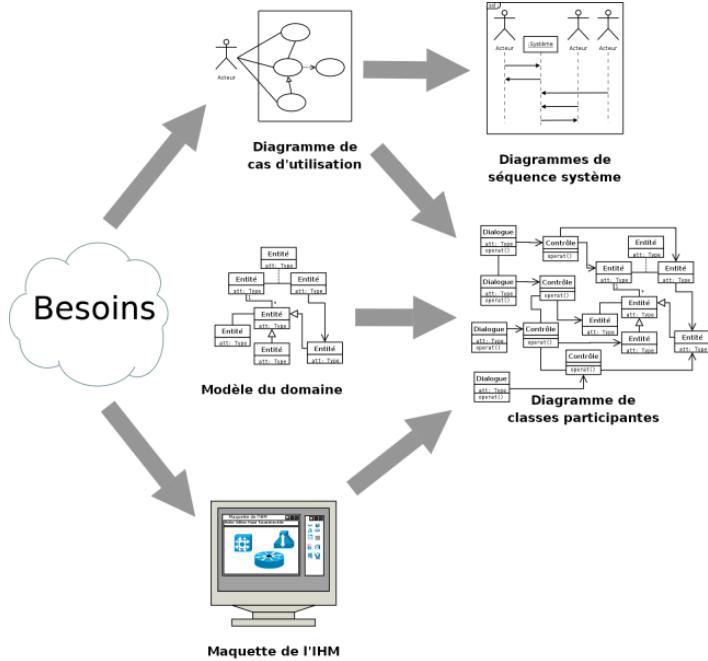
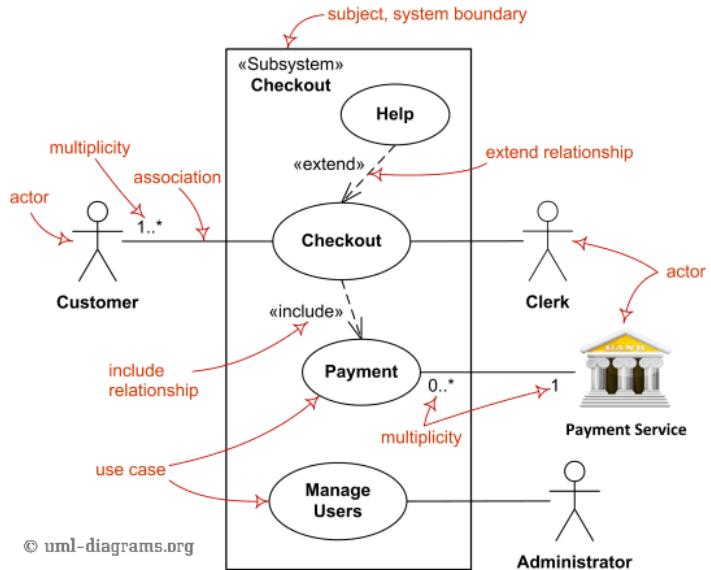


Exemple de Projet : Système de Gestion de Bibliothèque



Pour illustrer la création de diagrammes UML pour un projet de système de gestion de bibliothèque, nous allons détailler plusieurs types de diagrammes couramment utilisés dans l'UML.

