

Compte rendu de projet - Bookify -

Brief N°: 5

par : Ibtissam EL HANI

Introduction

Le projet Bookify vise à établir une plateforme web de réservation. L'objectif principal est de concevoir et mettre en place une base de données solide pour la gestion efficace des réservations sur le site. Ce compte rendu met en lumière la conception de la structure de données, l'utilisation de SQL et UML pour modéliser l'architecture, ainsi que la planification pour garantir la continuité de la plateforme.

Modélisation UML

Le modèle UML (Unified Modeling Language) est utilisé pour représenter graphiquement l'architecture du projet.

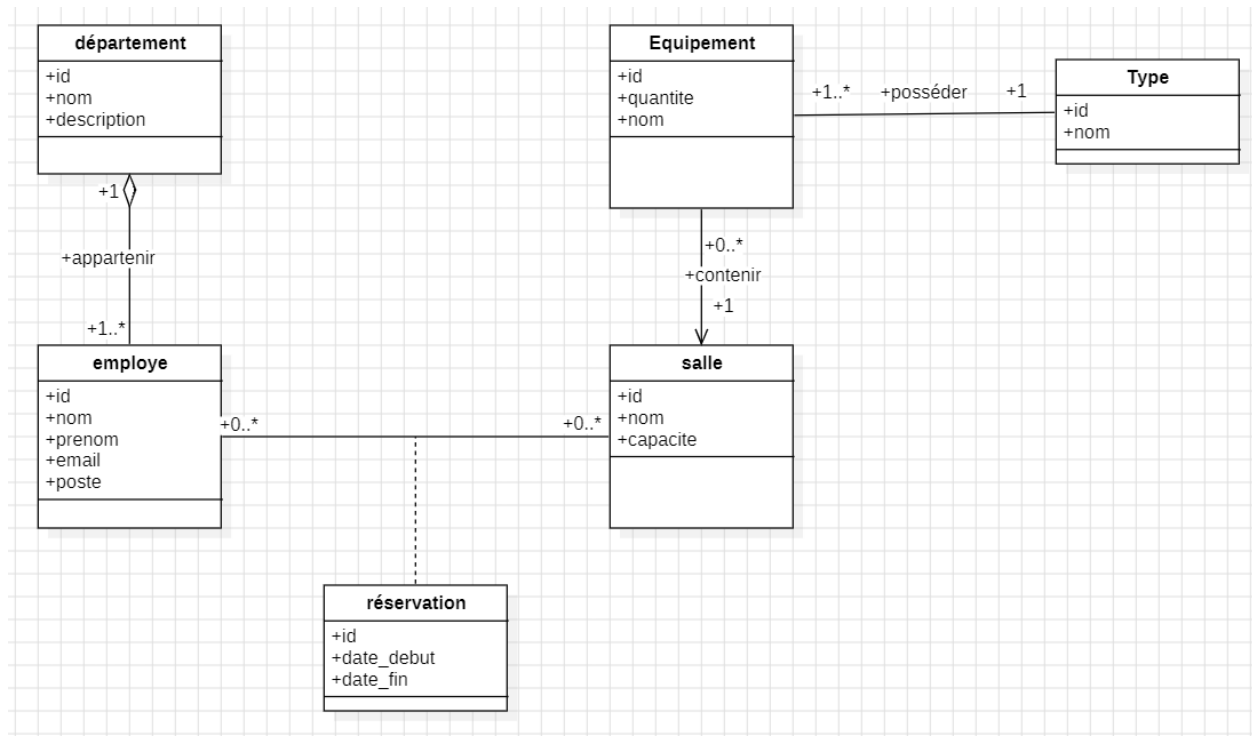
- **StarUML comme outil de modélisation**

Dans le cadre du projet Bookify, l'outil StarUML a été utilisé pour faciliter la modélisation de l'architecture du système. StarUML, en tant qu'outil de modélisation UML, a apporté une contribution significative à la compréhension visuelle et à la documentation de la structure de données du projet.



