



Projet JEE

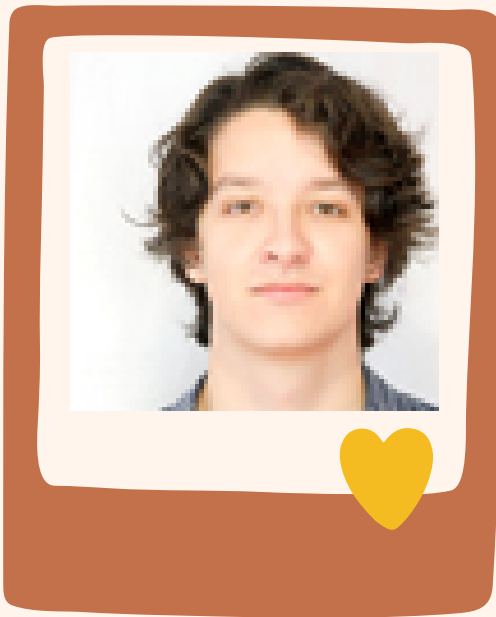
Matthieu, Bastien, Leo, Titouan

START






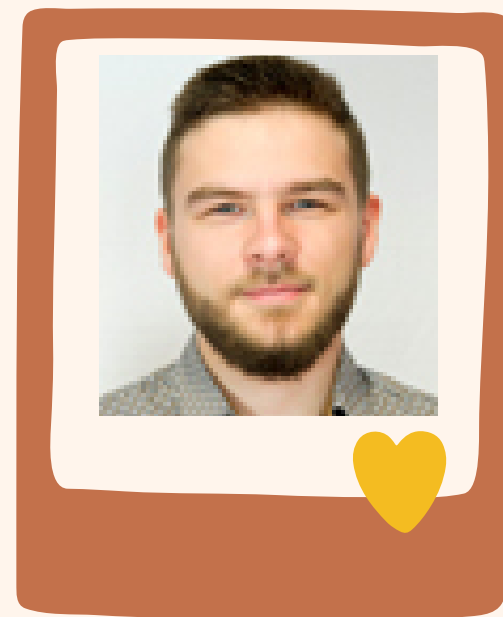
Notre équipe



Matthieu

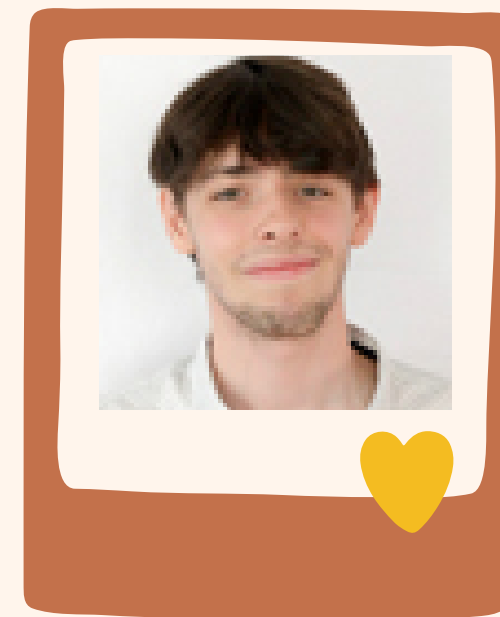


Sécurité/auth
Front
Backend



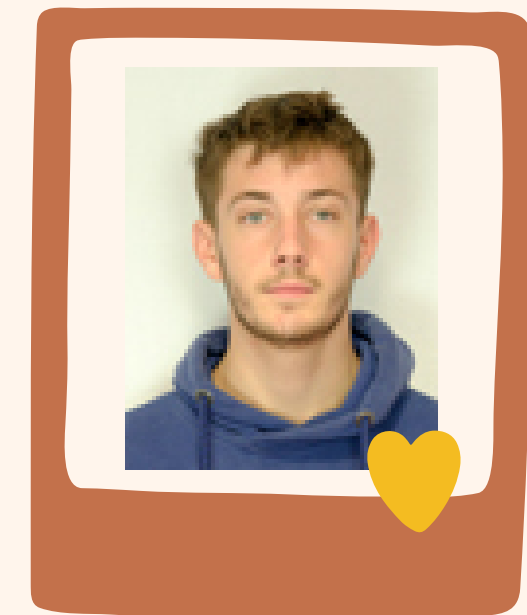
