc = document.getElementsByClassName("accordion");
let i;

for (i = 0; i < acc.length; i++) {
  acc[i].addEventListener("click", function() {
    this.classList.toggle("active");

    let panel = this.nextElementSibling;
    if (panel.style.display === "block") {
      panel.style.display = "none";
    } else {
      panel.style.display = "block";
    }
  });
}

function date(){
  let date= new Date();
  let da = new Intl.DateTimeFormat('en', { day: '2-digit' }).format(date);
  let mo = new Intl.DateTimeFormat('fr', { month: 'short' }).format(date);
  let ye = new Intl.DateTimeFormat('en', { year: 'numeric' }).format(date);
  let newDate="Nous sommes le "+da+" "+mo+" "+ye+" ";
  document.getElementById('date').innerHTML=newDate;
}
date();
```

```
/* For JS accordion animation */
.accordion {
  border-radius: 0.5em;
  color: #444;
  cursor: pointer;
  padding: 1em;
  text-align: center;
  border: none;
  outline: none;
  transition: 0.4s;
  width:100%
}

.active, .accordion:hover {
  background-color: #E6EBE0;
}

.panel {
  padding: 0 1em;
  background-color: white;
  display: none;
  overflow: hidden;
}
```

Documentations

Syntax

```
new Intl.DateTimeFormat()  
new Intl.DateTimeFormat(locales)  
new Intl.DateTimeFormat(locales, options)
```

Examples

Using DateTimeFormat

In basic use without specifying a locale, `DateTimeFormat` uses the default locale and default options.

```
var date = new Date(Date.UTC(2012, 11, 20, 3, 0, 0));  
  
// toLocaleString without arguments depends on the implementation,  
// the default locale, and the default time zone  
console.log(new Intl.DateTimeFormat().format(date));  
// → "12/19/2012" if run with en-US locale (language) and time zone America/Los_Angeles (UTC-0800)
```