1. Diagramme de Cas d'Utilisation



a. Description

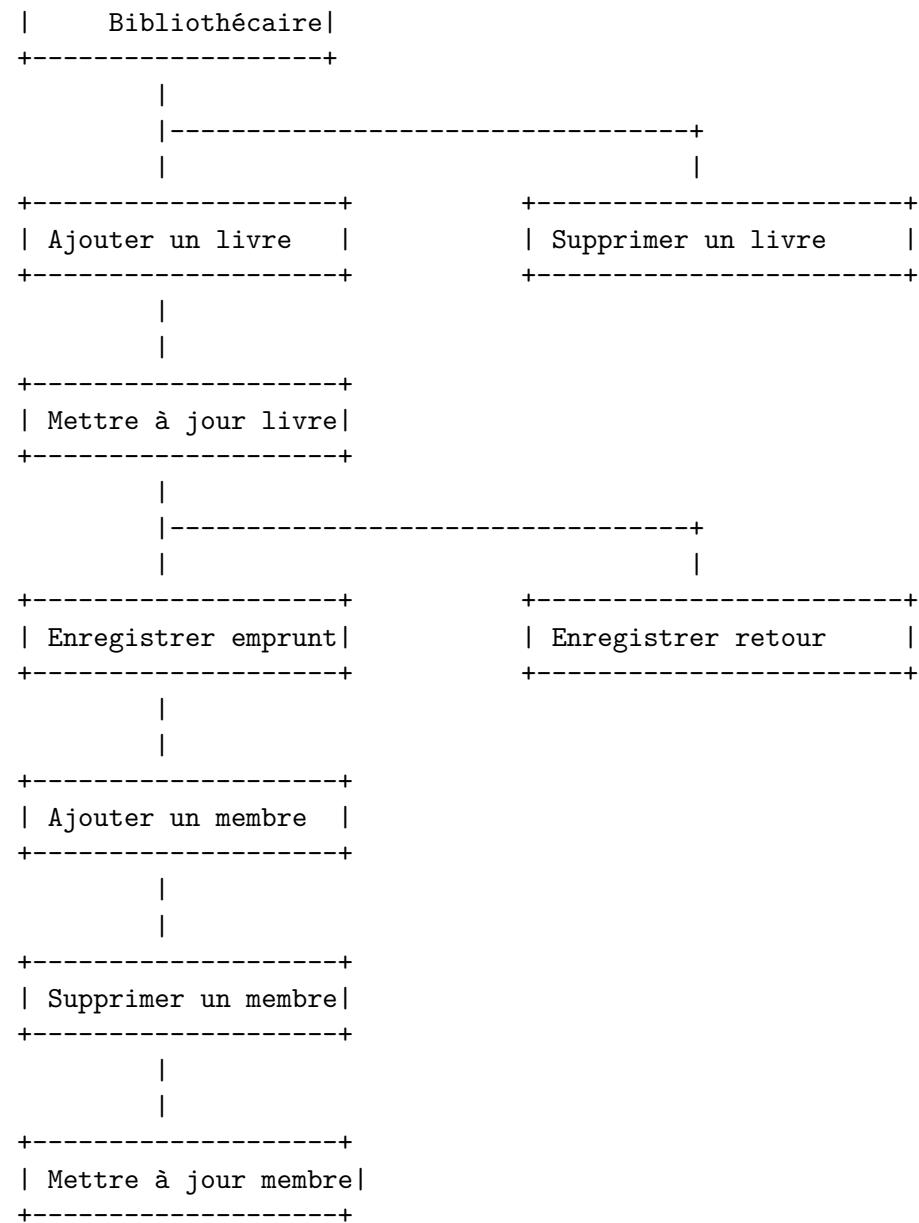
Ce diagramme montre les interactions entre les utilisateurs (acteurs) et le système pour accomplir différentes tâches.

b. Acteurs

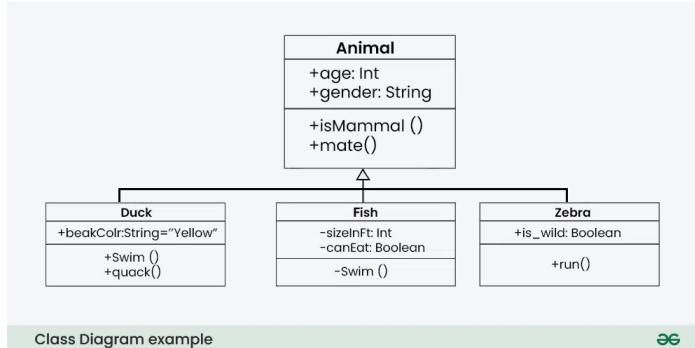
- **Bibliothécaire** : Gère les livres et les emprunts.
- **Membre** : Emprunte et rend les livres.
- **Administrateur** : Gère les comptes utilisateurs et les priviléges.

c. Cas d'utilisation

- Ajouter un livre
- Supprimer un livre
- Mettre à jour les informations d'un livre
- Enregistrer un nouvel emprunt
- Enregistrer le retour d'un livre
- Ajouter un membre
- Supprimer un membre
- Mettre à jour les informations d'un membre



2. Diagramme de Classes



a. Description

Ce diagramme montre la structure statique du système en représentant les classes, leurs attributs, méthodes et les relations entre elles.

b. Classes principales

• Livre

- Attributs : titre, auteur, ISBN, datePublication
- Méthodes : ajouterLivre(), supprimerLivre(), mettreAJourLivre()

• Membre

- Attributs : nom, identifiant, adresse, numéroTéléphone
- Méthodes : ajouterMembre(), supprimerMembre(), mettreAJourMembre()

• Emprunt

- Attributs : dateEmprunt, dateRetour, livreID, membreID
- Méthodes : enregistrerEmprunt(), enregistrerRetour()

• Bibliothécaire

- Attributs : nom, identifiant
- Méthodes : gérerLivres(), gérerEmprunts()