Bastien

Front



Leo

Backend
Controler/services



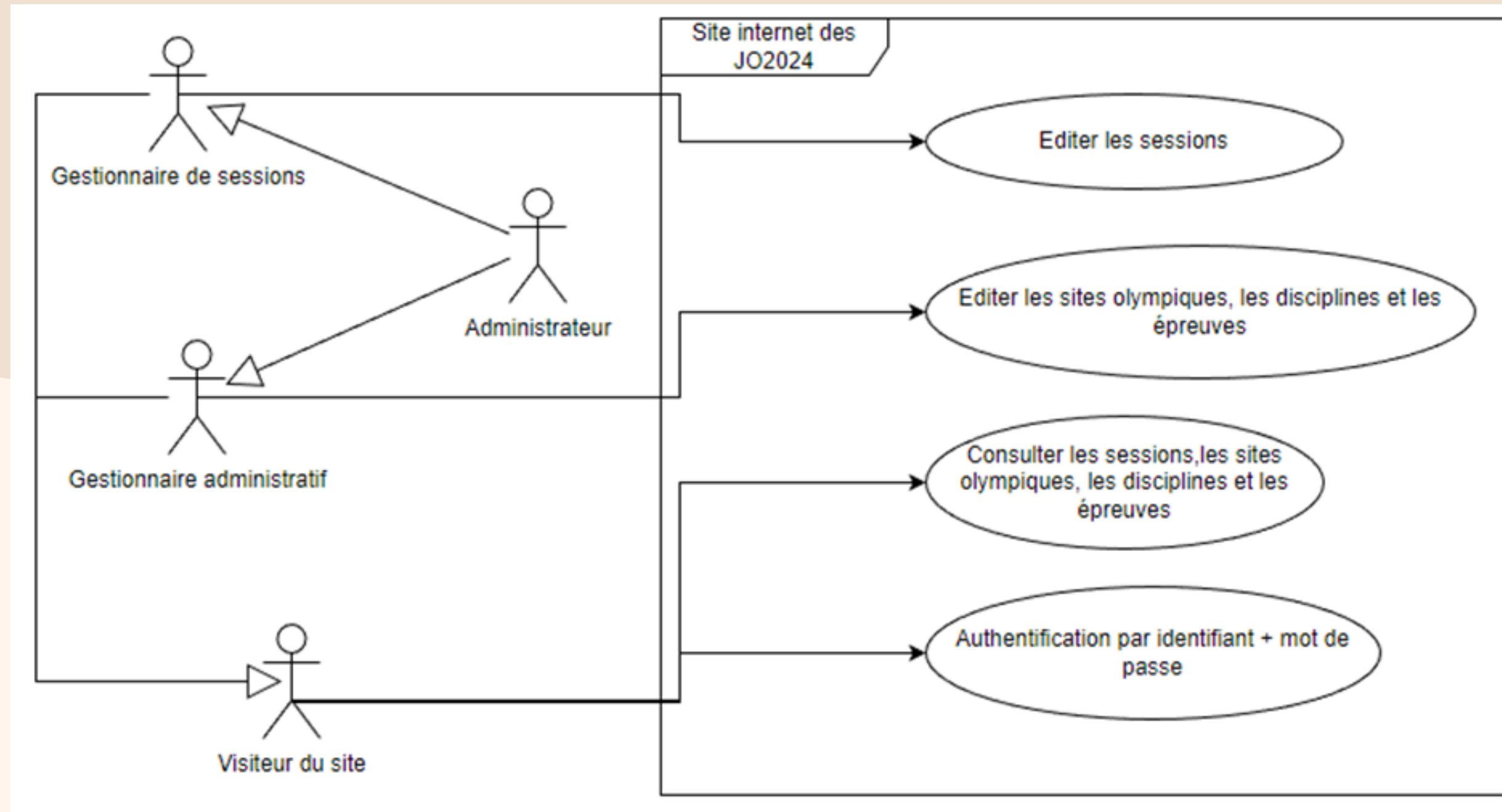
Titouan

Backend
Controler
BDD

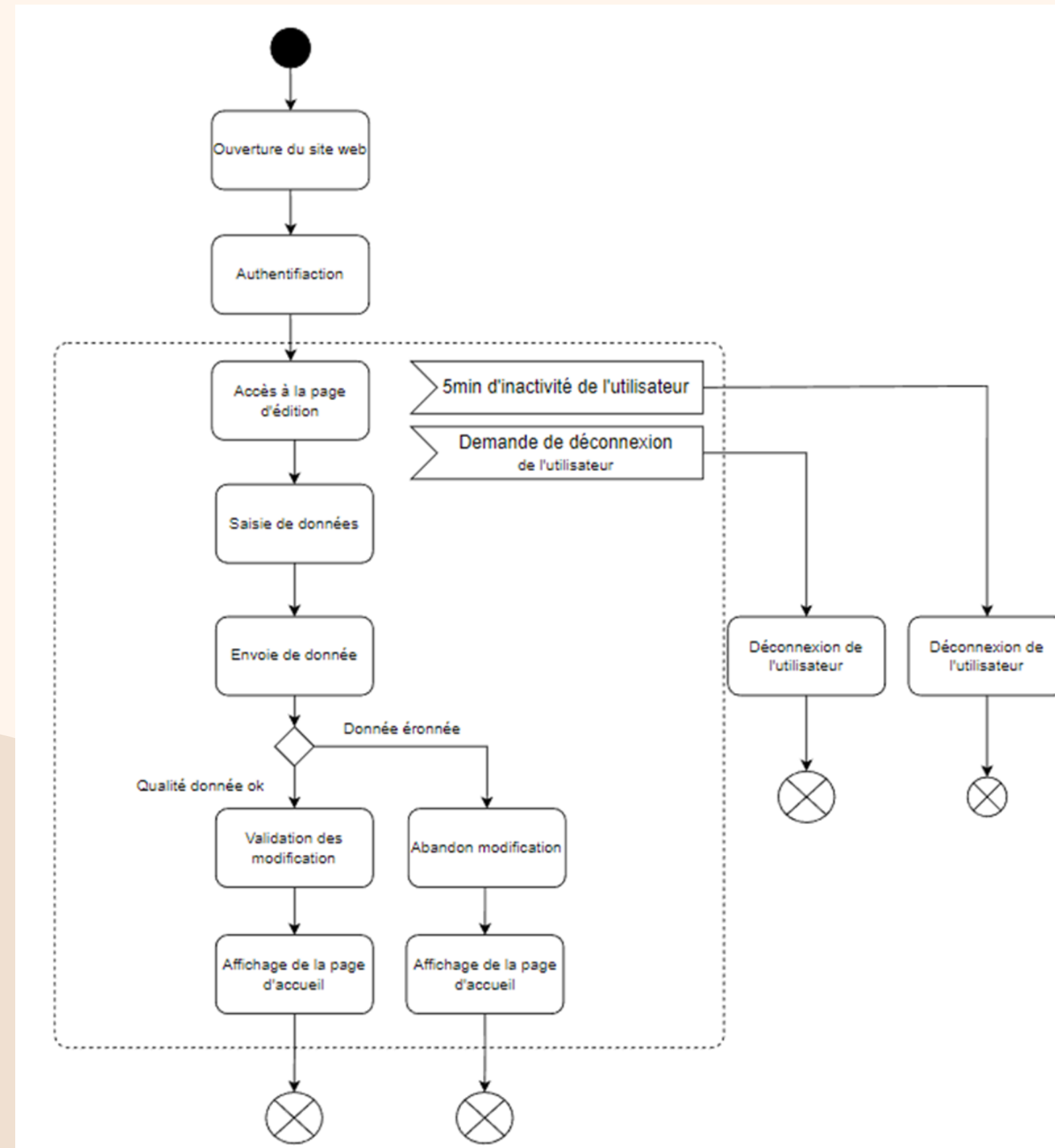


UML

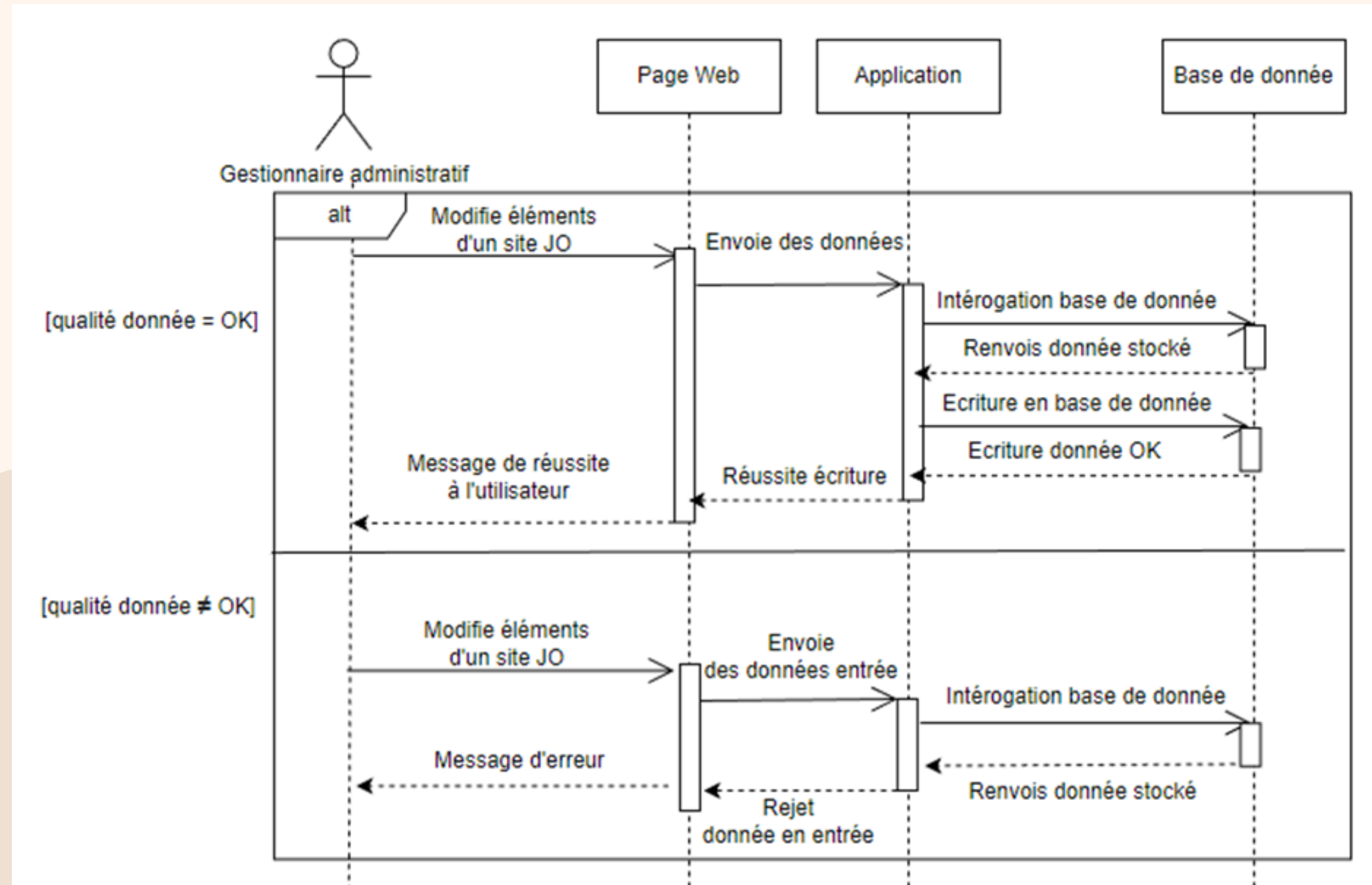
UML: Diagramme des UC



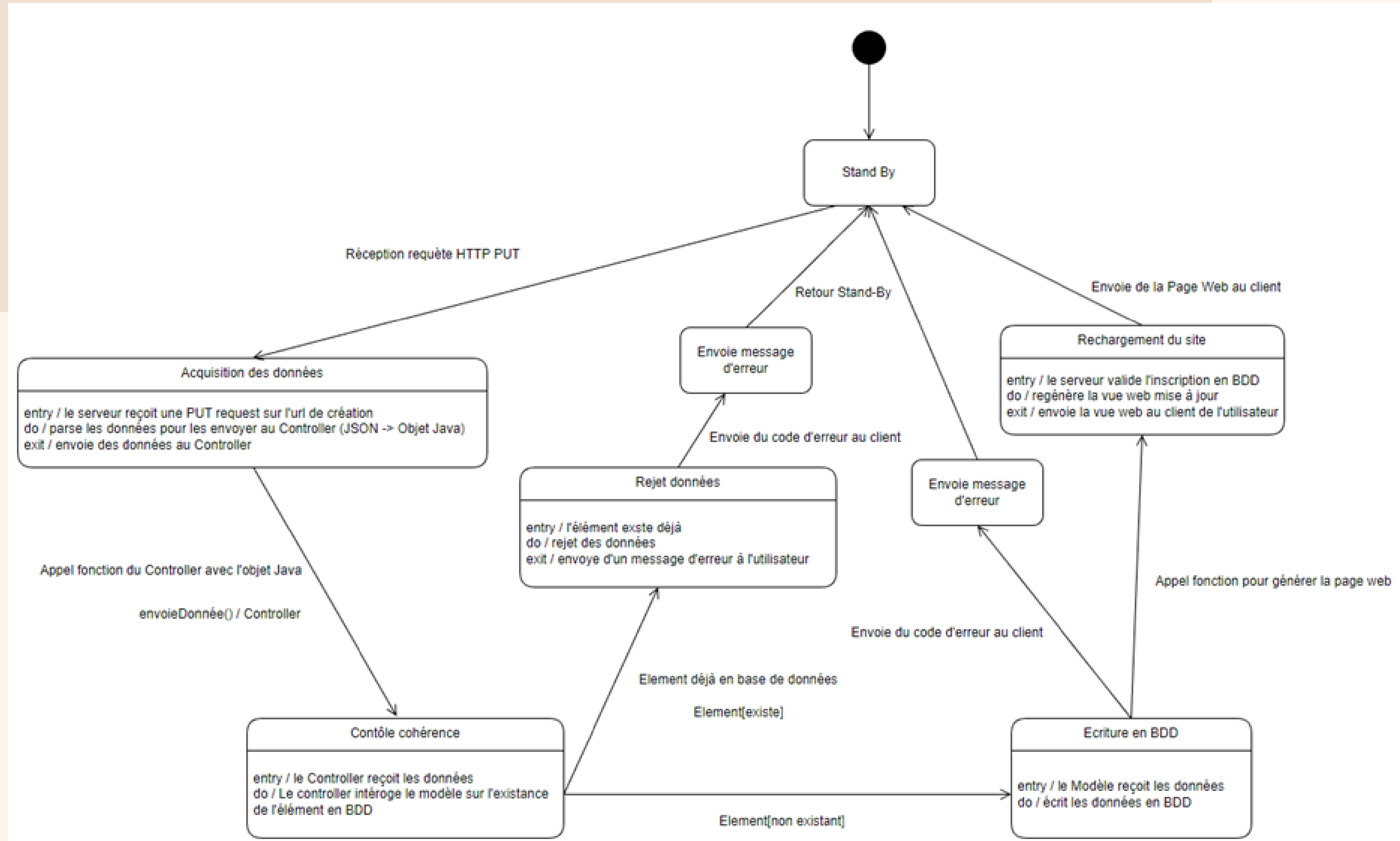
