



---

# DOCUMENT D'ARCHITECTURE TECHNIQUE GESTION DE SERVICES

---

*Membres :*

- GHERNAOUT MOHAMED WASSIM
- KAHOUADJI LYDIA
- LAADIDAOUI OMAR
- MEDJOUB NAOUAL
- MEGHOUCHE RAFIK
- MERZEG RAMZI AMEUR
- MOKRANE ISLAM
- SALAS GAUTHIER

*Client :*

Pr. Djelloul ZIADI

# Historique

Version	Date	Modifications réalisés	Rédigé par	Relu par
<b>0.1</b>	29/10/2021 30/11/2021	Rédaction	Kahouadji Lydia	
<b>0.1</b>	22/12/2021	Vérification		Toute l'équipe

# Table des matières

<b>A propos du document</b>	<b>4</b>
L'objet . . . . .	4
Historique . . . . .	4
Documents de référence . . . . .	4
Glossaire . . . . .	4
<b>Justification du choix des technologies d'implémentation</b>	<b>5</b>
<b>Architecture générale du projet</b>	<b>6</b>
Le schéma général du projet . . . . .	6
Description des constituants . . . . .	6
Description du constituant Serveur Node - Front-End . . . . .	6
Description du constituant Serveur Node - Back-End . . . . .	7
Description du constituant Serveur Apache - DataBase . . . . .	7
Le front-end . . . . .	7
Liste des composants . . . . .	7
Liste des services . . . . .	7
Le back-end . . . . .	8
Liste des contrôleurs . . . . .	8
Liste des modèles . . . . .	8
Modèle Conceptuel des Données . . . . .	8
Schéma de la base de données . . . . .	9
<b>Description des scénarios</b>	<b>10</b>
Gestion des authentifications . . . . .	10
Authentification . . . . .	10
Gestion des utilisateurs . . . . .	11
Ajout d'un utilisateur . . . . .	11
Modification d'un utilisateur . . . . .	11
Suppression d'un utilisateur . . . . .	12
Consulter des programmes horaires . . . . .	12
Demande d'un programme horaire . . . . .	13
Modification d'un programme horaire . . . . .	13
Validation d'un programme horaire . . . . .	14
Gestion des Sessions de choix . . . . .	14
Ouverture d'une session de choix . . . . .	14
Fermeture d'une session de choix . . . . .	15
Gestion des formations . . . . .	15
Ajout d'une formation . . . . .	15
Mise à jour d'une formation . . . . .	16
Suppression d'une formation . . . . .	16
<b>Traçabilité</b>	<b>17</b>

# Table des figures

1	Schéma d'architecture général du projet . . . . .	6
2	Diagramme de classes . . . . .	8
3	Schéma de la base de données . . . . .	9
4	Diagramme de séquence authentification . . . . .	10
5	Diagramme de séquence ajout d'un utilisateur . . . . .	11
6	Diagramme de séquence modification d'un utilisateur . . . . .	11
7	Diagramme de séquence suppression d'un utilisateur . . . . .	12
8	Diagramme de séquence consultation des programmes horaires . . . . .	12
9	Diagramme de séquence demande d'un programme horaire . . . . .	13
10	Diagramme de séquence modification d'un programme horaire . . . . .	13
11	Diagramme de séquence validation d'un programme horaire . . . . .	14
12	Diagramme de séquence ouverture d'une session de choix . . . . .	14
13	Diagramme de séquence fermeture d'une session de choix . . . . .	15
14	Diagramme de séquence ajout d'une formation . . . . .	15
15	Diagramme de séquence mise à jour d'une formation . . . . .	16
16	Diagramme de séquence suppression d'une formation . . . . .	16

# A propos du document

## L'objet

Description de l'architecture du projet et des composantes du projet Gestion des services. Ce document fournit une vue globale de l'architecture technique envisagée pour la réalisation de l'application web de gestion des services proposée par M. Djelloul Ziadi. Le modèle d'architecture décrit à l'aide de schémas ainsi que des descriptions comment l'application doit être conçue de manière à répondre aux spécifications fonctionnelles demandées par le client. Son but est donc de fournir une conception détaillée des différents modules pour faciliter leur développement ainsi que la mise en œuvre des spécifications fonctionnelles déjà citées dans le document de la spécification technique du besoin.

## Historique

Version	Date	Modifications réalisés	Rédigé par	Relu par
0.1	29/11/2021	Rédaction	Toute l'équipe	
0.2	02/12/2022	Mise à jour	Toute l'équipe	Toute l'équipe
0.3	14/01/2022	Vérification	Toute l'équipe	Rafik Meghouche, Lydia Kahouadji

La version actuelle du document est **1.1** réalisée le **13 décembre 2021**.

Il y a donc une version antérieure **1.0** établie pendant le semaine du **Lundi 29 novembre 2021** au **02 décembre 2021**.

## Documents de référence

- Le document de la spécification fonctionnelle du besoin (STB).

## Glossaire

- **API** : Interface de programmation (Application Programming Interface).
- **Front-end** : Constituant représentant l'interface graphique d'une application.
- **Back-end** : Le back-end est la combinaison d'une base de données et d'un logiciel écrit dans un langage côté serveur.
- **Contrôleur** : Un contrôleur correspond à une entité permettant d'appliquer sur le modèle la logique dictée par l'utilisateur, dans notre projet cette partie est assurée par l'API REST faite avec NODEJS.
- **Service** : Un service correspond à une entité permettant au composants de générer des requêtes HTTP de manière cataloguée.
- **Rest** : Rest est une architecture logicielle assurant le traitement des demandes des clients, vers un serveur en transmettant les données. demandées.

# Justification du choix des technologies d'implémentation

React JS est une technologie imposée pour le projet, il présente l'avantage de fonctionner sur le principe de composant, les composants représentent un élément affichable dépendant de paramètre, donc modulaire. De plus le Framework est minimaliste de cette façon il convient à notre projet qui tourne sur un serveur disposant de ressources limitées, et propose une prise en main facile car le nombre de choses à connaître pour débiter est inférieur à ceux d'un Framework plus classique (ex Angular).

NODEJS est la technologie qui nous a été imposée pour le projet. Le Back-End de notre application fonctionnera donc avec un serveur express. Une technologie qui nous permettra de développer nos idées plus rapidement. Cette dernière est plus concise et le risque d'erreurs potentielles est minimal. De plus, celui-ci utilise l'api REST, ce qui permet de partager les données rapidement entre le Front-end et Back-end selon la norme de JSON.

SQL est la technologie de gestion de base de données relationnelle qui nous a été imposée. ça permet de personnaliser la façon dont les données sont stockées. De plus, cette technologie possède une grande communauté, ce qui permet de trouver facilement des articles exploitables sur internet.

