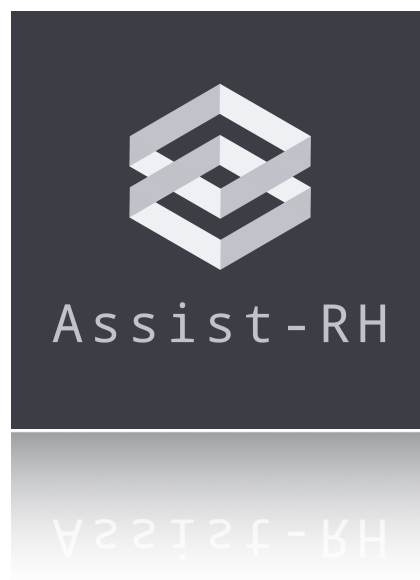


PROJET DE FIN D'ÉTUDES

Application web de gestion en ressources humaines :



Rudy TOURROU

José-Alexandre GIRY

Jimmy RAKOTOSON

Amine DARABID

1. Introduction	3
2. Étude	4
2.1. Définition du besoin	4
2.2. Définition des attendus	5
3. Organisation	8
3.1. L'équipe	8
3.2. Définition des tâches	9
3.3. Planification des tâches	10
4. Conception	13
4.1. Modélisation UML	13
4.2. Modélisation de l'application	17
4.3. Architecture physique de l'application	20
5. Réalisation	22
5.1. Développement des fonctionnalités	22
5.2. Écriture de la documentation technique	24
5.3. Tests	24
6. Livraison	25
6.1. Déploiement de l'application	25
7. Conclusion	27

1. Introduction

Dans le cadre de notre projet de fin d'études en mastère informatique, il nous a été demandé de concevoir et développer une application web dans les langages et technologies étudiées, en équipe de quatre personnes et sur une durée de quatre jours.

Ce dossier a pour objectif de présenter et justifier de :

- L'étude préalable et le cadrage du projet → définition des attendus client.
- L'organisation → planification de l'utilisation des ressources.
- La conception → modélisation graphique et fonctionnelle de l'application.
- La réalisation → développement des fonctionnalités et rédaction des documentations.
- La livraison → mise à disposition de l'application.

2. Étude

2.1. Définition du besoin

L'application dont il est sujet est un logiciel de gestion RH ou SIRH (système d'information de gestion des ressources humaines) où il est demandé de venir développer, dans le respect de certains attendus et contraintes techniques, un module de « formation » avec toutes les fonctionnalités qui l'incombent.

Ce module doit permettre de :

A. Gagner du temps sur la gestion des formations, en :

- Automatisant le calcul des coûts de formations.
- Permettant aux salariés et employeurs de s'exprimer.

B. Construire efficacement un plan de formation en :

- Maîtrisant le budget de formation.
- Suivant les dépenses selon leur type ou nature et dispositifs de financement.
- Évaluant les compétences acquises.

C. Faire participer les équipes dans le processus de formation en :

- Identifiant les besoins prioritaires de formation.
- Proposant des formations pertinentes.
- Évaluant les formations réalisées.

Il existe en outre trois rôles différents permettant des actions distinctes :

Employé

- S'authentifier.
- Consulter le catalogue de formations.
- Se voir proposer des formations selon son métier.
- Évaluer chacune de ses formations réalisées.

Manager

- Effectuer les mêmes actions que les employés.
- Gérer les demandes de formation de son équipe.
- Consulter l'historique de formation de son équipe.
- Gérer les compétences de son équipe.

Gestionnaire RH

- S'authentifier.
- Gérer le catalogue de formation.
- Gérer les budgets de formation.
- Valider les compétences une fois la formation réalisée.

Le processus métier est le suivant :

1. Le gestionnaire RH crée une formation qui remonte ensuite dans le catalogue RH.
2. L'employé consulte le catalogue de formation et peut faire la demande d'une formation.
3. La demande est transférée à son manager :
 - S'il la valide, l'employé en est informé et pourra réaliser cette formation.
 - Si elle est refusée, le salarié en est informé et reçoit ou consulte le motif de refus.
4. Chacun des acteurs doit se connecter avant de pouvoir agir sur les formations.

2.2. Définition des attendus

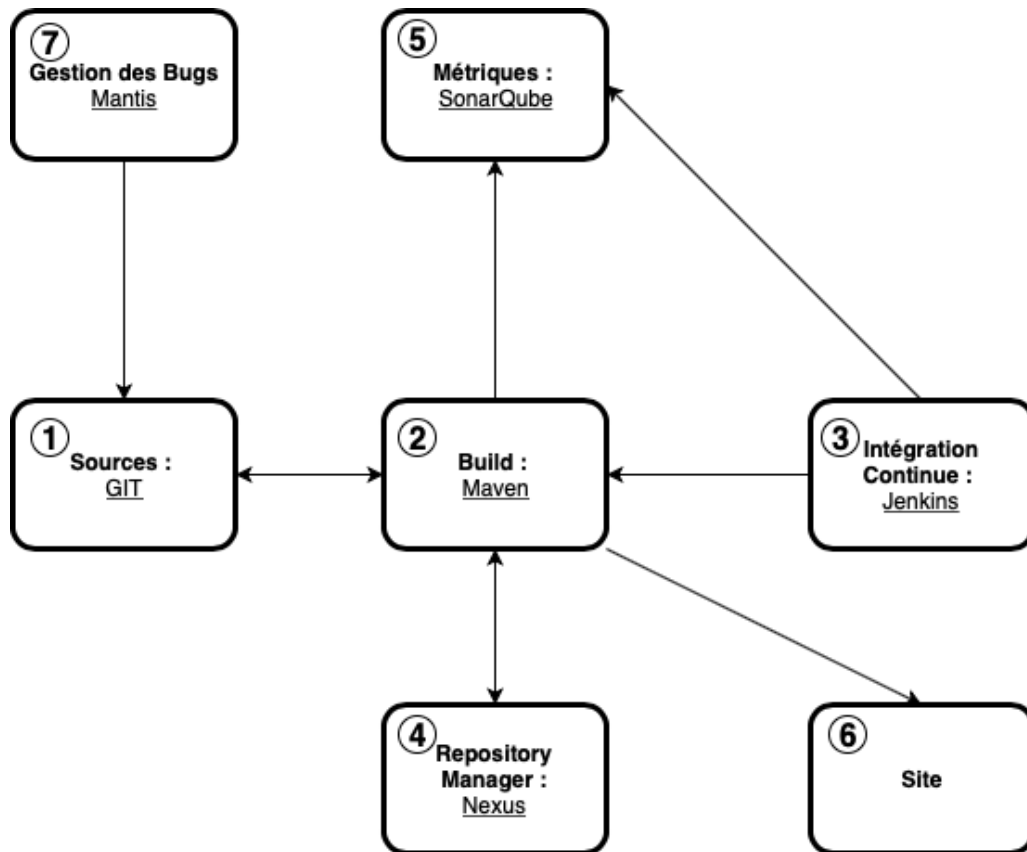
En plus des fonctionnalités citées ci-dessus, des exigences clients sont à prendre en compte.