UML: Diagramme d'activité



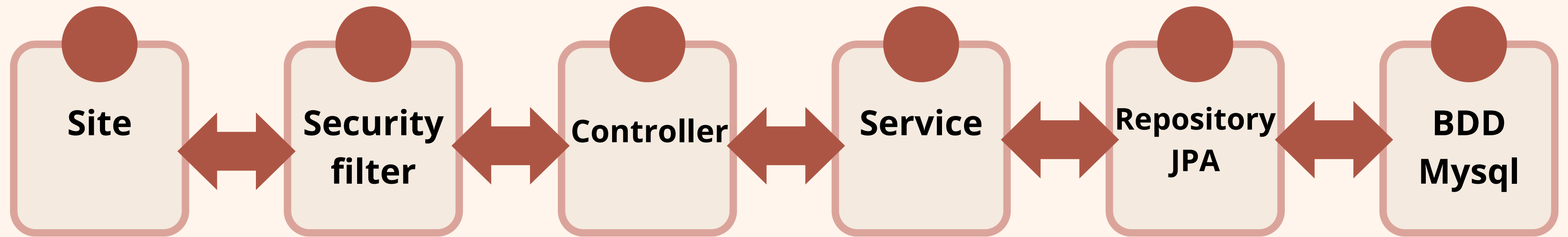
UML: Diagramme de séquence



UML: Diagramme d'état-transition



Process





Vue 3



Vite



Vuetify



node js



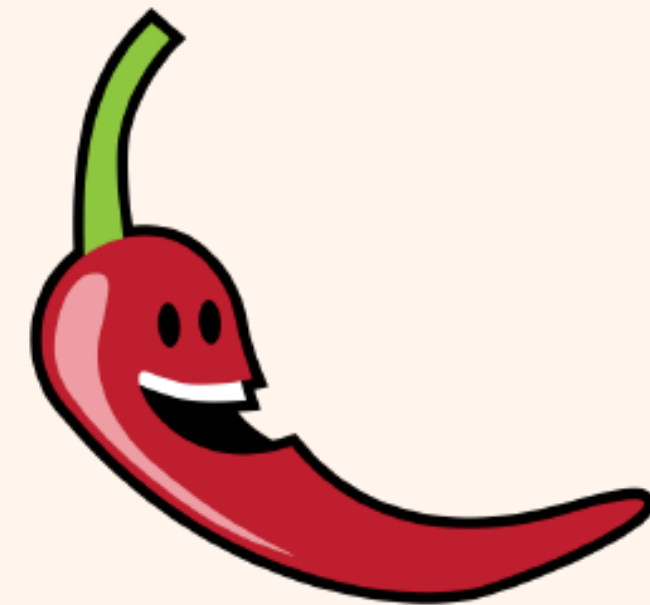
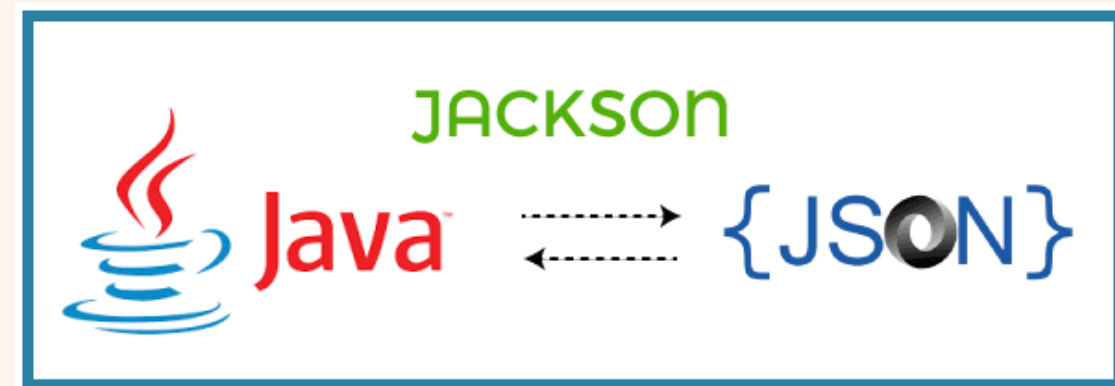
npm

Frontend: communication

```
submitLogin() {
  const credentials = {
    username: this.login.username,
    password: this.login.password
  };
  axios.post( url: 'http://localhost:3001/auth/signin', credentials)
    .then(response => {
      const { token, firstName, lastName, role } = response.data;
      localStorage.setItem('jwtToken', token);
      localStorage.setItem('firstName', firstName);
      localStorage.setItem('lastName', lastName);
      localStorage.setItem('role', role);

      this.login.username = '';
      this.login.password = '';
      this.$router.push('/');
      EventBus.login(role);

      this.$notify( options: {
        type : 'success',
        title: 'Connexion réussie',
        text: 'Vous êtes maintenant connecté'
      });
    })
    .catch(error => {
      console.error('Erreur lors de la connexion:', error);
    });
},
```