# Architecture générale du projet

## Le schéma d'architecture général du projet

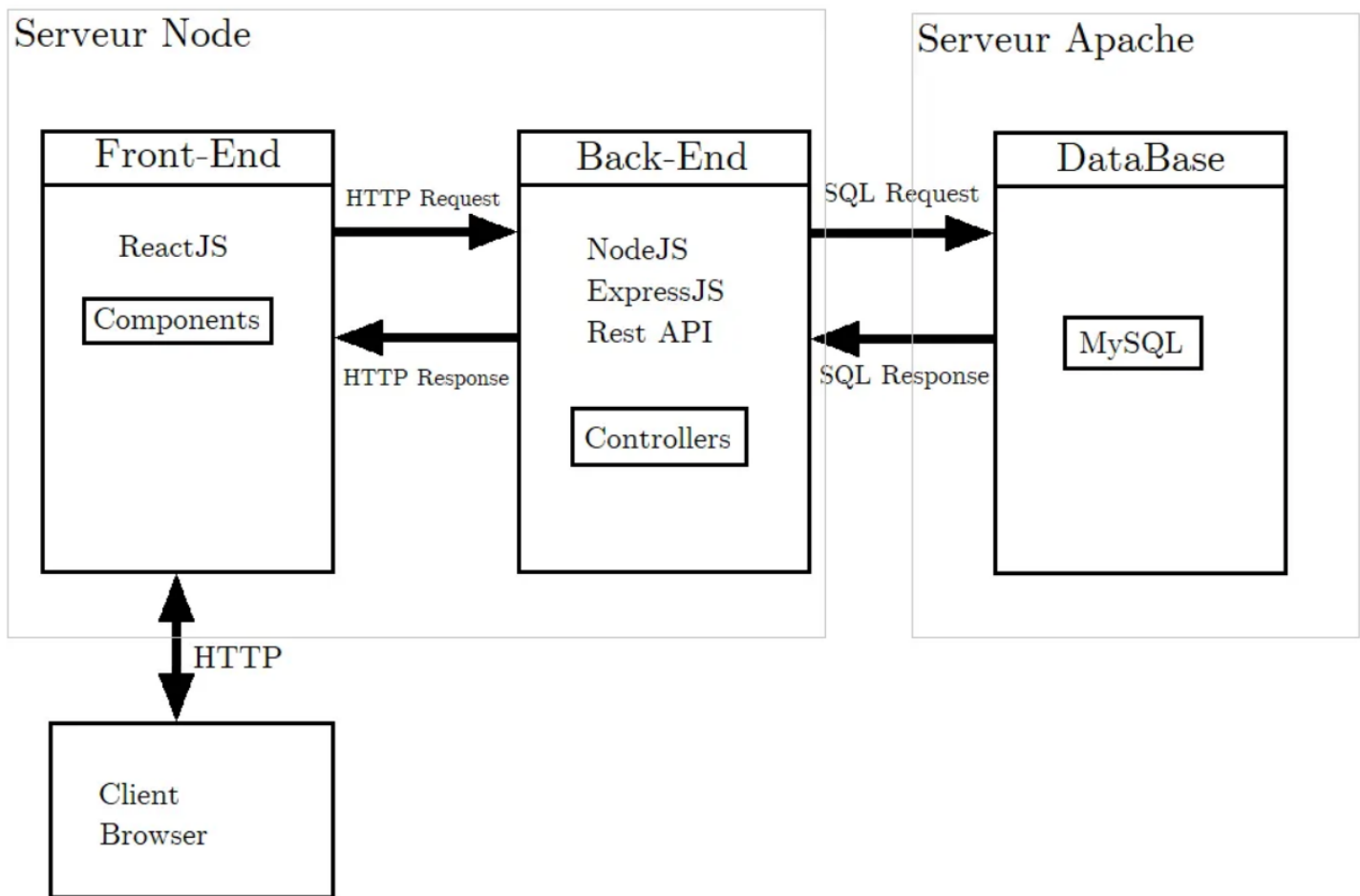


FIGURE 1 – Schéma d'architecture général du projet

## Description des constituants

### Description du constituant Serveur Node - Front-End

- **Rôle** : Interface Graphique
- **Services offerts (interfaces)** : Propose un service interactif à l'utilisateur pour accéder aux fonctionnalités du projet.
- **Dépendances avec d'autres constituants (services utilisés, composants « sur étagère » utilisés)** : Serveur Node - Back-End
- **Langage de programmation et framework** : JavaScript, React Js
- **Taille et complexité** : Moyenne

---

## Description du constituant Serveur Node - Back-End

- **Rôle** : Calcul et Communication avec la BDD
- **Services offerts (interfaces)** : REST Api
- **Dépendances avec d'autres constituants (services utilisés, composants « sur étagère » utilisés)** : NodeJS (ExpressJS) 6
- **Langage de programmation** : Javascript
- **Taille et complexité** : Moyenne

## Description du constituant Serveur Apache - DataBase

- **Rôle** : Stocker les informations (utilisateurs, planning, etc.)
- **Services offerts (interfaces)** : API
- **Dépendances avec "autres constituants (services utilisés, composants « sur étagère » utilisés)** : NodeJS
- **Langage de programmation** : MYSQL
- **Taille et complexité** : Moyenne

## Le front-end

### Liste des composants

- **ComposantAuthentification** : Composant en charge de la validation des tentatives d'accès au site.
- **ComposantProgrammeHoraire** : Composant en charge de la demande d'un programme horaire et de sa consultation.
- **ComposantGestionProgrammesHoraires** : Composant en charge de la gestion des programmes horaires (consultation, validation, modification)
- **ComposantGestionUtilisateurs** : Composant en charge de la gestion des enseignants (ajout, suppression, modification)
- **ComposantGestionFormation** : Composant en charge de la gestion des formations et modules (ajout, suppression, modification)
- **ComposantGestionSessions** : Composant en charge de la gestion des sessions de choix (ouverture, fermeture)

### Liste des services

- **UtilisateurService** : Service qui permet au composant ComposantAuthentification de générer des requêtes HTTP(GET) à destination du contrôleur pour traiter les demandes d'authentification des utilisateurs .
- **GestionUtilisateursService** : Service qui permet au composant ComposantGestionUtilisateurs de générer des requêtes HTTP(GET, POST, PUT, PATCH, DELETE) à destination du contrôleur pour traiter les utilisateurs.
- **GestionProgrammeService** : Service qui permet au composant ComposantGestionProgrammesHoraires de générer des requêtes HTTP(GET, POST, PUT, PATCH, DELETE) à destination du contrôleur pour traiter les programmes horaires.
- **GestionSessionService** : Service qui permet au composant ComposantGestionSessions de générer des requêtes HTTP(GET, POST, PUT, PATCH, DELETE) à destination du contrôleur pour traiter les sessions de choix.



# Le back-end

## Liste des contrôleurs

- **UtilisateurController** : Contrôleur en charge de vérifier les informations d'authentification fournie par l'utilisateur essayant d'effectuer cette dernière.
- **GestionUtilisateursController** : Contrôleur en charge de gérer les utilisateurs (ajout, mise à jour, suppression).
- **GestionProgrammeController** : Contrôleur en charge de gérer les programmes (demande, validation, consultation, modification).
- **GestionSessionController** : Contrôleur en charge de gérer les sessions (ouverture, fermeture).

## Liste des modèles

- **Utilisateur** : Représentation en données d'un dit "utilisateur" (identifiant, email, mot de passe, rôle).
- **Module** : Représentation en données d'un dit "module" (identifiant, intitulé).
- **Formation** : Représentation en données d'une dite "formation" (identifiant, intitulé).
- **Programme horaire** : Représentation en données d'un dit "programme horaire" (identifiant, nombre d'heure, nombre de groupes, enseignant associé).
- **Université** : Représentation en données d'une dite "université" en matière de sessions (identifiant, intitulé, statut de la session, description de la session).