Exigences technologiques

Outils de développement	
GitHub	Service web d'hébergement et de gestion de développement logiciel utilisant l'outil de gestion de versions Git.
Angular 7	Framework front-office.
WebService REST	Style d'architecture logicielle permettant l'interopérabilité entre ordinateurs sur internet.
JPA	Permet la persistance des données.
Spring Boot	Micro-framework permettant de générer l'architecture d'une application Java.
Spring Cloud	Configuration des microservices.
Déploiement de l'application sur le cloud	URL fournie à la fin du projet.

Intégration et déploiement continu

L'intégration et le déploiement continu permettent une automatisation de tâches dès que le code est modifié.



Exigences de qualité

► Ergonomie → l'ergonomie représente la disposition et l'accessibilité de chacune des fonctionnalités listées ci-dessus. Afin de répondre du mieux possible aux besoins de notre client, la navigation doit être la plus simple et intuitive possible. Nous allons donc présenter par la suite des maquettes de l'interface générale de notre application, point de départ pour le développement global de l'application.

► Fiabilité → la fiabilité d'une application se caractérise par sa capacité à fonctionner comme prévu, sans bug ou ralentissement altérant l'expérience utilisateur. C'est au travers d'un développement méthodologique et testé à chaque phase que nous évaluerons la fiabilité. Enfin, lors de la mise en production nous analyserons et comblerons les dernières anomalies éventuellement détectées.

▸ **Maintenabilité** → il s'agit de toujours penser aux possibles améliorations qui pourraient être implémentées par la suite, que ce soit des nouvelles fonctionnalités ou une modification d'interface. Cela implique donc une culture de codage où chaque fonctionnalité peut fonctionner indépendamment l'une de l'autre.

▸ **Portabilité** → l'application doit fonctionner sur les différents navigateurs web de nos clients. Un axe de d'amélioration possible sera de proposer une application « responsive », c'est à dire qui est compatible aux écrans de smartphones ou tablettes.

3. Organisation

3.1. L'équipe

L'équipe est constituée de quatre personnes issues de cursus scolaires similaires mais de métiers différents. Cette mixité des profils admet non seulement un échange des connaissances acquises dans les domaines de chacun, mais aussi un partage des points de vue et des expériences personnelles sur les sujets appréhendés.

▸ **José-Alexandre GIRY** → en tant que consultant SIRH de métier, José-Alexandre contribue de par ses qualités de conseil client à la retranscription des besoins et à la bonne détermination des attendus techniques. Sa vision à long terme et son savoir-faire informatique permettent d'instaurer une parfaite corrélation entre les retours terrain commerciaux et la finalité logicielle. Sa connaissance des métiers RH est un atout majeur dans ce projet et en fait un élément essentiel de l'équipe.

▸ **Rudy TOURROU** → fort de ses expériences en développement informatique et en conduite de projets dans des administrations publiques comme le ministère de la Défense ou d'autres grandes entreprises avec SUEZ et ERA Immobilier, Rudy apporte une vision claire et réaliste des chemins à prendre pour atteindre les objectifs techniques. Organisé et méthodique, il coordonne les équipes dans la réalisation des différentes tâches constitutives du livrable commercial.

▸ **Jimmy RAKOTOSON** → reconnu pour ses qualités de conseil auprès de clients dans le monde de l'assurance, Jimmy sera un maillon fort de notre projet. Ces connaissances en développement et son savoir purement métier seront des éléments indispensables à la bonne réalisation et entre en totale adéquation avec les qualités des autres membres de l'équipe.

▸ **Amine DARABID** → ayant déjà assuré la réalisation d'une multitude de projets, Amine viendra compléter les compétences déjà présentes dans l'équipe. Son expérience nous permettra d'avancer avec des objectifs toujours bien déterminés et cohérents avec l'attendu.

3.2. Définition des tâches

Il nous a été accordé pour ce projet quatre jours, du lundi 16 heures au vendredi 13 heures. D'un effectif de quatre personnes, nous avons une capacité de travail maximale de seize jours.

Les phases du projet ont été retranscrites dans ce tableau :

Phase	Explication	Nb jours / homme
Définition du projet	Découverte et réflexion sur le projet.	1
Pilotage	Action menées tout au long du projet afin de le cadrer.	1
Modélisation	Les interactions entre les acteurs et le système seront modélisées via le langage UML.	4
Préparation des environnements	Afin de maximiser la productivité de chacun des membres de l'équipes les environnements seront mis en place très tôt dans le projet.	0,5
Réalisation	Développement front et back.	12
Tests	Réalisation des tests pour notre application.	2
Documentation	Réalisation de la documentation de notre application.	0,5
Déploiement	Déploiement de notre application sur le cloud.	1
Total		22

Selon nos estimations, ce projet nécessite 22 jours homme. Possédant 16 jours au maximum, nous pouvons penser que le projet ne sera pas entièrement finalisé lors de sa livraison. De ce fait, nous allons approfondir le déroulement et le contenu de chacune de ses phases.

3.3. Planification des tâches

Définition du projet (1 jour / homme)

Tâches	Priorité	Ressources
Prise de connaissance du projet	/	Equipe
Définition du paramètre	/	Equipe
Analyse métier	/	Equipe
Analyse fonctionnelle	/	Equipe

Pilotage (1 jour / homme)

Tâches	Priorité	Ressources
Définition des tâches	/	Amine / José
Estimation du temps par tâches	/	Amine / José
Répartition des tâches	/	Amine / José
Suivi	/	Equipe

Modélisation (4 jours / homme)

Tâches	Priorité	Ressources
Diagrammes cas d'utilisation	/	Amine / José
Diagramme de classes	/	Equipe
Diagramme d'activité	/	Amine / José
Diagramme de séquence	/	Amine / José
Diagramme de déploiement	/	Amine / José
Plan du site	/	Rudy
Maquettage du site	/	Rudy

Préparation environnement (0,5 jour / homme)

Tâches	Priorité	Ressources
Préparation environnement	/	Equipe

Documentation (0,5 jour / homme)

Tâches	Priorité	Ressources
JavaDoc	/	Jimmy
ReadMe	/	Amine

Réalisation (12 jours / homme)

Tâches	Priorité	Ressources
Front		
Page Login	/	Rudy
Page Formation	/	Rudy
Page Demande en cours	/	Rudy
Page Historique	/	Rudy
Page Compétences	/	Rudy
Page Gestion des formations	/	Rudy
Page Plan de formation	/	Rudy
Page validation des compétences	/	Rudy
Page Budget annuel	/	Rudy
Back		
S'authentifier	1	Amine / José
Gérer formations	1	Jimmy / Rudy
Valider compétences	4	Jimmy
Gérer budget	4	Jimmy
Gérer plan de formation	4	Jimmy
Voir liste formation	1	Jimmy / Rudy
Demander formation	2	Jimmy / Rudy

