

Chapitre 1

Release 1

1.1 Introduction

Le terme release désigne ici une version de notre application constituée d'une suite d'itérations qui se terminent quand les incréments de ces derniers construisent un produit présentant suffisamment de valeur à l'utilisateur.

Dans ce chapitre, nous nous intéressons principalement par l'application web ,et nous allons nous intéresser à la première release dans laquelle nous traitons en détails chacun des cas d'utilisation préalablement présentés. Ainsi ,selon le sprint ,on précisons l'analyse qui consiste à illustrer le diagramme de cas d'utilisation ,la conception pour présenter le diagramme de séquence et le schéma de base de données et nous finissons par le test en présentant les interfaces correspondantes.

Pour chaque sprint on va présenter le cas d'utilisation,la conception qui consiste à présenter le diagramme de séquence et nous finissons par l'illustration du schéma de la base de données des modules correspondants.

Ainsi ,contrairement aux cycles de développement séquentiel, avec la méthodologie agile, le test n'est pas une phase qui se déroule après la fin de développement. En effet, les tests seront intégrés dès le début du premier sprint jusqu'à la livraison du produit final.

Enfin ,nous illustrons la méthode de liason de données que nous avons suivi ainsi nous présentons le diagramme de classes ,le diagramme d'activité et le diagramme de déploiement pour mieux expliquer notre application.

1.2 Sprint 1

1.2.1 Analyse

Diagramme de cas d'utilisation "Gérer un membre"

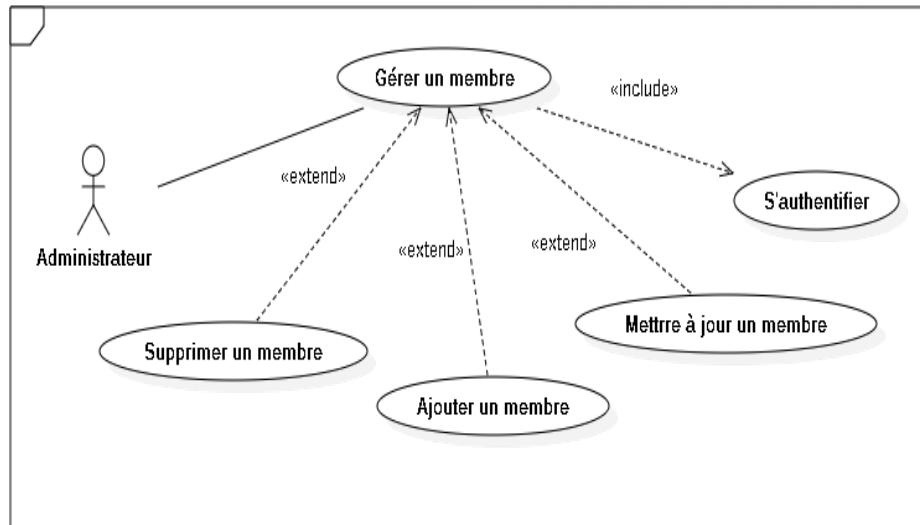


FIGURE 1.1 – Gérer un membre.

1.2.2 Conception

Le diagramme de séquence indique l'interaction entre plusieurs acteurs afin d'expliquer le déroulement des différents scénarios entre les différents éléments du projet.

Le scénario « Création d'un membre »

Le diagramme de séquence « Ajout d'une tâche » présente le séquençement des interactions entre Administrateur, Application et Base de données (BD).

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
login	varchar(30)	YES		NULL	
password	varchar(15)	YES		NULL	
firstname	varchar(50)	YES		NULL	
lastname	varchar(30)	YES		NULL	
email	varchar(100)	YES		NULL	
hiredate	datetime	YES		CURRENT_TIMESTAMP	
color	varchar(6)	YES		NULL	
projects_id	int(11)	YES	MUL	NULL	
pname	varchar(50)	YES		NULL	

TABLE 1.1 – Members.

1.2.3 Schéma

La table « members »

1.2.4 Test

Dans ce premier sprint, nous testons la fonctionnalité « Gestion des membres » qui affiche, ajoute, supprime les membres de la base de données. Ci-dessus nous ajoutons des captures écran du sprint réalisé.