## Diagramme de classes

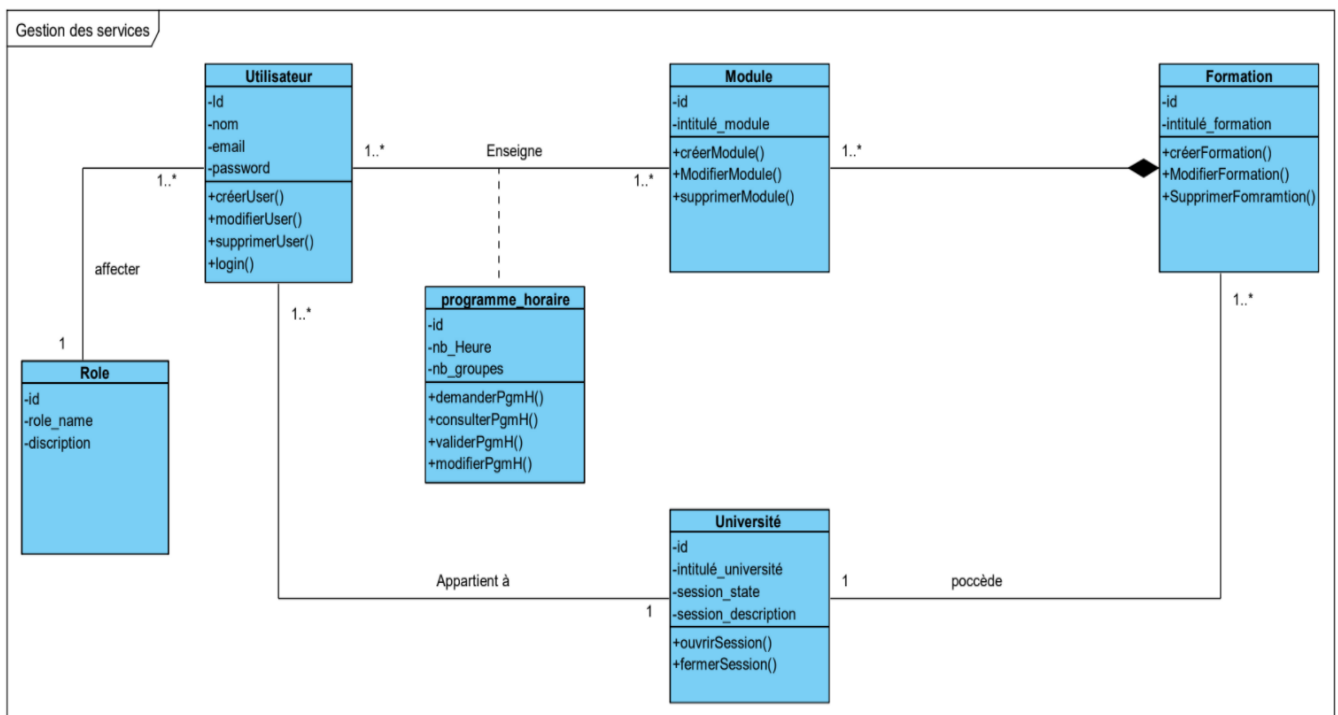


FIGURE 2 – Diagramme de classes

## Schéma de la base de données

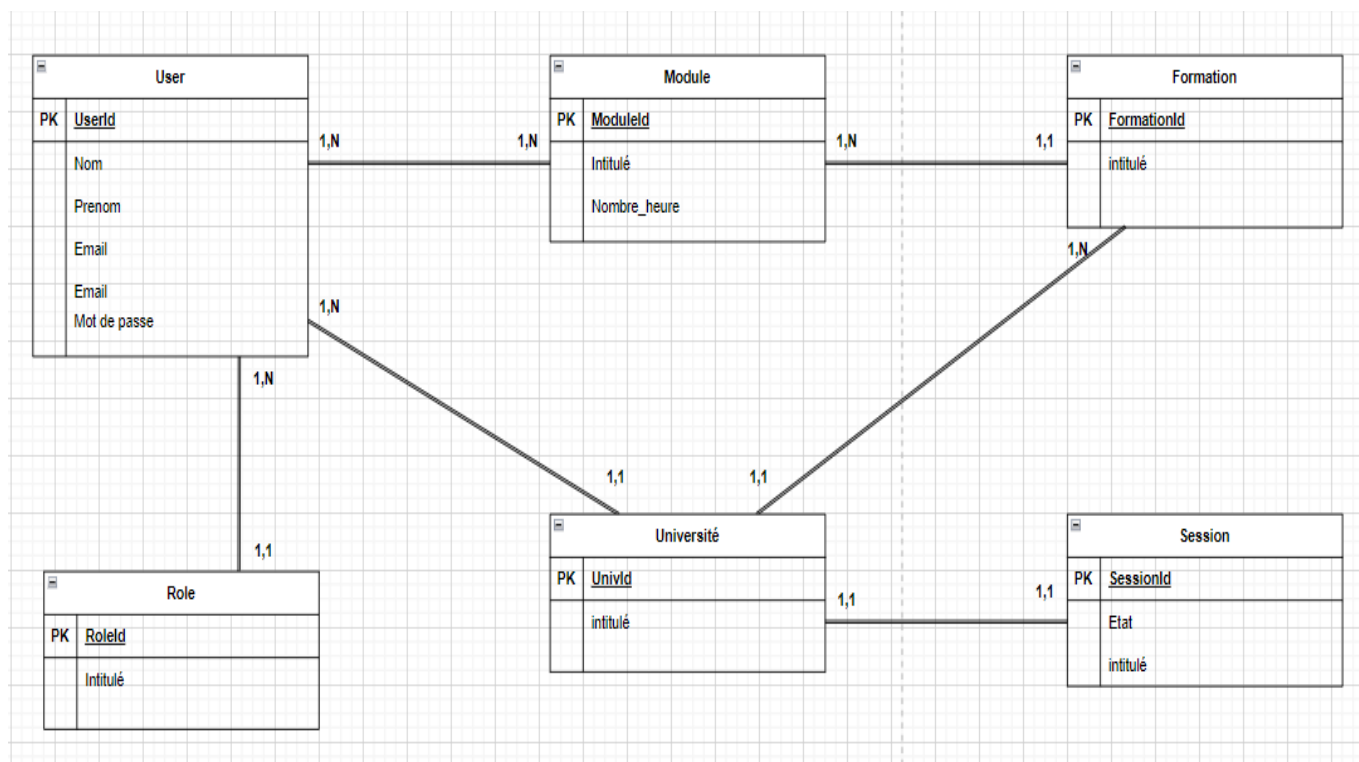


FIGURE 3 – Schéma de la base de données

# Description des scénarios

## Gestion des authentifications

### Authentification

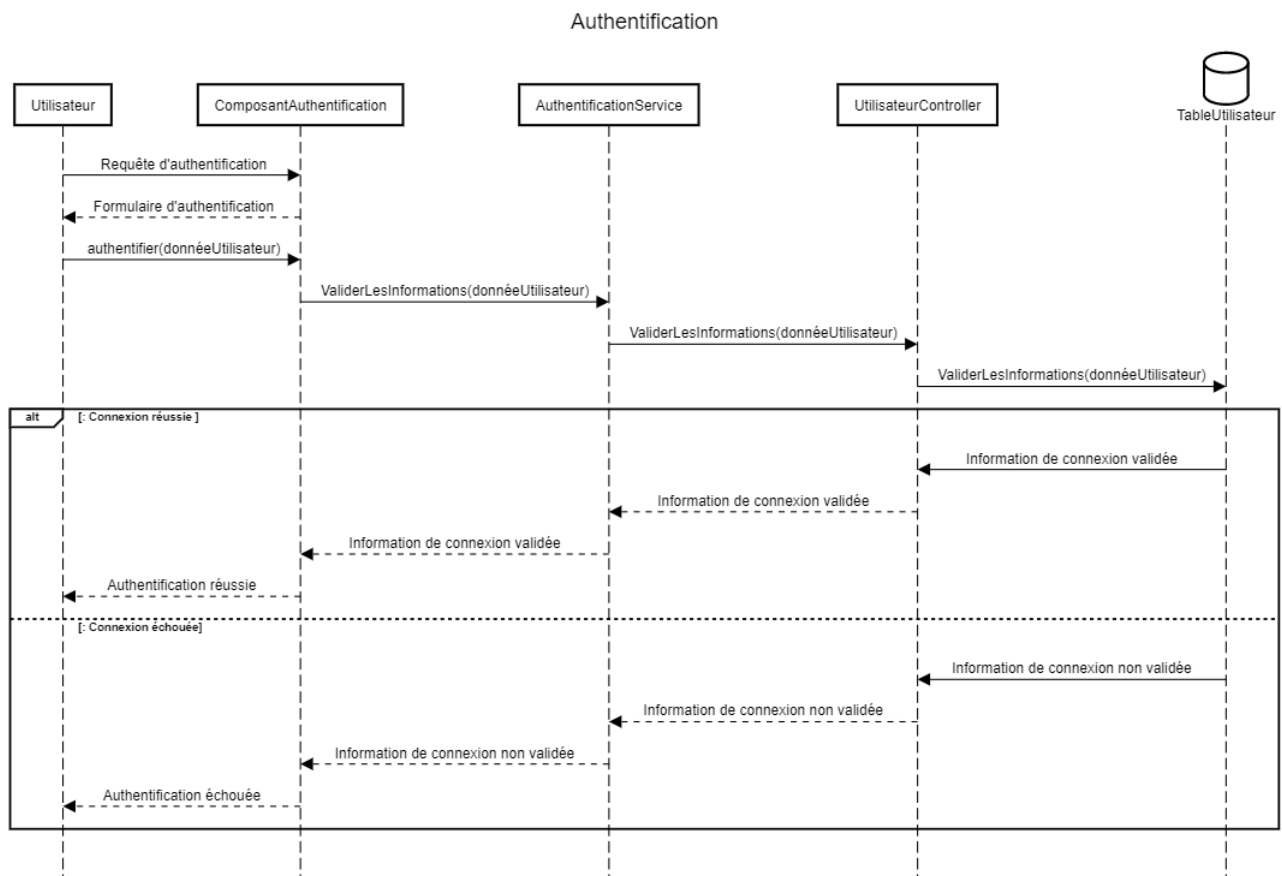


FIGURE 4 – Diagramme de séquence authentification