Tâches	Priorité	Ressources
Voir historique formation	3	Jimmy
Gérer demande employé	2	Jimmy
Gérer compétences employé	4	Jimmy
/	/	Equipe

Test (2 jour / homme)

Tâches	Priorité	Ressources
Test unitaire	/	Jimmy
Test d'intégration	/	Equipe
Test de non régression	/	Jimmy
Test utilisateur	/	Equipe

Déploiement (1 jour / homme)

Tâches	Priorité	Ressources
Déploiement	/	Rudy

4. Conception

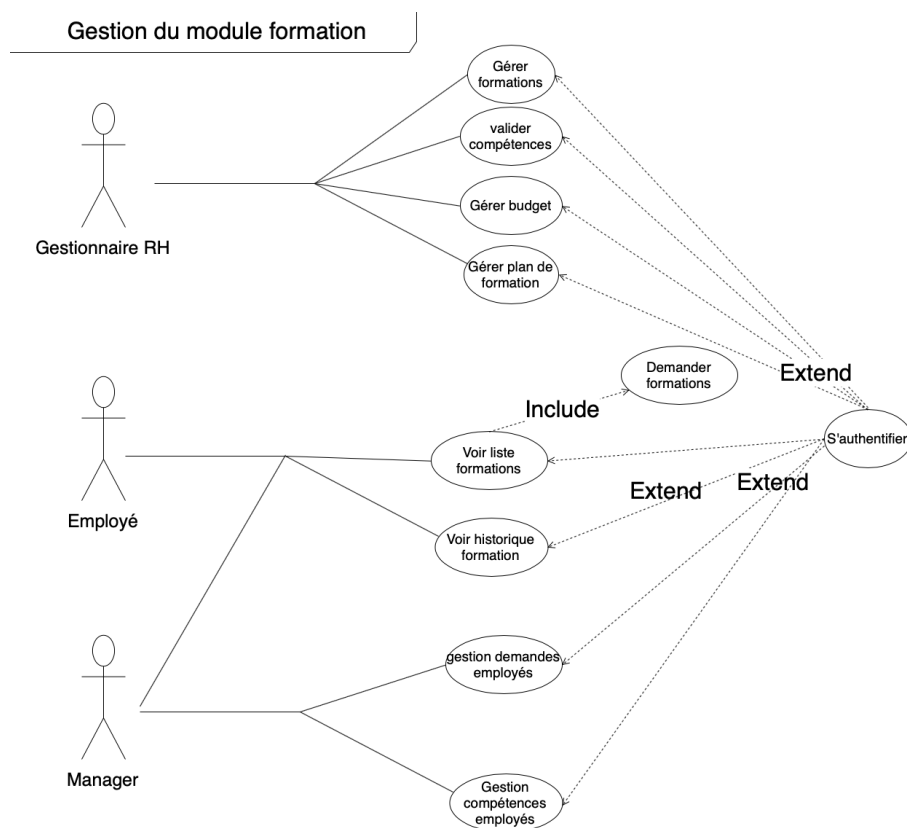
4.1. Modélisation UML

Le langage UML (*Unified Modeling Language*) est un standard de modélisation graphique. Il permet la visualisation de la conception d'un système. Nous avons utilisé ce langage afin modéliser notre application et produire les diagrammes suivants :

- Diagramme de cas d'utilisation
- Diagramme de classe
- Diagramme d'activité
- Diagramme de séquence
- Diagramme de déploiement

Diagramme de cas d'utilisation

Ce diagramme permet de modéliser les principales interactions entre les utilisateurs et le système.



Nous identifions trois acteurs : le gestionnaire RH, l'employé et le manager.

Le gestionnaire RH :

- Gère les formations
- Valide les compétences
- Gère le budget
- Gère le plan de formation

L'employé :

- Consulte la liste des formations
- Effectue des demandes de formation
- Consulte son historique de formation

Le manager :

- Gère les demandes de formations de son équipe
- Gère les compétences de son équipe

Diagramme de classe

Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations qu'elles entretiennent.

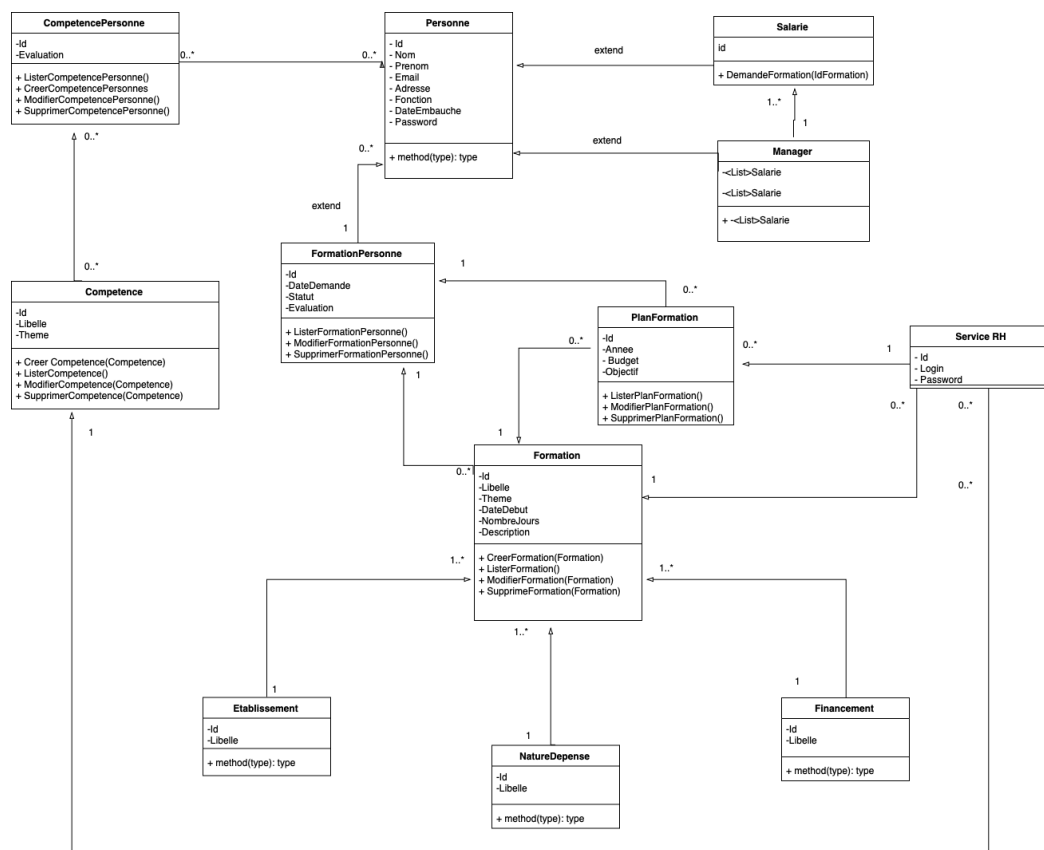
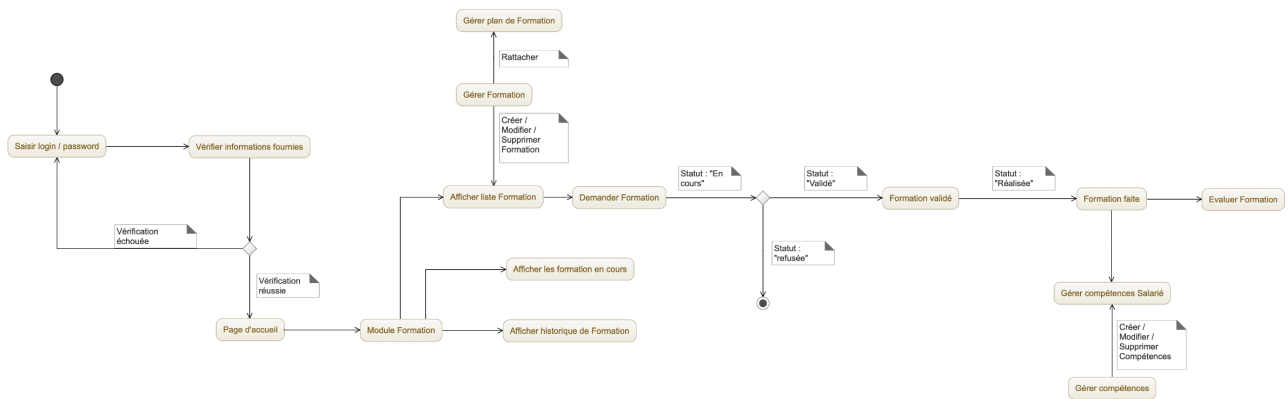