Lors de la création d'un utilisateur , un email contenant son mot de passe lui sera envoyé, ainsi si on modifie les paramètres d'utilisateur , un nouveau mot de passe lui sera envoyé sur la nouvelle adresse email .

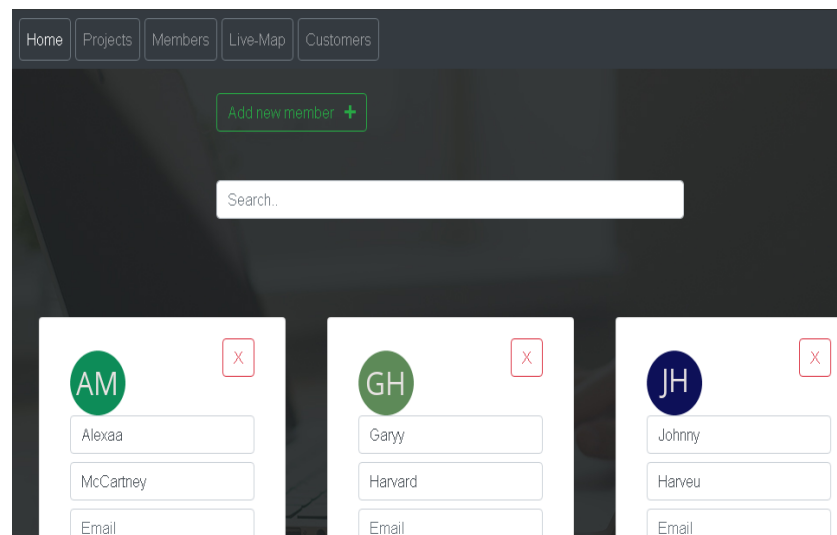


FIGURE 1.2 – Gestion des membres.1.

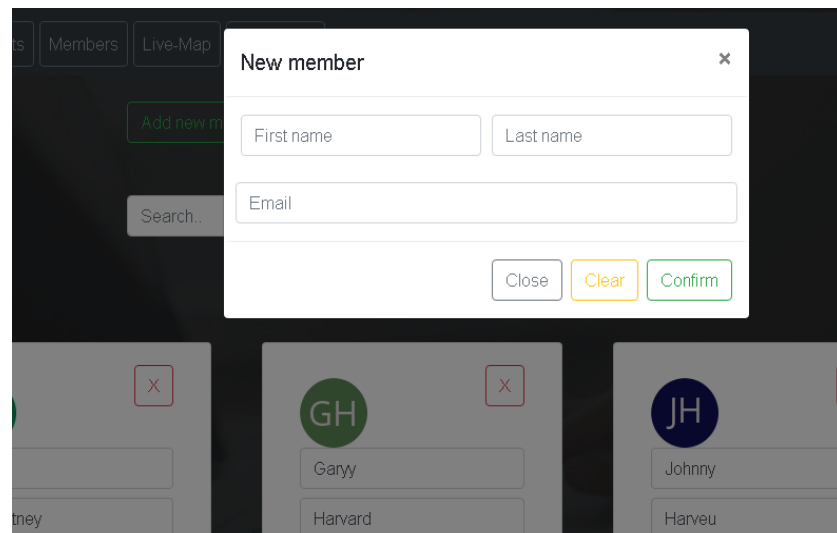


FIGURE 1.3 – Gestion des membres.2.

AM

X

Alexaa

McCartney

nouveauemail@gmail.com

Hire date:

2019-05-29

Current project :

Video Making

save

FIGURE 1.4 – Gestion des membres.3.