# Gestion des utilisateurs

## Ajout d'un utilisateur

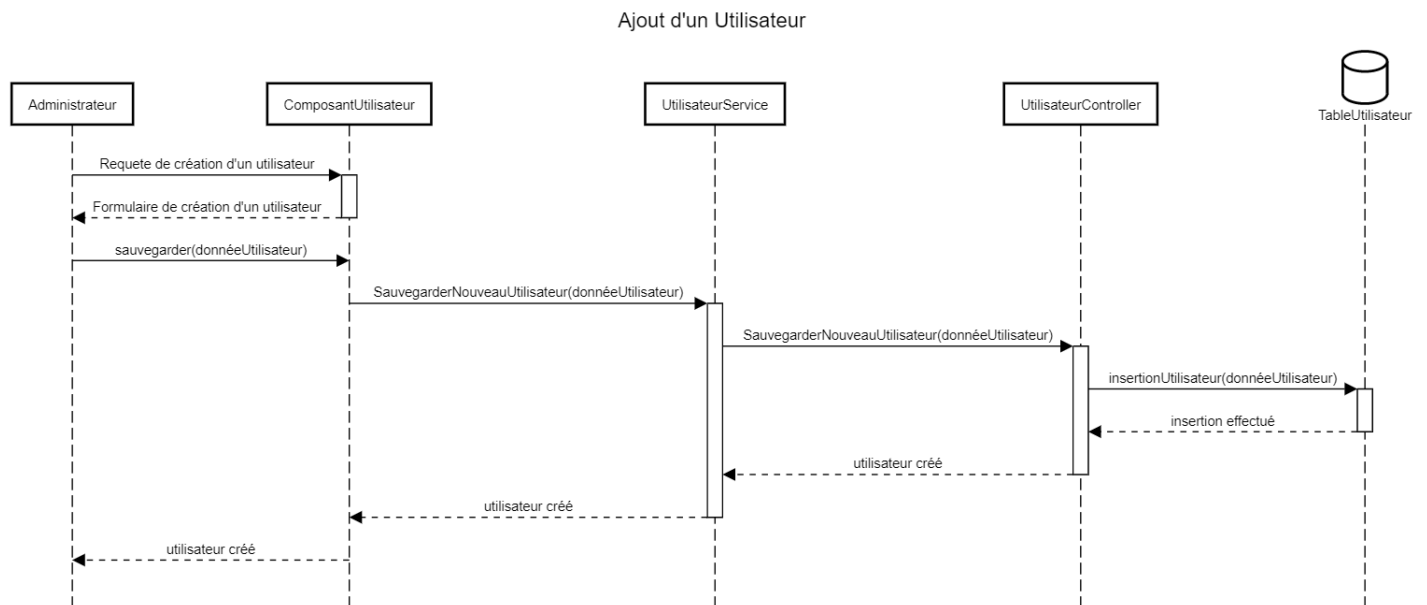


FIGURE 5 – Diagramme de séquence ajout d'un utilisateur

## Modification d'un utilisateur

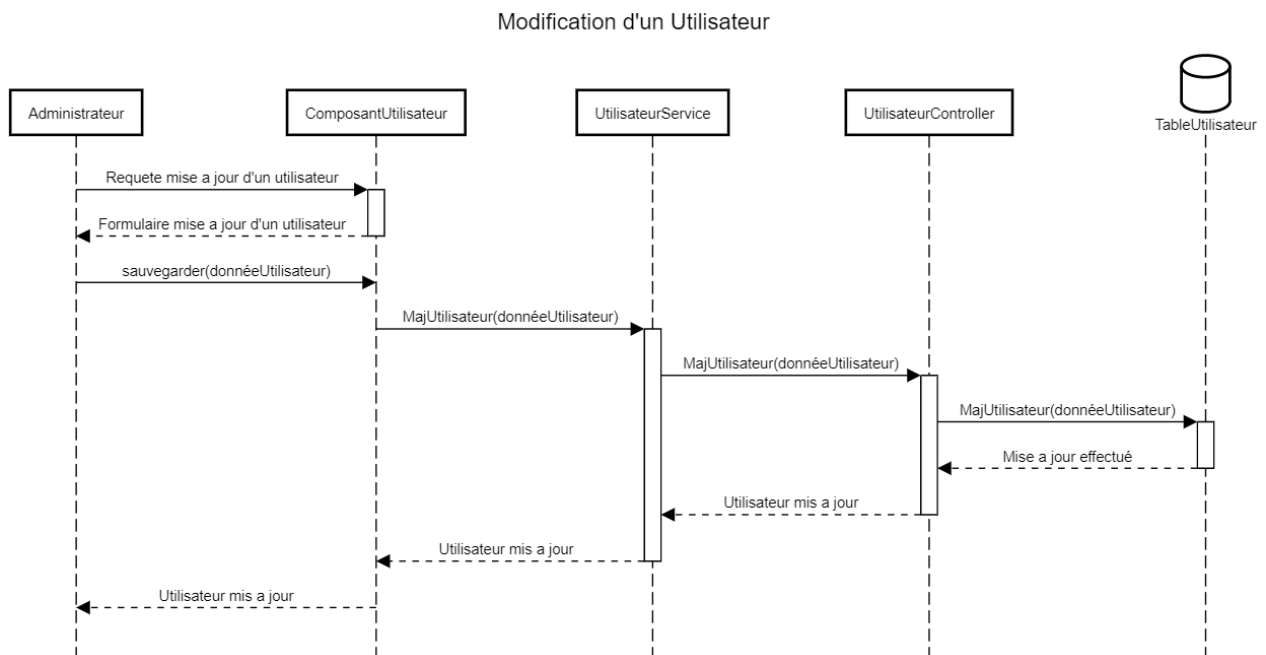


FIGURE 6 – Diagramme de séquence modification d'un utilisateur

## Suppression d'un utilisateur

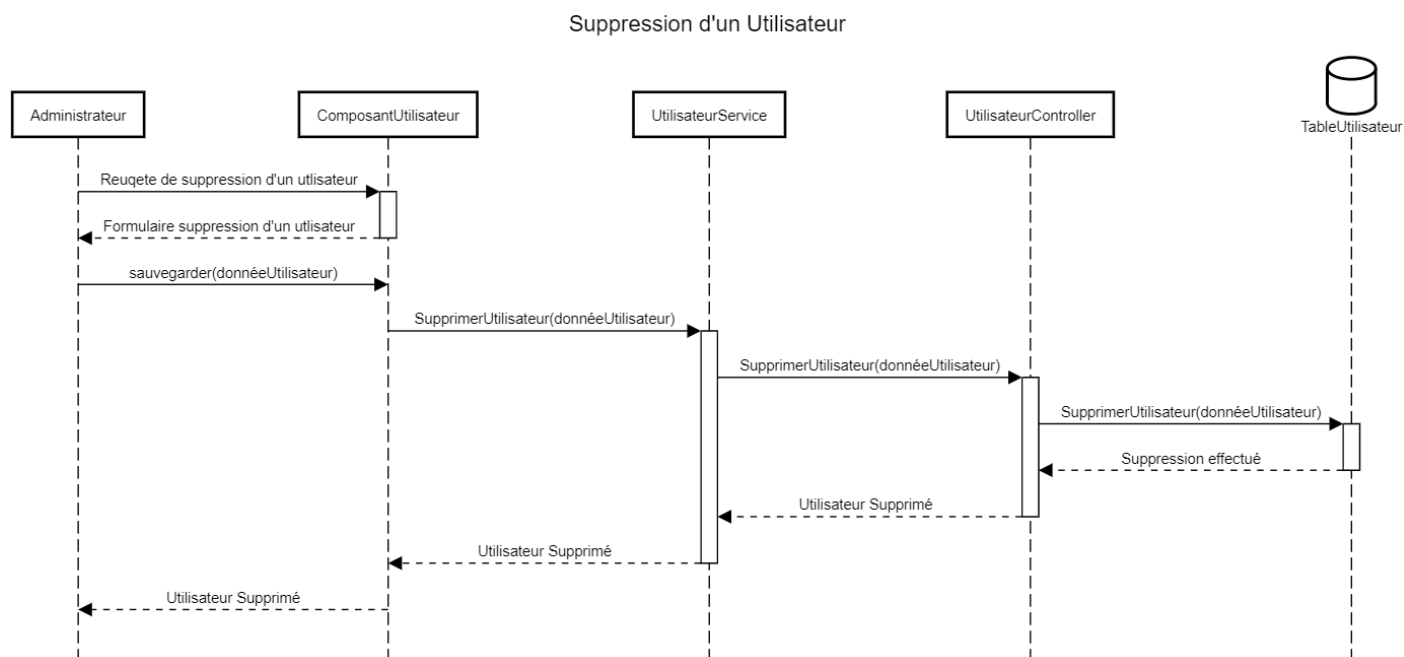


FIGURE 7 – Diagramme de séquence suppression d'un utilisateur