Diagramme d'activité

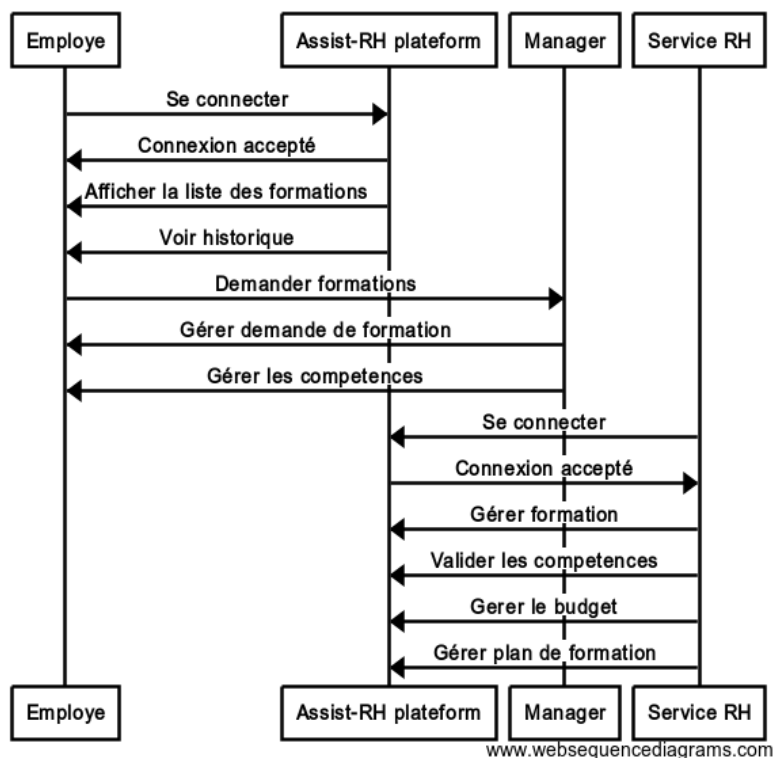
Le diagramme d'activité permet de modéliser le déclenchement d'événements en fonction de l'état du système. Il sert également à décrire un flux de travail.



Nous pouvons par exemple connaître le statut d'une demande de formation en fonction des évènements déclenchés dans le système.

Diagramme de séquence

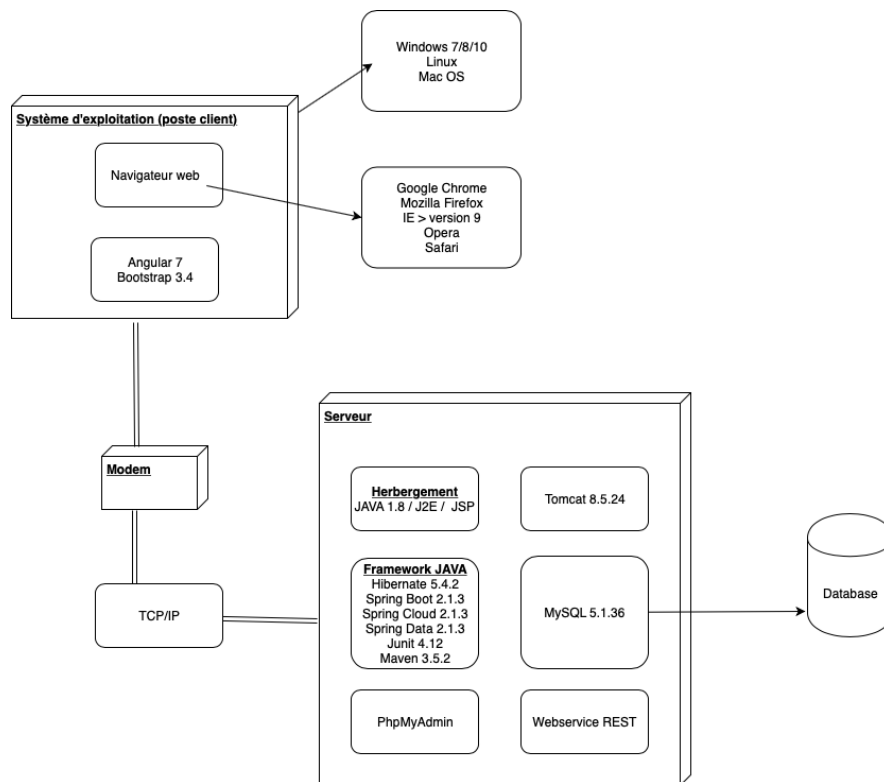
Le diagramme de séquence est la modélisation des interactions entre les acteurs et le système dans l'ordre chronologique.



Par exemple, afin de pouvoir faire une action, un acteur doit préalablement se connecter.

Diagramme de déploiement

Le diagramme de déploiement permet de représenter de façon statique l'infrastructure physique du système.