1.3 Sprint 2

1.3.1 Analyse

Diagramme de cas d'utilisation "Gérer un client"

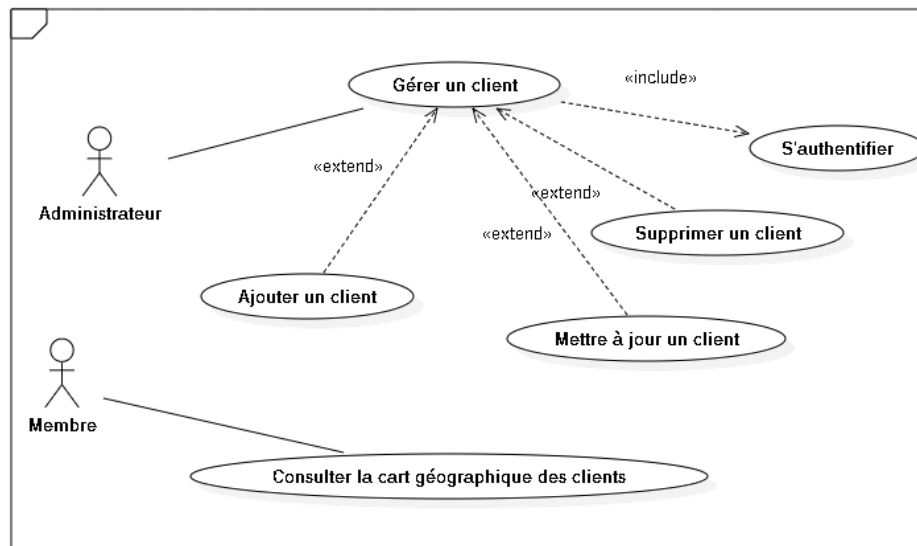


FIGURE 1.5 – Gérer un client.

1.3.2 Conception

Le scénario « Création d'un client »

Le diagramme de séquence « Ajout d'une tâche » présente le séquençement des interactions entre Administrateur, Application et Base de données (BD).

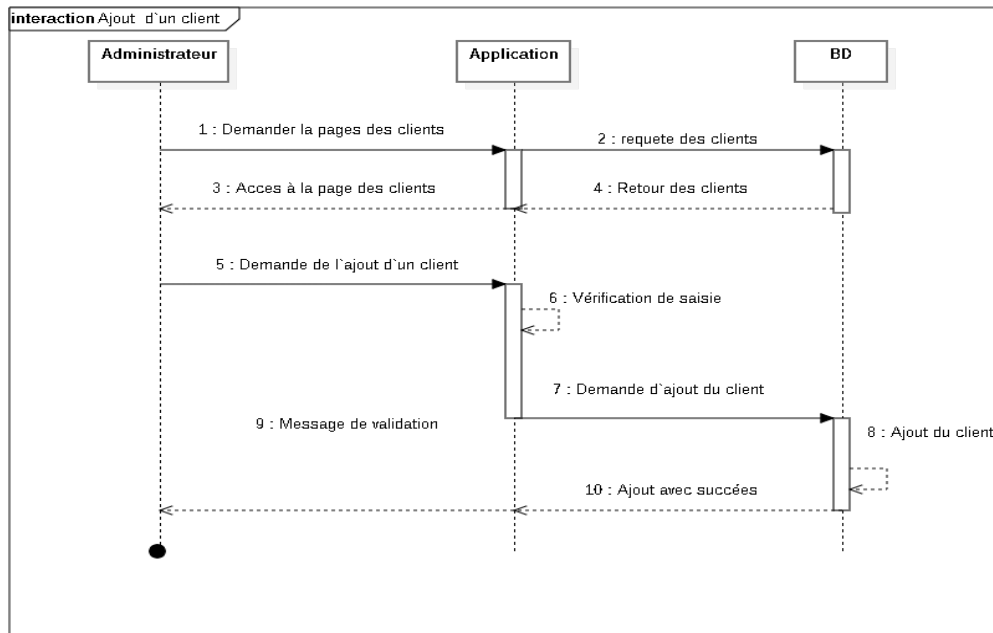


FIGURE 1.6 – Création d'un client.

Le scénario « Consultation de la carte géographique des clients»

Le diagramme de séquence « Ajout d'une tâche » présente le séquençement des interactions entre Administrateur, Application et Base de données (BD).