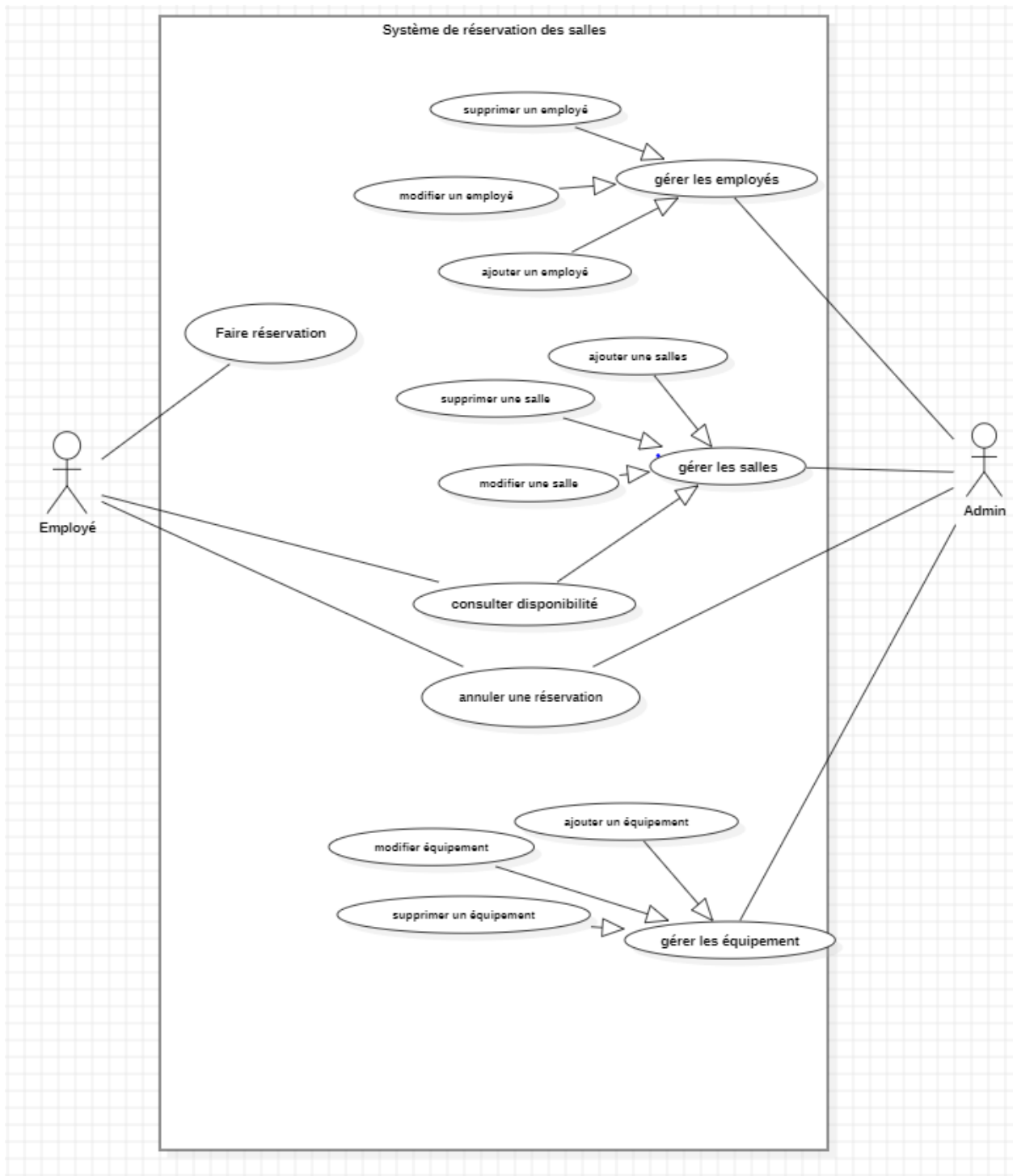
• Diagramme de classe

Le diagramme de classe généré offre une vue d'ensemble des entités fondamentales du système, telles que les classes "Salle", "Employé" et "Réservation", ainsi que de leurs relations. Les fonctionnalités avancées de StarUML ont été exploitées pour définir les propriétés des classes, les associations entre les entités et les multiplicités, offrant une représentation visuelle détaillée de la structure de la base de données.