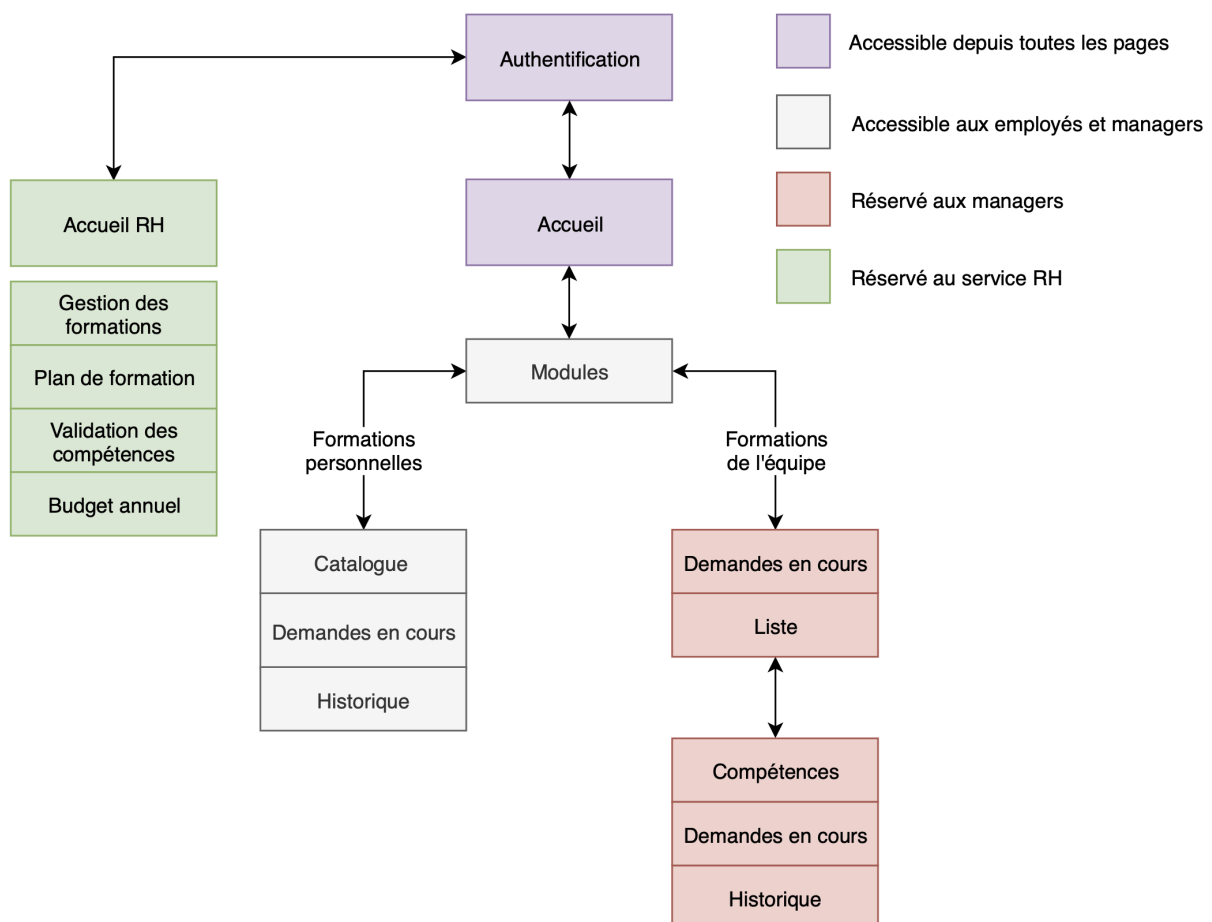


4.2. Modélisation de l'application

Il s'agit ici de décrire l'interface de notre application. Pour se faire, nous allons modéliser le plan du site, puis nous réaliserons les maquettes.

Plan du site

Le plan est la « carte » de l'application : toutes les pages y sont répertoriées et hiérarchisées ; leur accessibilité dans le déroulement logique du processus utilisateur est explicité.

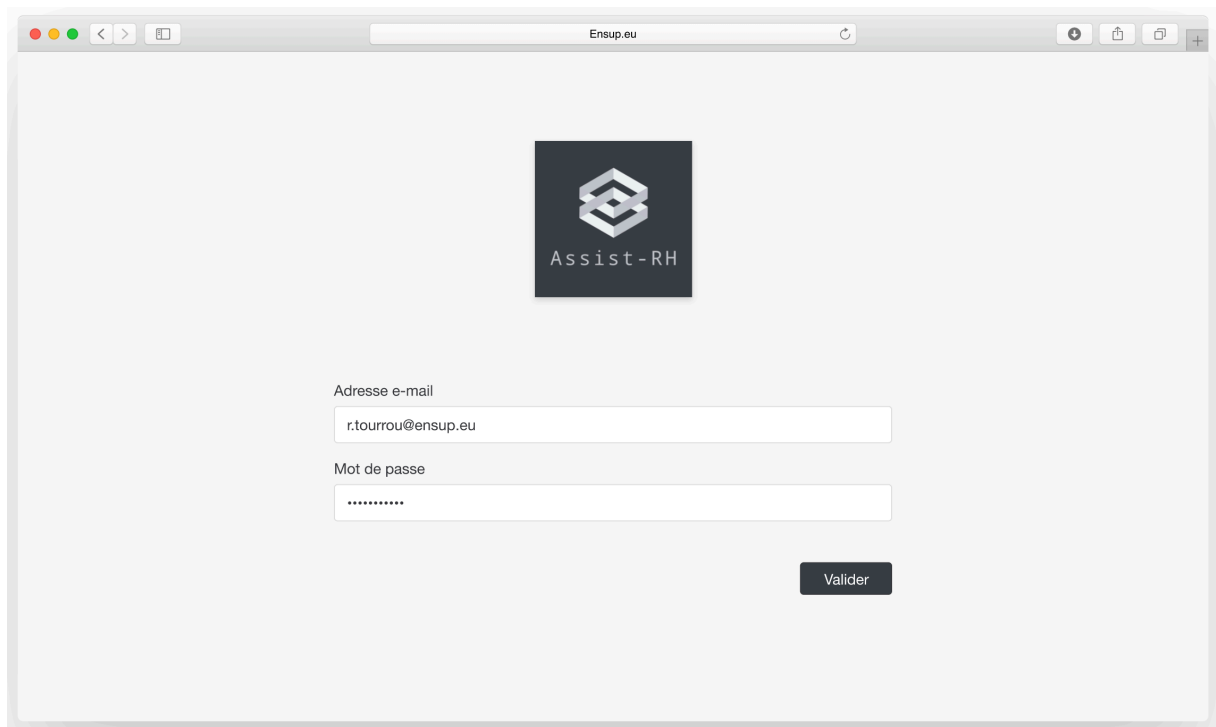


La première page accessible est la page authentification. Une fois connecté, chaque acteur à des fonctionnalités propres représentées par un code couleur.

Maquette de l'application

Cette étape consiste à réaliser les écrans sur un logiciel adapté, tels que les verront les utilisateurs finaux. Les maquettes serviront par la suite lors du développement de la partie front.