## Consultation des programmes horaires

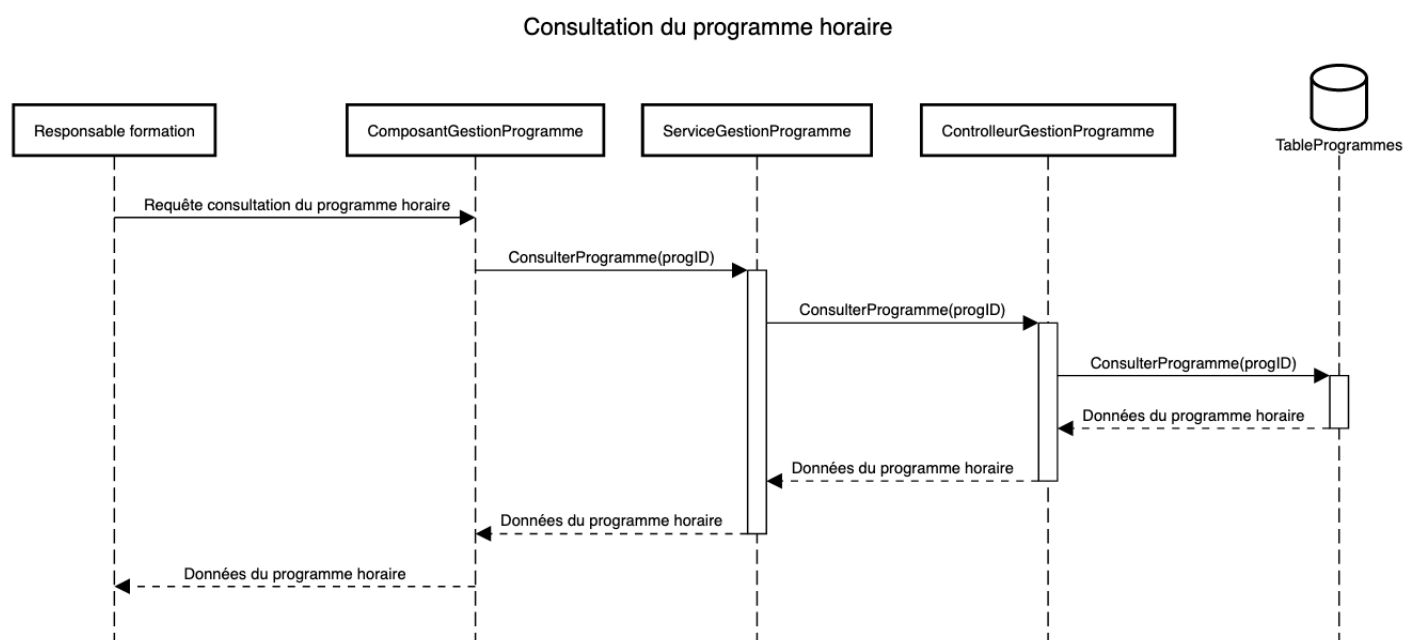


FIGURE 8 – Diagramme de séquence consultation des programmes horaires

# Demande d'un programme horaire

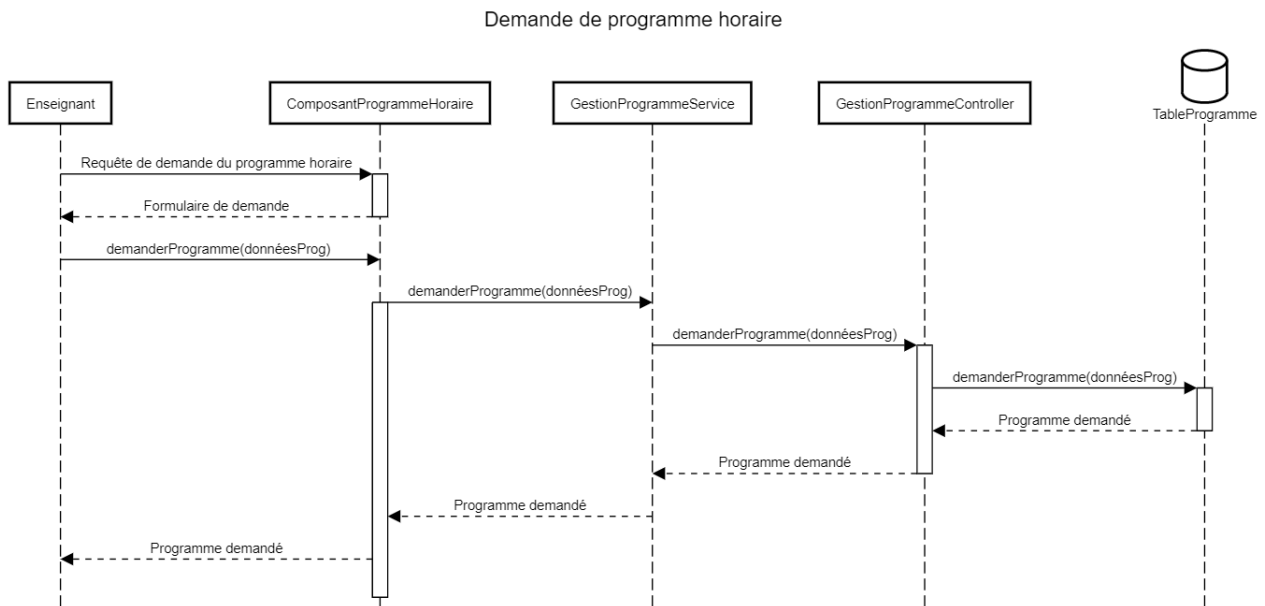


FIGURE 9 – Diagramme de séquence demande d'un programme horaire

# Modification d'un programme horaire

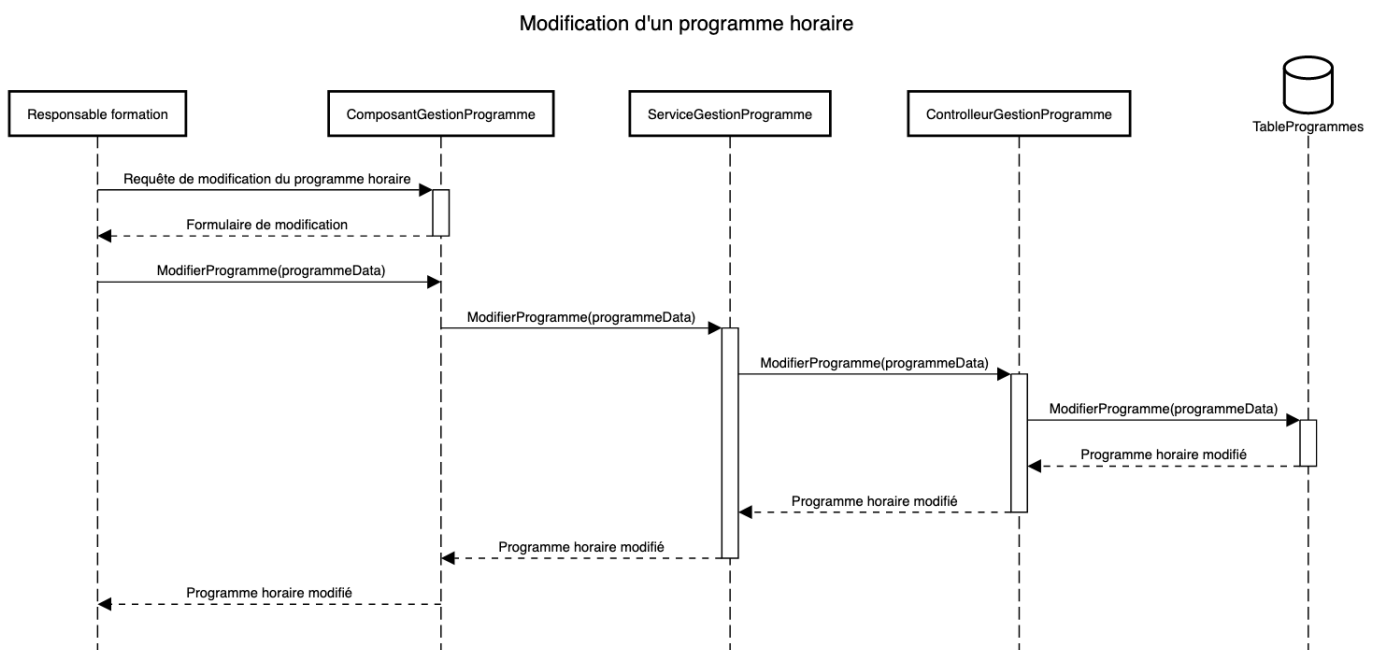


FIGURE 10 – Diagramme de séquence modification d'un programme horaire