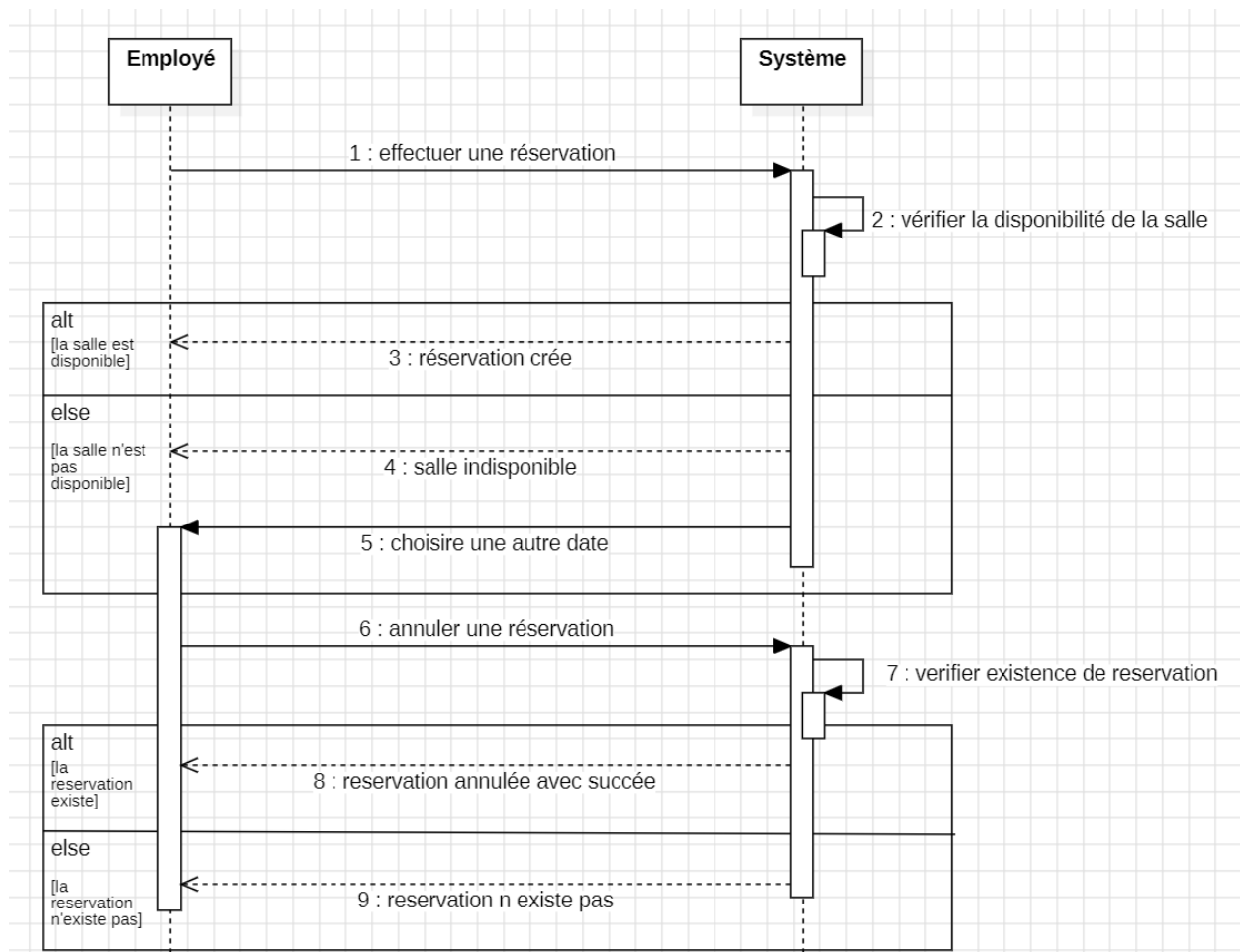
• Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation pour le projet Bookify offre une vue globale des interactions entre les acteurs (employées) et le système. Il identifie les différentes fonctionnalités offertes par la plateforme, telles que la gestion de salles, la réservation, et la gestion des employées.