Afin de réaliser les maquettes graphiques, nous avons utilisé Adobe XD. Cet outil permet également la création de scénarios : en cliquant sur un bouton de la maquette, l'écran suivant correspondant s'affiche.



Déconnexion

Assist-RH

< Formation

CatalogueDemandes en coursHistorique

Rechercher un thème...

#	Thème	Date de début	Durée	Description	
837481	Comptabilité analytique	12 sept. 2019	5j	Étude de la rentabilité d'une entreprise.	
837482	Opérations de change sécurisés	28 sept. 2019	2j	Produits et règles de base des opérations.	
837483	Développement JAVA	03 oct. 2019	4j	Initiation au développements logiciels via le langage Java.	

< 1 2 3 >

Déconnexion

Assist-RH

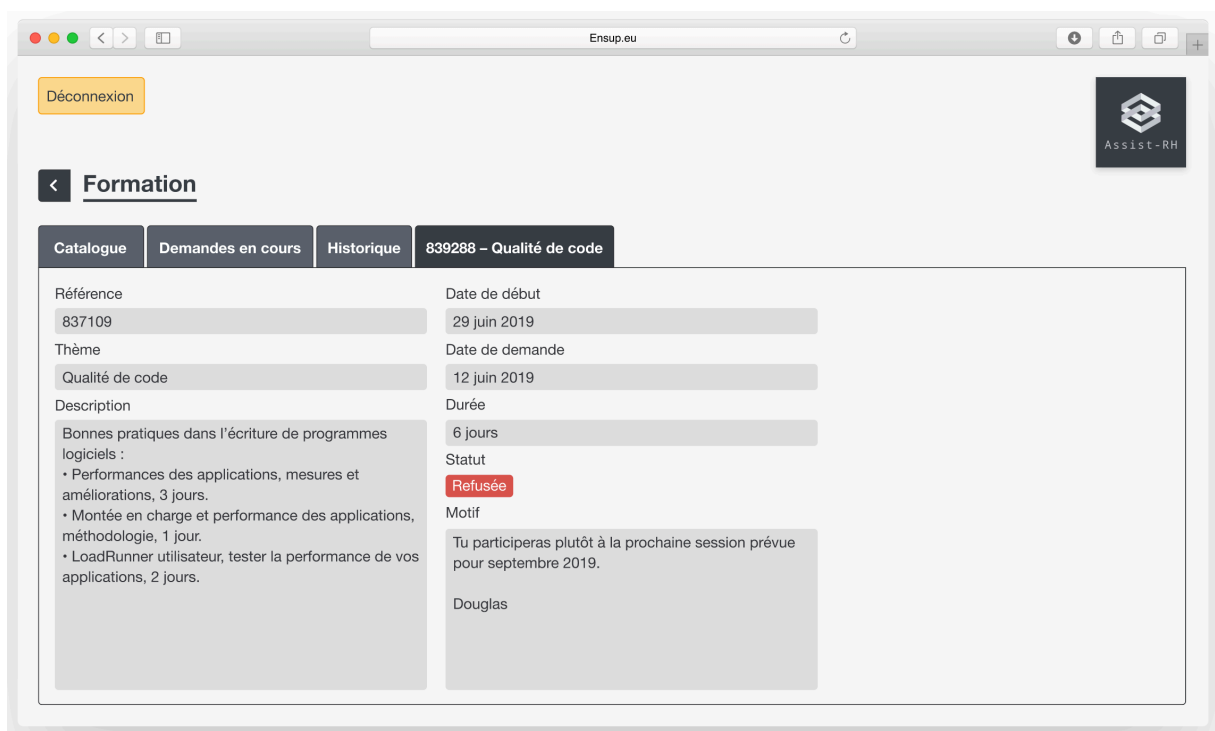
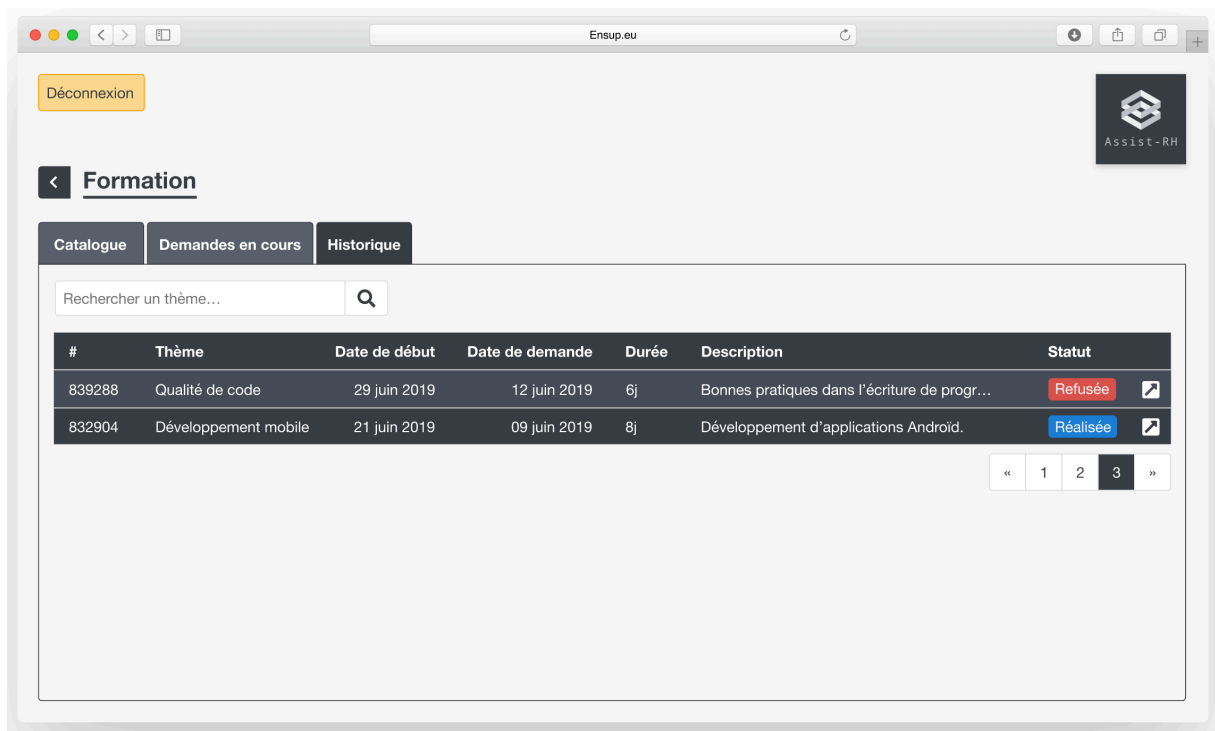
< Formation

CatalogueDemandes en coursHistorique

Rechercher un thème...