+-----+ <td></td>	
Livre	
+-----+	
- titre	
- auteur	
- ISBN	

```

| - datePublication   |
+-----+
| + ajouterLivre()   |
| + supprimerLivre() |
| + mettreAJourLivre()|
+-----+
|           |
|           |
|           |
+-----+
| Emprunt          |
+-----+
| - dateEmprunt    |
| - dateRetour      |
| - livreID         |
| - membreID        |
+-----+
| + enregistrerEmprunt() |
| + enregistrerRetour()  |
+-----+
|           |
|           |
|           |
+-----+
| Membre           |
+-----+
| - nom            |
| - identifiant    |
| - adresse        |
| - numéroTéléphone |
+-----+
| + ajouterMembre() |
| + supprimerMembre()|
| + mettreAJourMembre()|
+-----+
|           |
|           |
|           |
+-----+
| Bibliothécaire   |

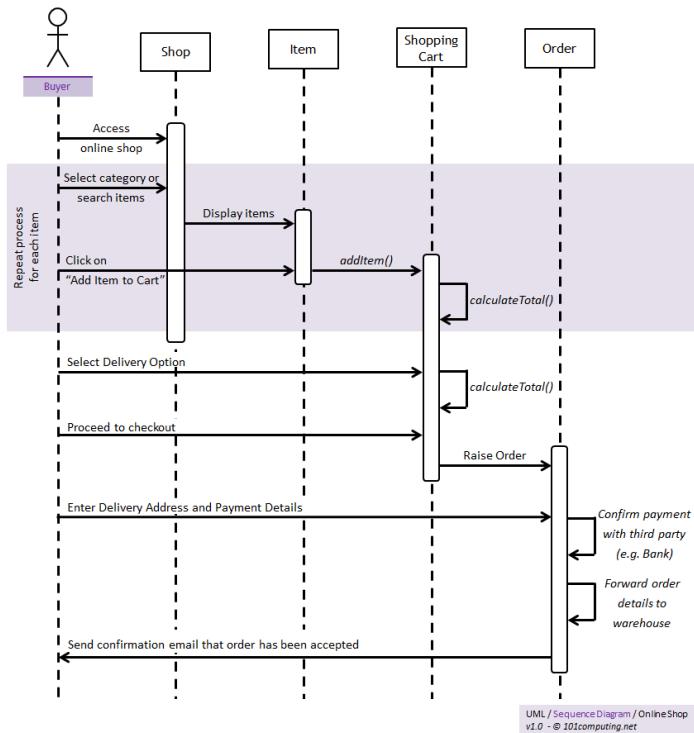
```

```

+-----+
| - nom           |
| - identifiant   |
+-----+
| + gérerLivres() |
| + gérerEmprunts() |
+-----+

```

3. Diagramme de Séquence



a. Description

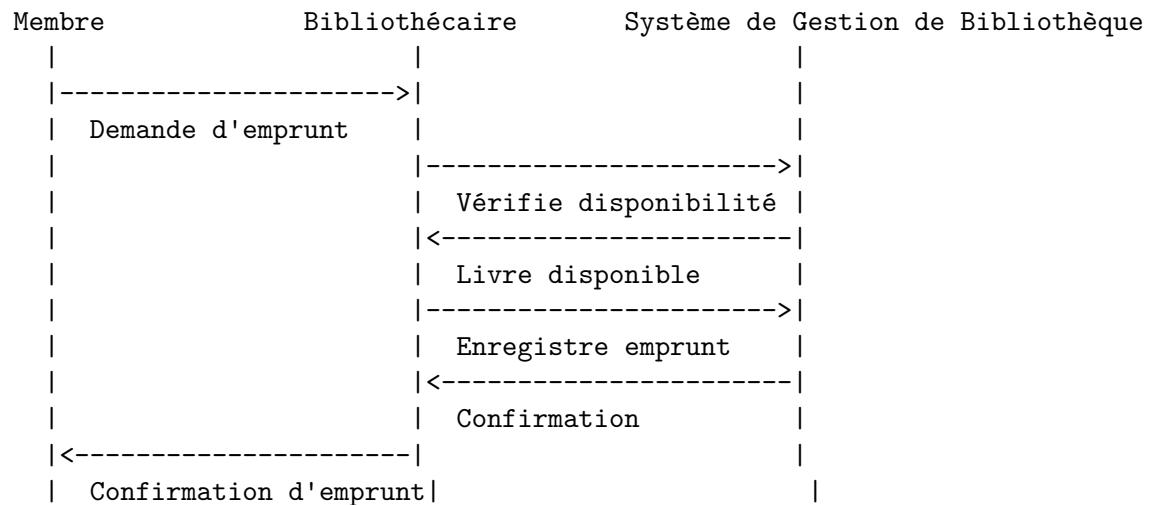
Ce diagramme montre les interactions entre les objets dans le temps pour un scénario spécifique, comme l'emprunt d'un livre.