Lombok

Controller

```
1 package com.jeeproject.jeeProject.controllers;
2
3 > import ...
12
13 @RestController
14 @RequestMapping("/discipline")
15 public class DisciplineController {
16     @Autowired
17     private DisciplineServiceImpl disciplineServiceImpl;
18
19     @GetMapping
20     public Iterable<DisciplineResource> getDisciplines(){return disciplineServiceImpl.getDisciplines();}
21
22     @GetMapping("/{id}")
23     public DisciplineResource getDiscipline(@PathVariable Long id){ return disciplineServiceImpl.getDiscipline(id); }
26
27     @PostMapping("/create")
28     @PreAuthorize("hasAnyRole('ROLE_ADMIN', 'ROLE_ADMINISTRATIVE_MANAGER')")
29     public ResponseEntity<Object> createDiscipline(@RequestBody DisciplineResource disciplineResource){
30         try{
31             DisciplineResource createdDisciplineResource = disciplineServiceImpl.createDiscipline(disciplineResource);
32             return ResponseEntity.status(HttpStatus.CREATED).body(createdDisciplineResource);
33         }catch (Exception e){
34             return ResponseEntity.status(HttpStatus.CONFLICT).body(e.getMessage());
35         }
36     }
```

Les Services

```
4 usages  🧑 Matthieu Serra *
public class SiteServiceImpl implements SiteService {
    @Autowired
    private SiteRepository siteRepository;
    @Autowired
    private SiteMapper siteMapper;
    1 usage  🧑 Matthieu Serra
    public Iterable<Site> getSites() { return siteRepository.findAll(); }
    1 usage  🧑 Matthieu Serra
    @Override
    public SiteResource getSite(Long id) {
        Site site = siteRepository.findById(id).orElseThrow(() -> new EntityNotFoundException("Site not found in db with id :"+id));
        return siteMapper.siteToSiteResource(site);
    }
    1 usage  new *
    public SiteResource updateSite(SiteResource siteResource, Long id){
        Site site = siteRepository.findById(id).orElseThrow(() -> new EntityNotFoundException("Site not found in db with id :"+id));
        siteMapper.updateSiteFromResource(siteResource, site);
        siteRepository.save(site);
        return siteMapper.siteToSiteResource(site);
    }
    1 usage  🧑 Matthieu Serra
    public SiteResource createSite(SiteResource siteResource) throws IOException {
        Site site = siteMapper.siteResourceToSite(siteResource);
        if(!estEnDB(siteResource)){
            siteRepository.save(site);
        }else{
            throw new IOException("Site déjà présent en DB");
        }
        return siteMapper.siteToSiteResource(site);
    }
}
```

Mappers

```
1 package com.jeeproject.jeeProject.mappers;
2
3 > import ...
10
    2 usages  ⚡ Matthieu Serra
11 @Mapper(config = MapperConfig.class)
12 public interface EpreuveMapper {
    6 usages  ⚡ Matthieu Serra
13     @Mapping(target = "disciplineNom",source="discipline.nom")
14     EpreuveResource epreuveToEpreuveResource(Epreuve epreuve);
15
    1 usage  ⚡ Matthieu Serra
16     @Mapping(target="id",ignore=true)
17     @Mapping(target = "discipline",ignore = true)
18     Epreuve epreuveResourceToEpreuve(EpreuveResource epreuveResource);
    1 usage  ⚡ Matthieu Serra
19     @Mapping(target="id",ignore=true)
20     void updateEpreuveFromResource(EpreuveResource epreuveResource, @MappingTarget Epreuve epreuve);
21 }
```

Data Transfer Object

Repository


4 usages  Matthieu Serra

@Repository

```
public interface SiteRepository extends CrudRepository<Site, Long> {
```