#	Thème	Date de début	Date de demande	Durée	Description	Statut	
837123	Sécurité informatique	09 sept. 2019	01 juil. 2019	4j	Concepts et méthodes de program...	Attente	
832499	Gestion de projet	14 sept. 2019	01 juil. 2019	3j	Méthodes de gestion agile de proje...	Refusée	
831233	Analyse du besoin	14 sept. 2019	01 juil. 2019	3j	Rédaction du cahier des charges...	Acceptée	

< 1 2 3 >



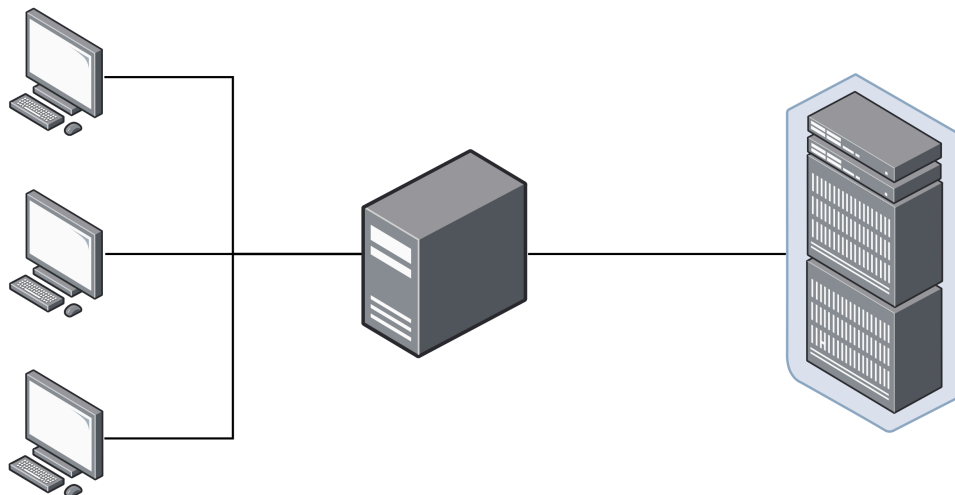
4.3. Architecture physique de l'application

L'architecture définie est connue sous le nom de « trois-tiers ». Elle est constituée de trois couches communiquant entre elles : la couche client, la couche serveur d'application et la couche serveur de base de données et est reconnue dans le monde professionnel de par sa bonne fiabilité, sécurité, disponibilité et évolutivité.

Pour l'hébergement de notre serveur d'application, notre choix s'est porté sur Amazon Web Services. En déposant simplement le fichier .war généré dans un répertoire dédié, l'application est déployée et accessible à l'URL suivante :

<http://assistrh-env-1.wmteyrppjf.us-east-1.elasticbeanstalk.com/connexion.html>

Pour l'hébergement de notre base de donnée, nous avons choisi la solution gratuite proposée par le site remotemysql.com qui répond entièrement aux besoins de l'application.



Architecture 3 tiers.

De gauche à droite : postes client, serveur d'application et base de données.

L'application en elle-même n'est déployée que sur la partie serveur : le client n'a besoin que d'un navigateur web compatible. Cette facilité de déploiement aura pour conséquence non seulement de réduire le coût de déploiement mais aussi de permettre une évolution régulière du système car elle nécessitera que la mise à jour de l'application sur le serveur applicatif.

5. Réalisation

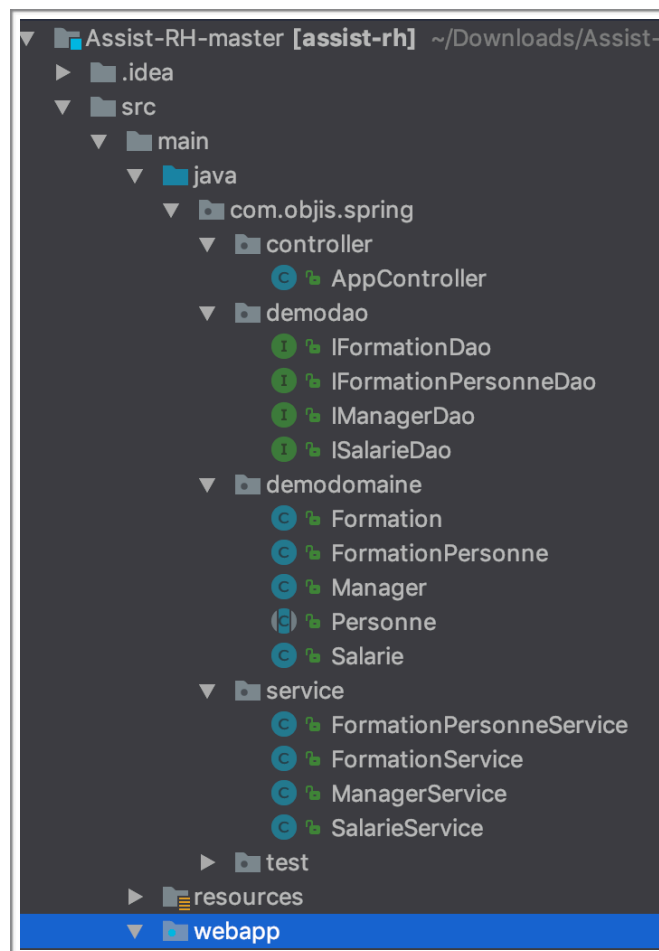
5.1. Développement des fonctionnalités

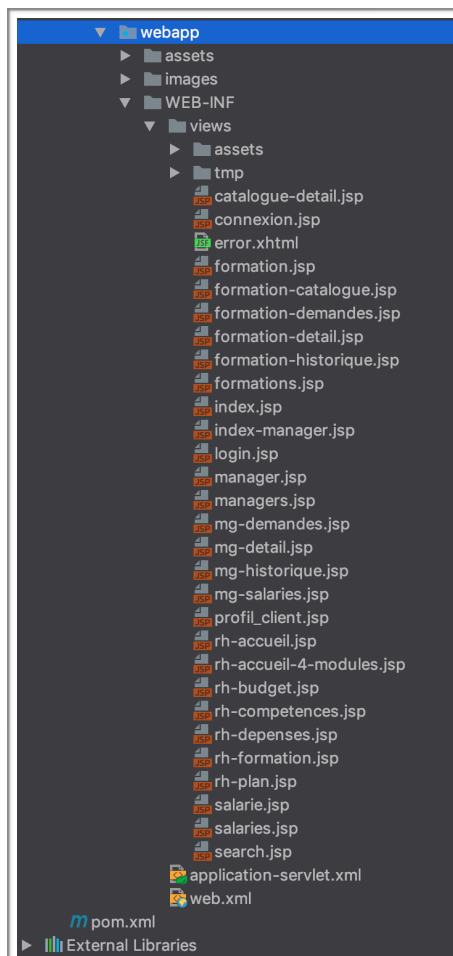
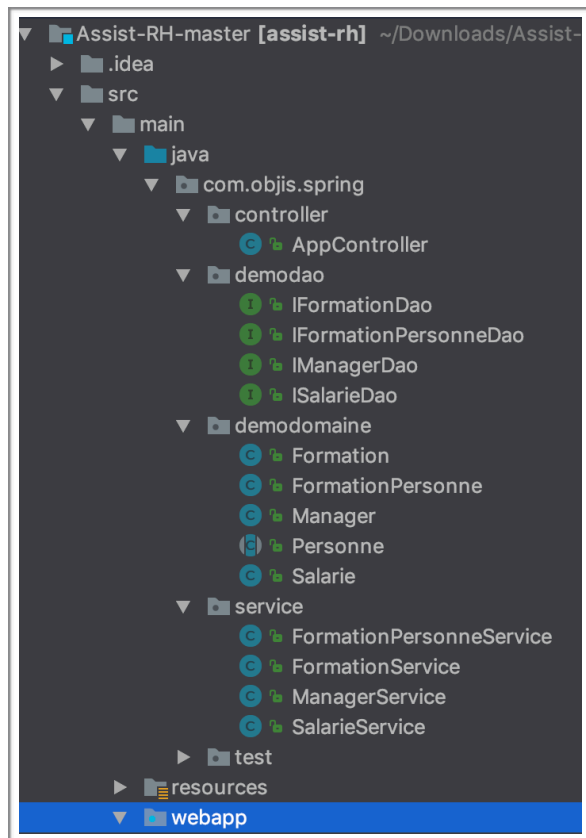
Architecture logicielle