# Validation d'un programme horaire

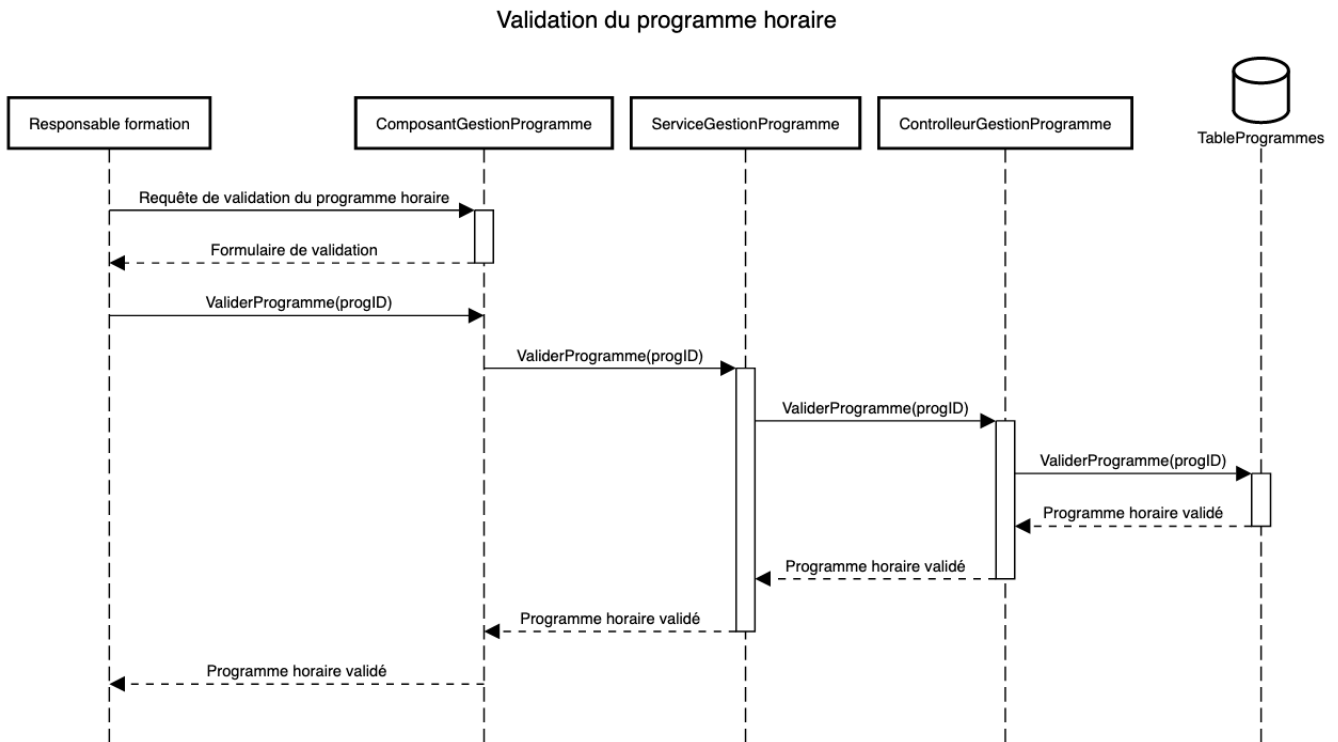


FIGURE 11 – Diagramme de séquence validation d'un programme horaire

## Gestion des sessions de choix

### Ouverture d'une session de choix

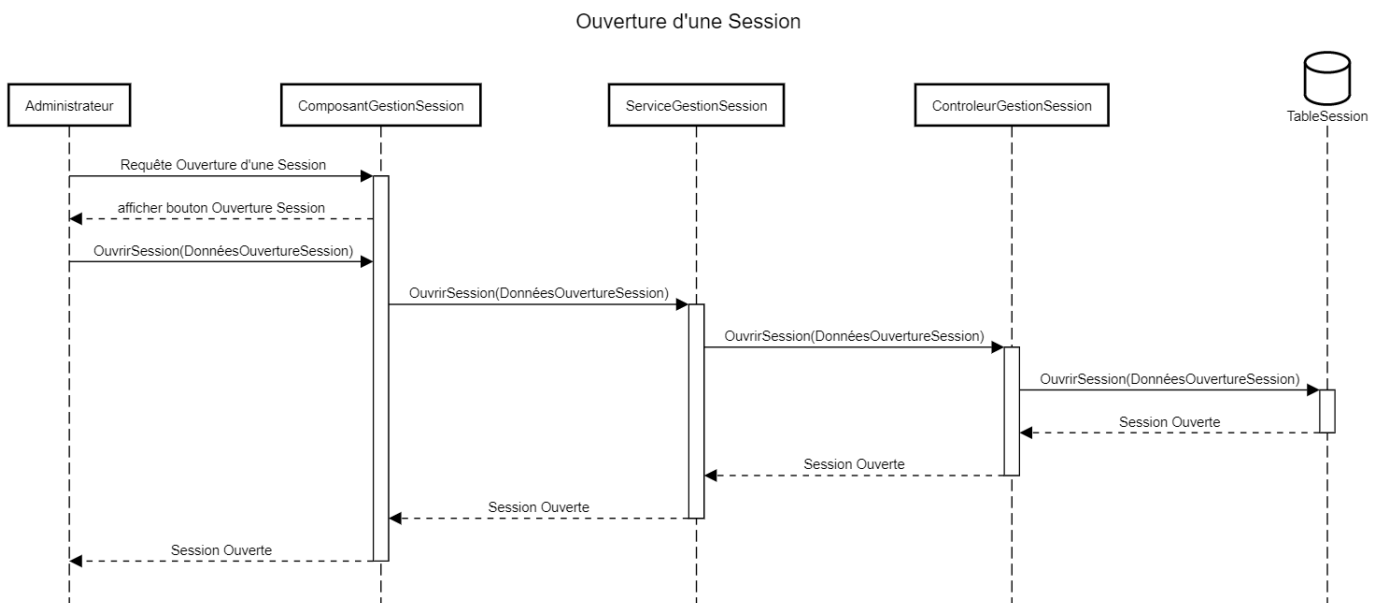


FIGURE 12 – Diagramme de séquence ouverture d'une session de choix

## Fermeture d'une session de choix

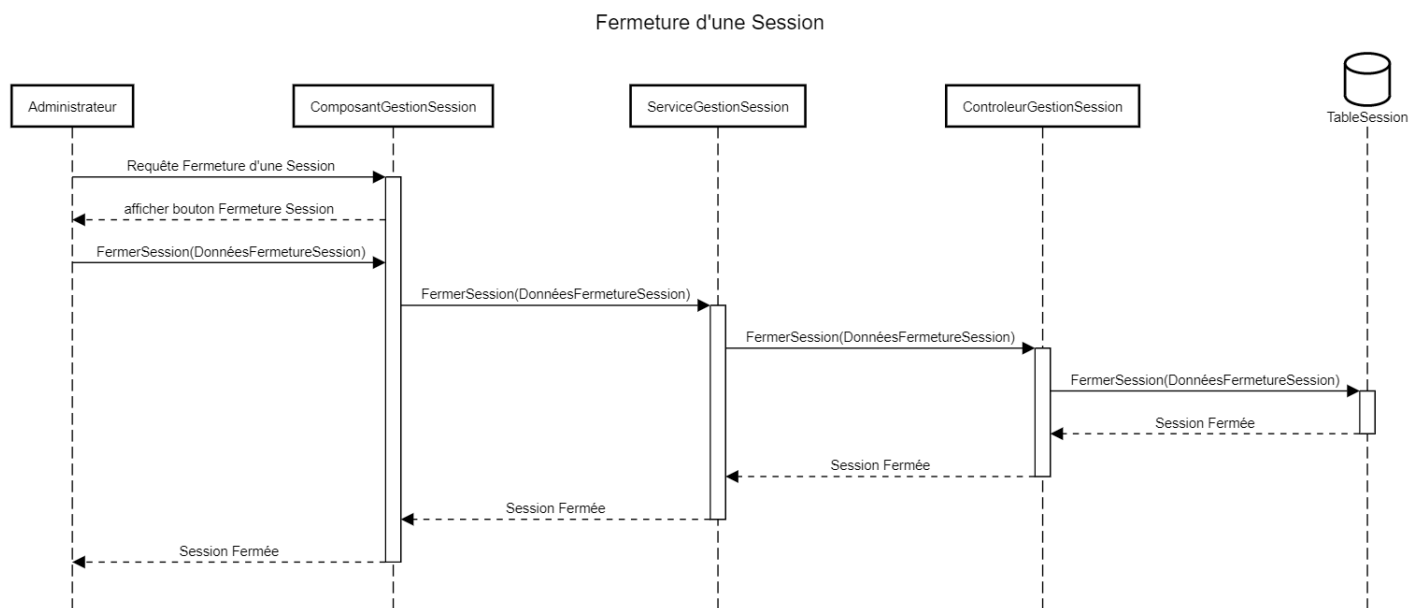


FIGURE 13 – Diagramme de séquence fermeture d'une session de choix