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(45)	YES		NULL	
phone	varchar(45)	YES		NULL	
address	varchar(1000)	YES		NULL	
email	varchar(50)	YES		NULL	
longitude	double	YES		NULL	
latitude	double	YES		NULL	
subscriptionDate	datetime	YES		CURRENT_TIMESTAMP	

TABLE 1.2 – Customers

Schéma

La table « customers »

1.3.3 Test

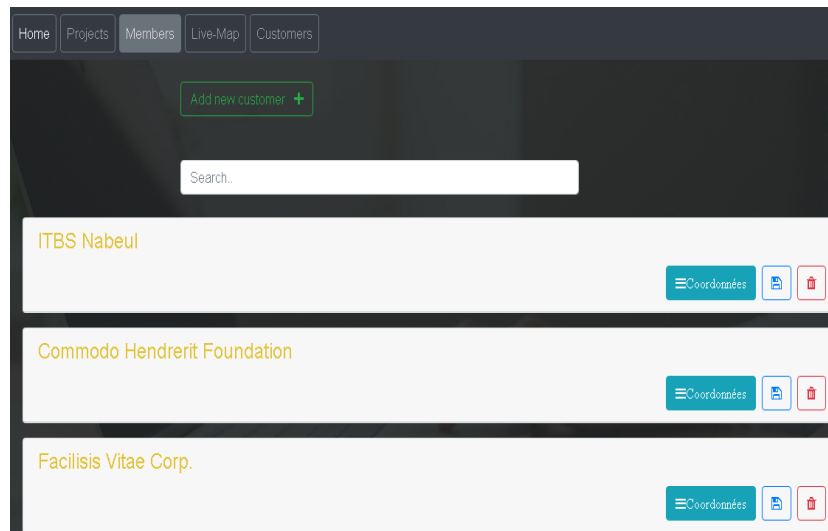


FIGURE 1.7 – Gestion des clients.1.

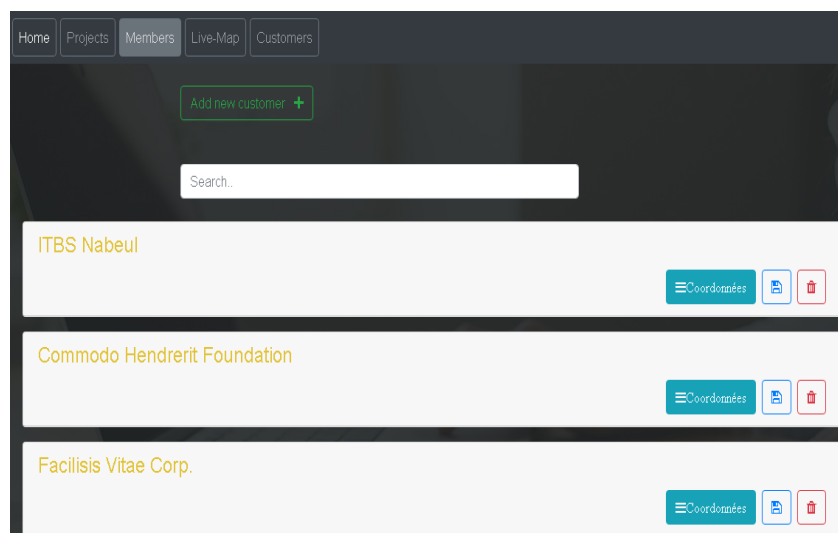


FIGURE 1.8 – Gestion des clients.2.