Pour répondre aux besoins de l'architecture trois tiers, nous utiliserons le modèle MVC ; modèle - vue - contrôleur.

- **Modèle** : contient les données et la logique de traitement (validation, lecture et enregistrement) via les classes métier. Il vient suppléer le contrôleur et est complété par le DAO (ou objet d'accès au données) qui est un patron de conception gérant l'accès aux données dans notre project. Il correspond ici à la partie domaine.
- **Vue** : interface graphique qui sert à la présentation et la modification des données du modèle. Elle est matérialisée ici par les différents fichiers JSP du dossier webapp.
- **Contrôleur** : gère la partie logique de l'interaction homme machine.

Voici dans le détail leur composition :





5.2. Écriture de la documentation technique

JavaDoc

La JavaDoc, outil développé par Oracle donne la possibilité de créer une documentation d'API au format HTML. Cette documentation se réalise à partir des commentaires présents dans le code source de l'application, qui sont écrits tout au long du développement.

ReadMe

Le fichier ReadMe est un fichier dans lequel apparaît des informations relatives aux autres fichiers constitutif du projet.

5.3. Tests

Tests réalisés	
Tests unitaires.	À l'apparition d'une nouvelle fonctionnalité, celle-ci est testé afin de voir si l'action réalisée est conforme à l'attendu.
Tests de non-régression.	Au fur et à mesure que notre site internet prend de l'ampleur et possède de multiples fonctionnalités, celles-ci sont à tester indépendamment dans l'environnement. Cela permet de vérifier qu'un nouvel ajout n'a pas détérioré une fonctionnalité déjà présente.
Tests de bout-en-bout.	Consiste à tester un process précis. Par exemple la demande et la validation de la demande de formation: faire la demande de formation par l'employé, gestion de cette demande par le manager, réponse visible par l'employé.

6. Livraison


6.1. Déploiement de l'application

L'un des pré-requis projet initiaux concernait la mise à disposition de l'application sur une adresse internet accessible depuis l'extérieur. Il a donc fallu sélectionner un hébergeur pour la base de données et l'application.

Application web


Nous avons utilisé l'offre « Elastic Beanstalk » d'Amazon Web Services, un des leaders du domaine du cloud computing à la demande, qui permet de déployer des applications et services web développés avec différents langages populaires comme Java, .NET, Node.js...

Il s'agit de venir déposer un fichier .zip ou .war du projet, le déploiement et la mise en ligne étant totalement automatique et optimisée.



Santé
OK
[Causes](#)

Version en cours d'exécution
assist-rh-source-2
[Charger et déployer](#)


Configuration
Tomcat 8.5 with Java 8 running
on 64bit Amazon Linux/3.1.6
[Modification](#)

Événements récents
[Afficher tout](#)

Horaires	Type	Détails
05-07-2019 11:47:28 UTC+0200	INFO	Environment health has transitioned from Info to Ok. Application update completed 32 seconds ago and took 18 seconds.
05-07-2019 11:46:28 UTC+0200	INFO	Environment health has transitioned from Ok to Info. Application update in progress on 1 instance. 0 out of 1 instance completed (running for 10 seconds).
05-07-2019 11:46:04 UTC+0200	INFO	Environment update completed successfully.
05-07-2019 11:46:04 UTC+0200	INFO	New application version was deployed to running EC2 instances.
05-07-2019 11:45:46 UTC+0200	INFO	Deploying new version to instance(s).


Base de données

L'hébergement de la base s'est fait sur un site gratuit, remotemysql.com. Les offres d'Amazon ont été jugées trop complexes et coûteuses par rapport aux dimensions projet.

Le site permet la mise en ligne de deux bases de données au maximum. Il fournit ensuite les identifiants pour l'accès à distance via des logiciels ou propose un accès direct via phpMyAdmin.

Databases

All your databases are listed below. Create new ones, upto a maximum of 3, with the button at the bottom.

USERNAME	SIZE	CREATED DATE	ACTION
nWAsOuCXUr	0.01 MB	2019-07-03 09:21:12	 ACTION

 CREATE NEW DATABASE

Statistics

DATABASES

User/Database Name	nWAsOuCXUr
Database size	0.01 MB
Total Connections	1288
Current Connections	0
Total Connected Time	63517.1 seconds
Total CPU time	296.9 seconds
Data Recieved	3.04 MB
Data Sent	0.07 MB
Rows fetched	15272
Rows Updated	176
Select Commands	6616
Update Commands	453

7. Conclusion

À travers ce dossier, nous avons décrit le processus de création de l'application Assist-RH.

Il a d'abord fallu prendre connaissance du sujet et évaluer un périmètre d'action. Nous nous sommes ensuite organisés et répartis les tâches en fonction de la charge, des délais accordés et des points forts de chacun. Nous avons conceptualisé l'application avec des outils et langages de modélisation (Adobe XD, UML) ce qui nous a conféré une vision claire de l'application finale.

La phase de développement a concrétisé notre travail. Des livraisons sur le service cloud d'Amazon ont été régulièrement effectuées, ainsi qu'une batterie de tests.

Pendant toute la durée du projet, nous avons réfléchi aux axes possible d'amélioration et en avons détecté un certain nombre, comme par exemple l'amélioration des contrôles de saisie utilisateur sur certaines parties de l'application. C'est dans cet objectif que notre architecture logiciel a été défini : elle est évolutive et facilement maintenable.

Nous avons su partager lors de ce projet en groupe nos idées et agir malgré des avis parfois divergents. De nombreuses compétences informatiques acquises ces deux dernières années ont pu être mises en oeuvre. Nous avons également été sensibilisés aux processus métiers relatifs à la gestion des formations.