## Gestion des formations

### Ajout d'une formation

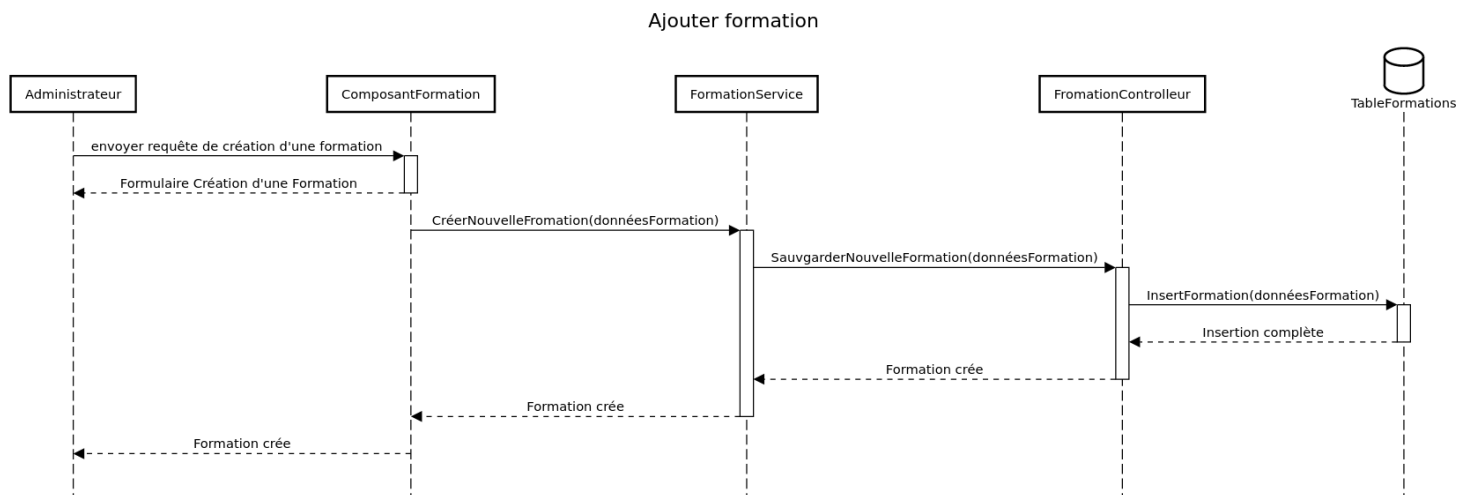


FIGURE 14 – Diagramme de séquence ajout d'une formation



## Mise à jour d'une formation

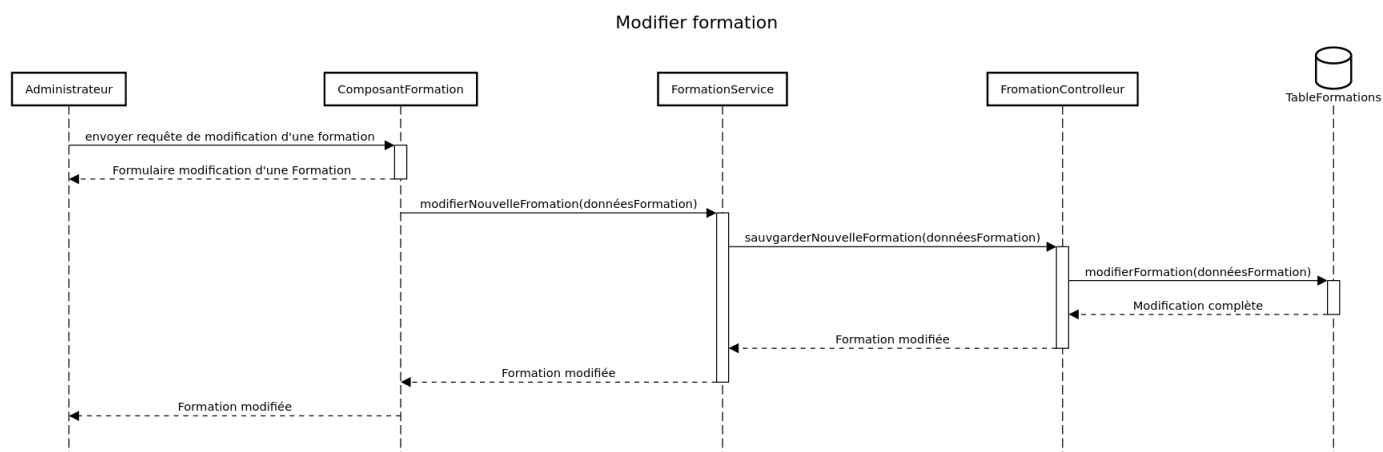


FIGURE 15 – Diagramme de séquence mise à jour d'une formation

## Suppression d'une formation

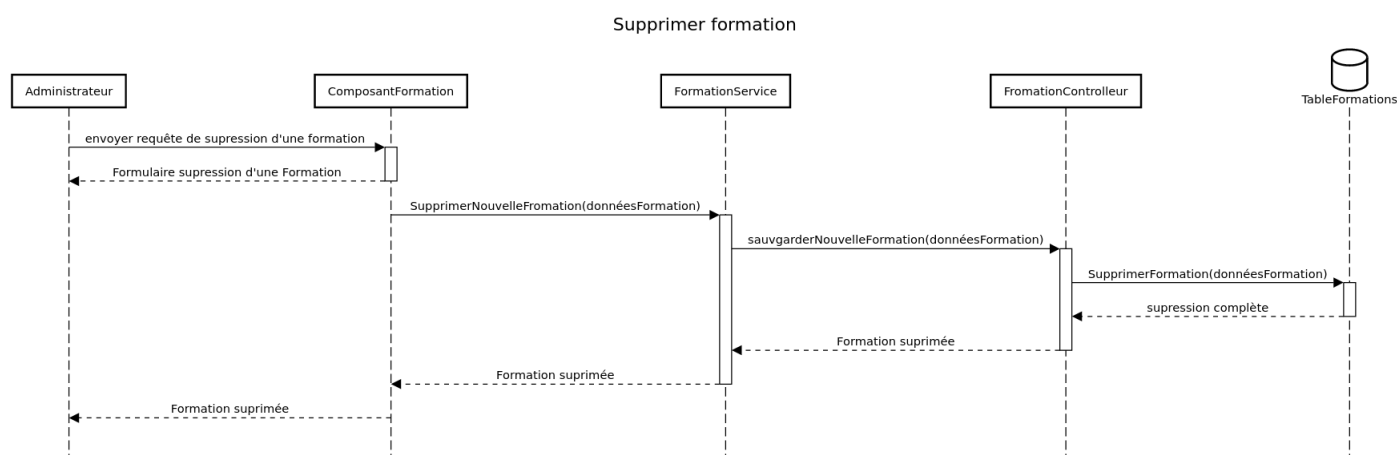


FIGURE 16 – Diagramme de séquence suppression d'une formation

# Tracabilité

Exigences dans la STB	Constituants du logiciel
STB-REA-01	Backend - Architecture générale du projet
STB-REA-02	Frontend - Architecture générale du projet
STB-REA-03	Backend - Architecture générale du projet
STB-QUA-01	Backend - Module des logs
STB-QUA-02	Frontend - composant des programmes horaires
STB-INT-01	Frontend - Interface homme machine
STB-INT -02	Frontend - Interface homme machine