1.3.4 Consultation carte géographique des clients

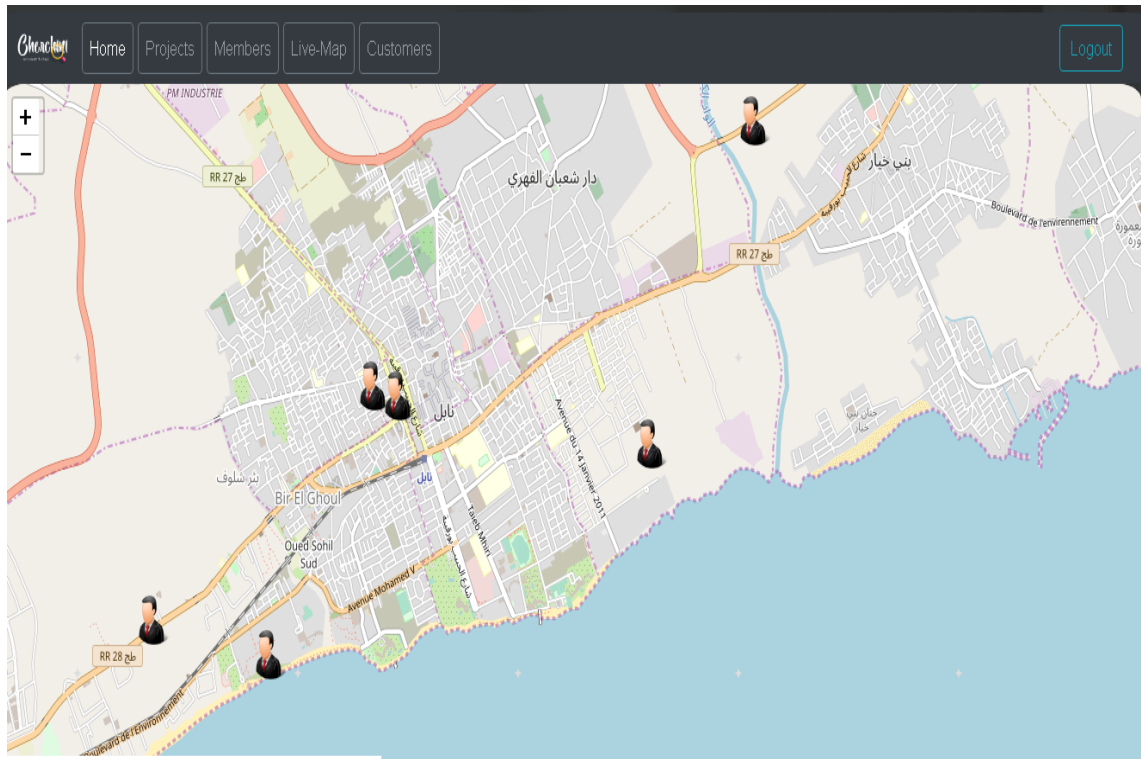


FIGURE 1.9 – Carte géographique.

1.4 Sprint 3

Lors de ce sprint , nous raffinons le cas d'utilisation "gestion du projet" .
Détaillez le cas d'utilisation "Gestion du projet " revient à détailler ses sous cas d'utilisation à savoir :

- Gestion des tâches.
- Gestion des projets

1.4.1 Analyse

1.4.2 Diagramme de cas d'utilisation "Gérer un projet"

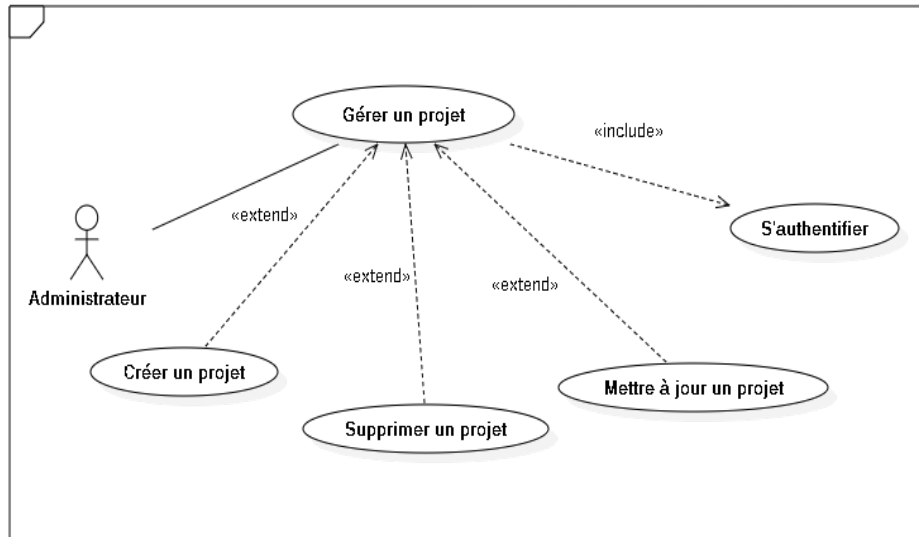


FIGURE 1.10 – Gérer un projet.