b. Scénario : Emprunter un livre

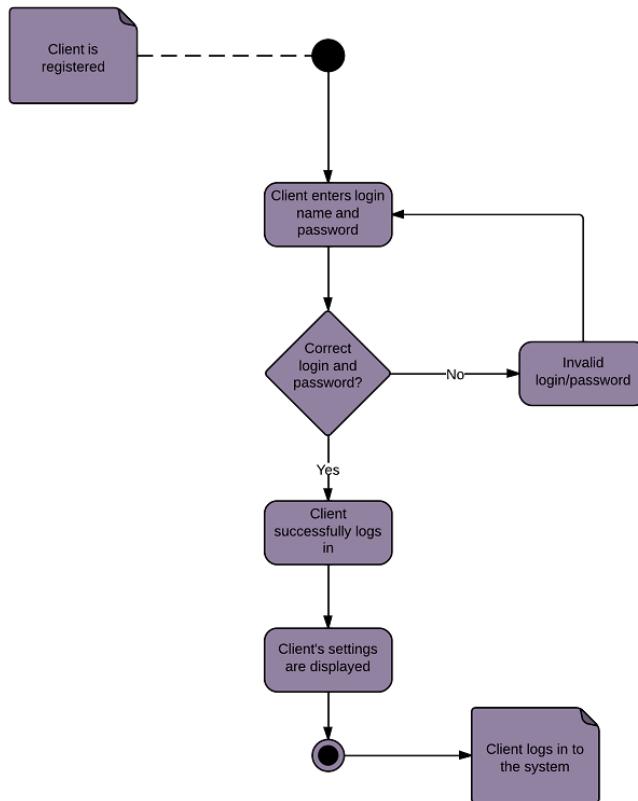
- Acteurs :** Membre, Bibliothécaire, Système de Gestion de Bibliothèque

- **Étapes :**

1. Le membre demande à emprunter un livre.
2. Le bibliothécaire vérifie la disponibilité du livre.
3. Le bibliothécaire enregistre l'emprunt dans le système.
4. Le système met à jour les informations du livre.