1 usage  Matthieu Serra

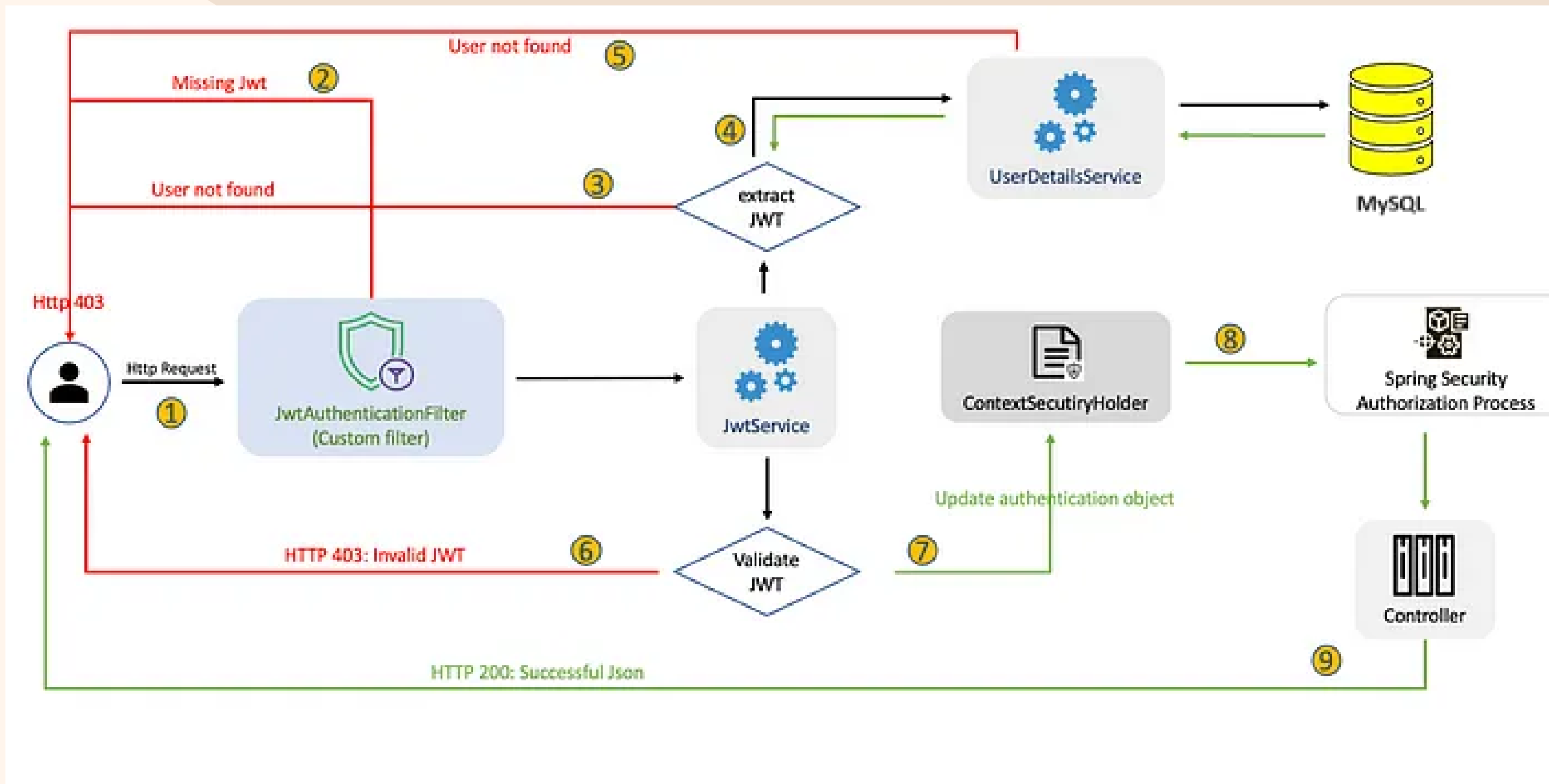
```
boolean existsByNomAndVilleAndCategorie(String nom, String ville, Site.CategorieSite category);
```

2 usages  Matthieu Serra

```
Site findByNom(String nom);
```

```
}
```

Mécanisme d'authentification



- Implémentation de l'authentification avec spring security
- Utilisation de token JWT pour valider l'authentification

Q

A

**Question
Time**

?