Diagramme de cas d'utilisation "Gérer une tâche"

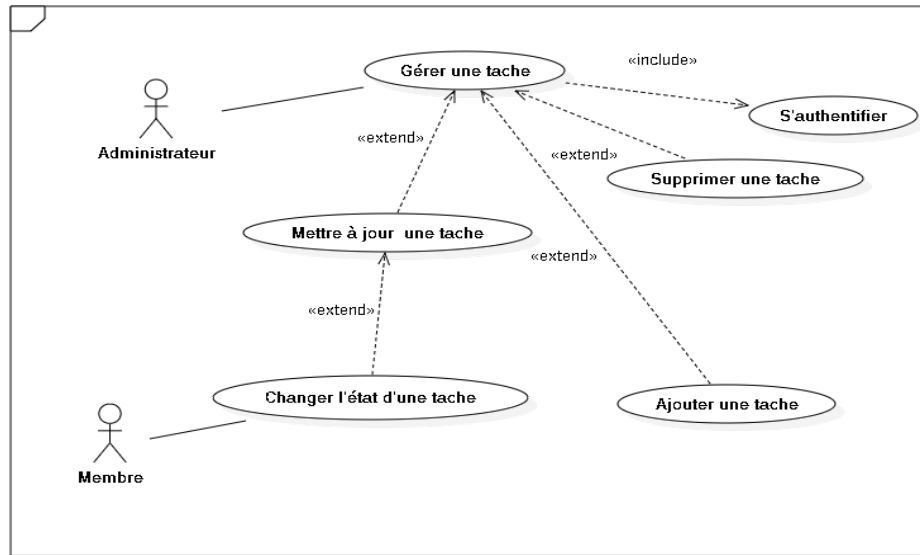


FIGURE 1.11 – Gérer une tâche.

1.4.3 Conception

1.4.4 Le scénario « Création d'un projet »

Le diagramme de séquence « Ajout d'un projet » présente un séquençement des interactions entre Administrateur, Application et Base de données (BD).

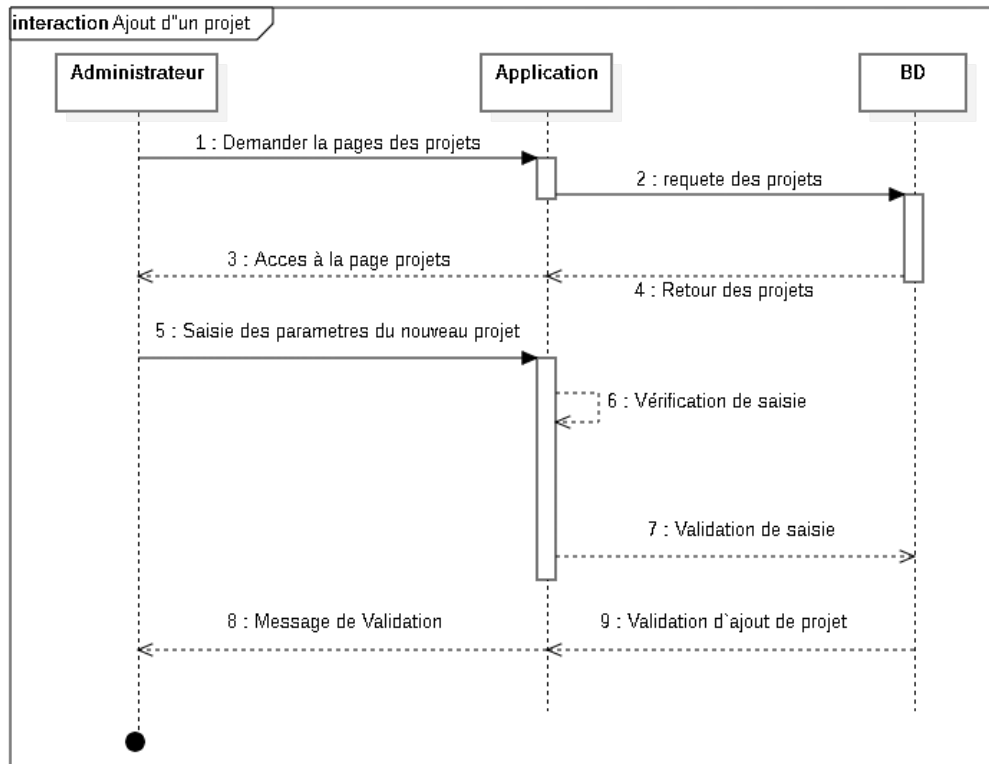


FIGURE 1.12 – Création d'un projet.