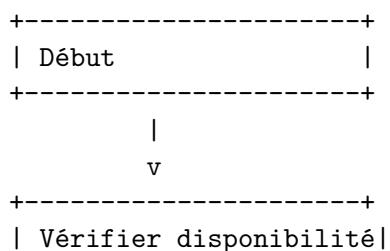
4. Diagramme d'Activités

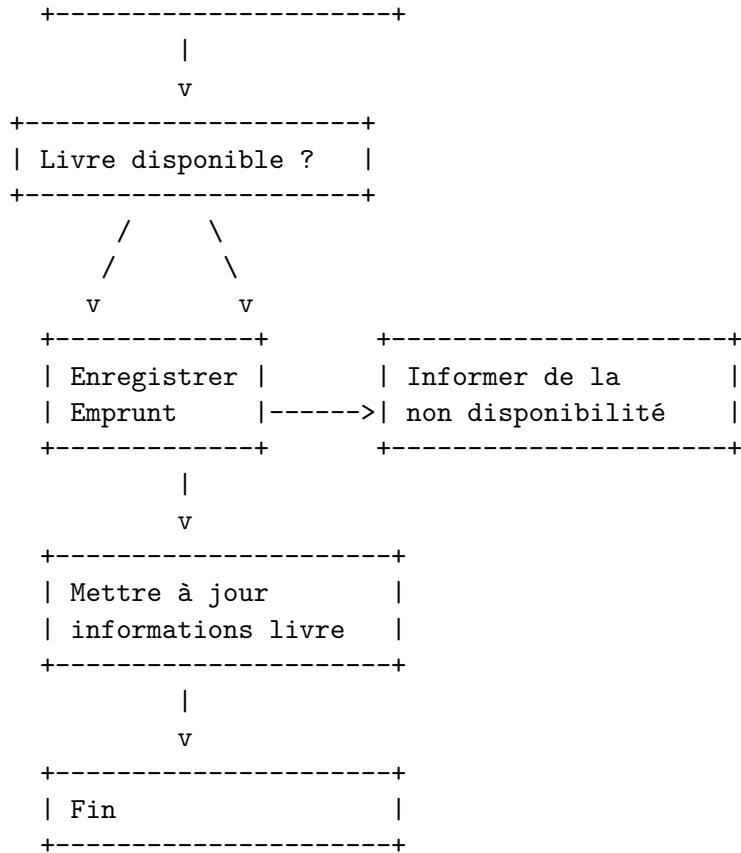


a. Description

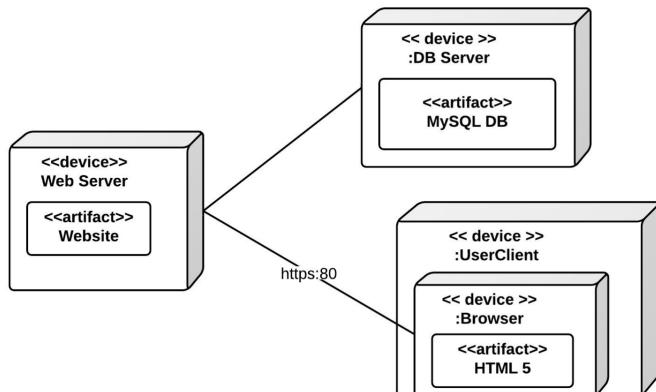
Ce diagramme décrit le flux de travail ou les activités pour un processus, tel que la gestion des emprunts.

b. Processus : Gestion des emprunts





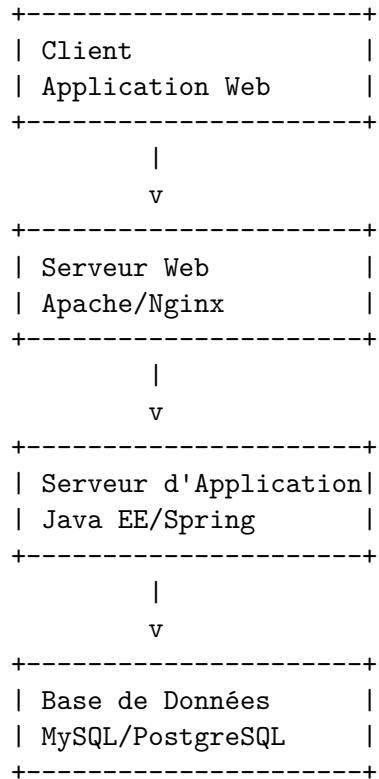
5. Diagramme de Déploiement



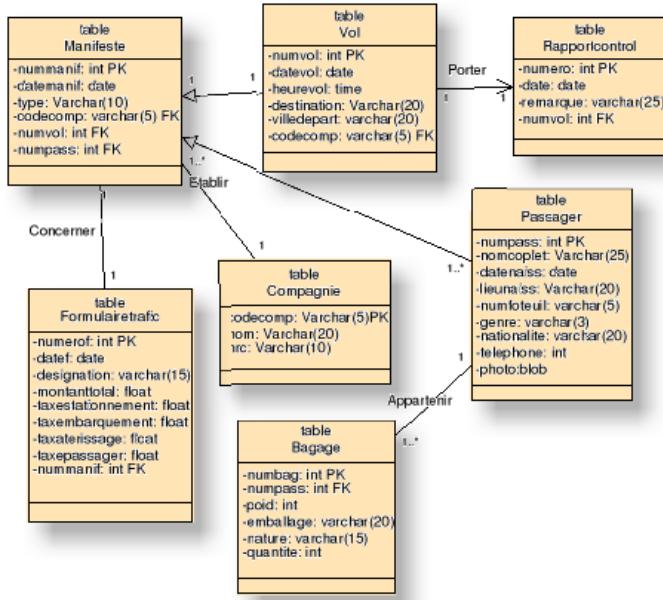
a. Description

Ce diagramme montre la configuration matérielle et logicielle du système.

b. Configuration



6. Diagramme de Relation d'Entité



a. Description

Un diagramme de relations d'entités (diagramme ER ou ERD - Entity Relationship Diagram) est une **représentation visuelle des relations qui existent entre les éléments d'une base de données**.

Les ERD sont des types d'organigramme spécialisés qui transmettent les types de relations entre les différentes entités d'un système.

Conclusion

Ces exemples de diagrammes UML fournissent une vue d'ensemble complète d'un système de gestion de bibliothèque, couvrant différents aspects tels que les interactions utilisateur, la structure du système, les processus métiers, et la configuration matérielle. En utilisant ces diagrammes, les développeurs et les parties prenantes peuvent mieux comprendre et concevoir le système avant d'entamer le développement.