- Diagramme de séquence

Le diagramme de séquence se concentre sur une séquence spécifique d'interactions entre les objets du système, illustrant comment ils coopèrent pour atteindre un objectif particulier. Dans le contexte de Bookify, un diagramme de séquence pourrait décrire le processus détaillé de réservation d'une salle. Il met en lumière l'enchaînement chronologique des messages échangés entre l'employé et le système, montrant comment chaque étape de la réservation est traitée.



Utilisation de SQL

SQL est utilisé comme langage de requête pour interagir avec la base de données. Des requêtes SQL sont écrites pour permettre des opérations efficaces telles que la recherche de salles , la création de nouvelles réservations, et la gestion des informations employé.

- **PHPMyAdmin**

PHPMyAdmin a servi d'interface web conviviale pour administrer la base de données MySQL, simplifiant la création, la modification et la visualisation des tables et des données.



- **MySQL**

MySQL, en tant que système de gestion de base de données relationnelle, a été le choix idéal pour stocker les informations essentielles au fonctionnement de Bookify, telles que les détails des salles, des employés et des réservations.



- **XAMPP**

XAMPP, en fournissant un environnement de développement local, a grandement simplifié le processus de configuration du serveur Apache, de PHP, de MySQL et de PHPMyAdmin sur les machines de développement.



- **Exemple de requête SQL**

▷ Execute

```
CREATE DATABASE brief;
```

▷ Execute

```
CREATE TABLE employes (  
    id int PRIMARY KEY AUTO_INCREMENT,  
    nom varchar(20),  
    prenom varchar(20),  
    email varchar(30),  
    poste varchar(20),  
    departement_id int,  
    FOREIGN KEY(departement_id) REFERENCES departements(id));
```

▷ Execute

```
INSERT INTO departements (nom, description)
VALUES
('Production','départements 1'),
('Ventes','départements 2'),
('Marketing','départements 3'),
('Comptabilité','départements 4'),
('Développement','départements 5');
```

▷ Execute

```
SELECT * FROM employes where nom='el hani';
```

▷ Execute

```
SELECT * FROM employes where nom='elkoh' AND prenom='zak';
```

▷ Execute

```
SELECT * FROM employes where nom='elkoh' OR nom='el hani';
```

▷ Execute

```
SELECT e.nom, e.prenom, e.poste, d.nom AS departement
FROM employes e
JOIN departements d ON e.departement_id = d.id;
```

▷ Execute

```
UPDATE employes
SET nom='aabila', prenom='said'
WHERE id=10;
```


▷ Execute

```
DELETE FROM equipements
WHERE nom='prises' AND quantite =2;
```

▷ Execute

```
CREATE Procedure chercher_employe(IN nom VARCHAR(20))
BEGIN
    SELECT * FROM employes e
    WHERE e.nom = nom;
END;
```

▷ Execute

```
CALL chercher_employe('el hani');
```

▷ Execute

```
CREATE Function max_quant() RETURNS int
BEGIN
    DECLARE m int;
    SELECT MAX(quantite) INTO m FROM equipements;
    RETURN m;
END;
```

▷ Execute

```
DELETE FROM equipements
WHERE nom='prises' AND quantite =2;
```

▷ Execute

```
CREATE INDEX IDX_employe_email ON employes(email);
```