1.4.5 Le scénario « Création d'une tâche »

Le diagramme de séquence « Ajout d'une tâche » présente le séquençement des interactions entre Administrateur, Application et Base de données (BD).

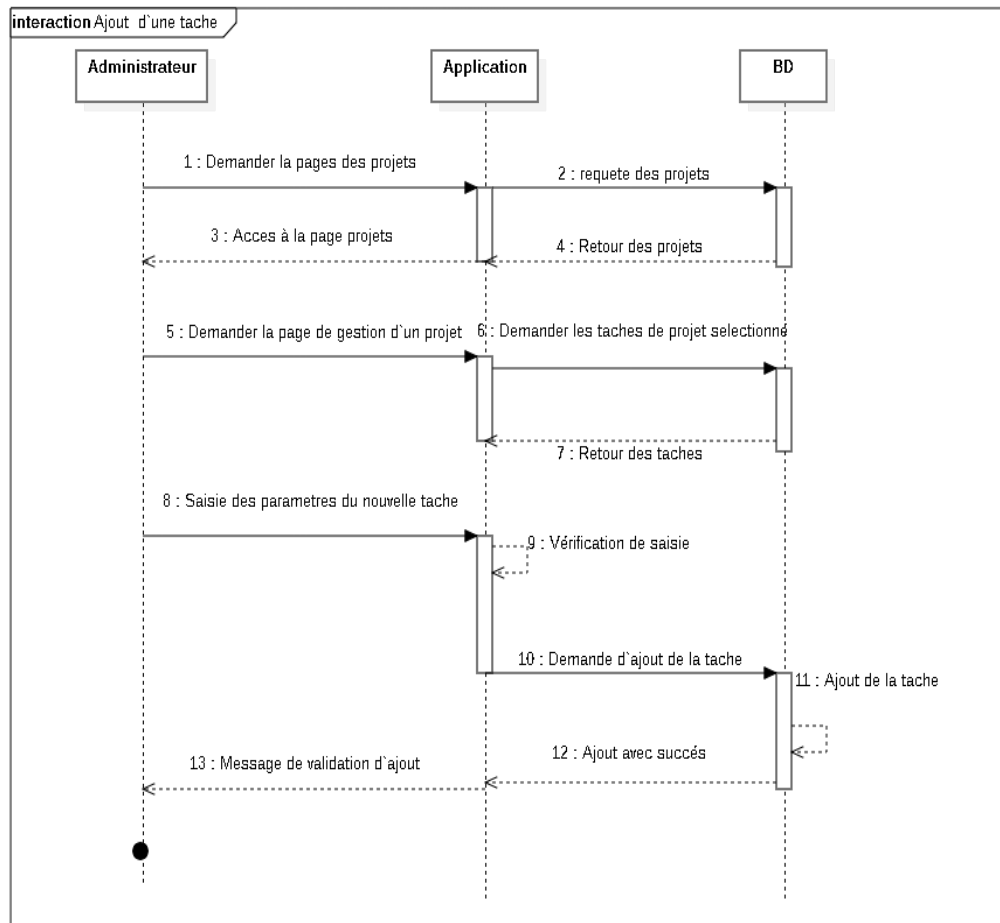


FIGURE 1.13 – Création d'une tâche.

1.4.6 Schémas

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(45)	NO		NULL	
description	varchar(1000)	YES		NULL	
start	date	NO		NULL	
end	date	NO		NULL	
category	varchar(60)	YES		NULL	
customers_id	int(11)	YES	MUL	NULL	

TABLE 1.3 – Projects.

1.4.7 Test

C'est l'interface principale de notre application

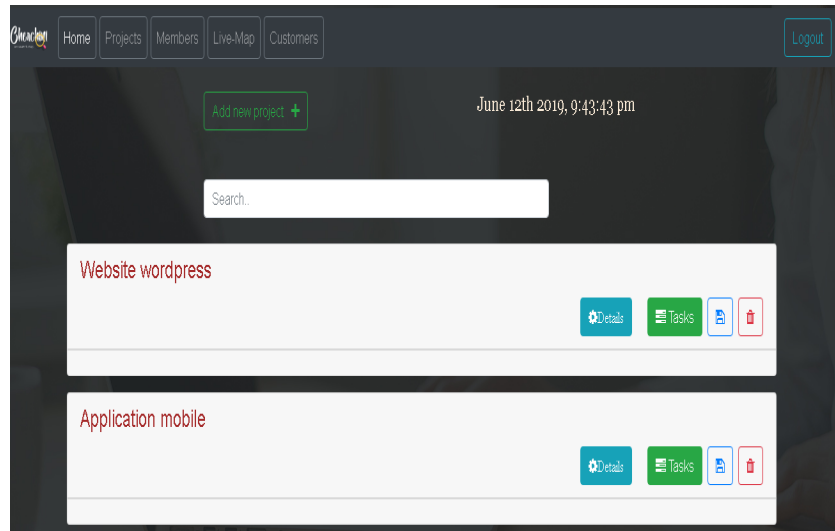


FIGURE 1.14 – Gestion de projet.1.

Nous ajoutons un nouveau projet et l'assignons un ou plusieurs membres.

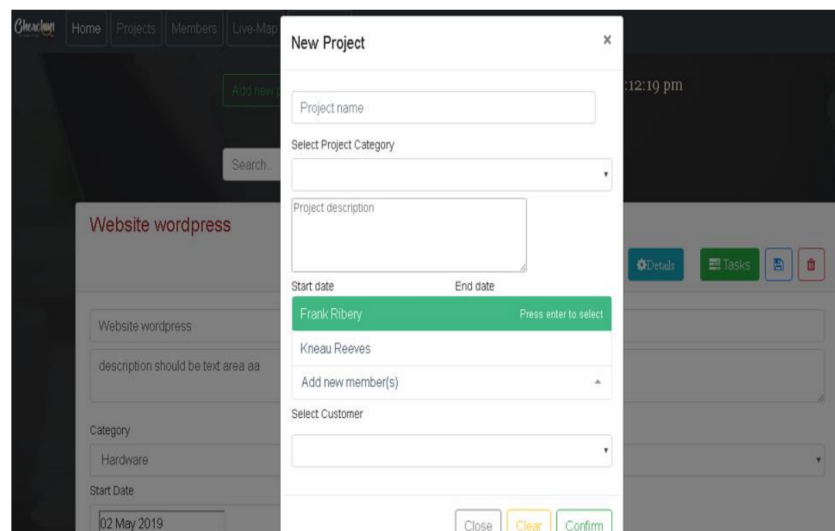


FIGURE 1.15 – Gestion de projet.2.

The screenshot shows a web application interface for managing a project titled "Website wordpress". At the top right, there are four buttons: "Details" (blue with a gear icon), "Tasks" (green with a list icon), and two smaller icons (blue and red). Below the title bar, there is a text input field containing "Website wordpress" and a larger text area below it containing "description should be text area ..". Underneath the text area is a "Category" dropdown menu currently set to "Hardware". Below the category are two date pickers: "Start Date" showing "02 May 2019" and "End Date" showing "04 Jul 2019". At the bottom, there are two lists of members. The "Available members" list contains "Frank Ribery" and "Kneau Reeves". The "Current Project Members" list contains "Johnny Harveu", "Wael Chorfan", and "David Alaba". At the very bottom, there are two buttons: "Add selected members" (blue) and "Delete selected members" (red).

FIGURE 1.16 – Gestion de projet.3.

Pour chaque projet en cliquant sur le buttons Details ,on peut modifier ses détails ,on peut sélectioner plusieurs membres et les ajouter . On ne peut pas modifier les dates de début et de fin puisque les dates des taches devraient etre pénibles à changer tache par tache .

En cliquant sur le button « Tasks » d’un certain projet , l’administrateur est amenée à l’interface de gestion des taches correspondantes dans laquelle il peut suivre la progression des taches ,consulter le diagramme de gannt dynamique et modifier les paramètres des taches .

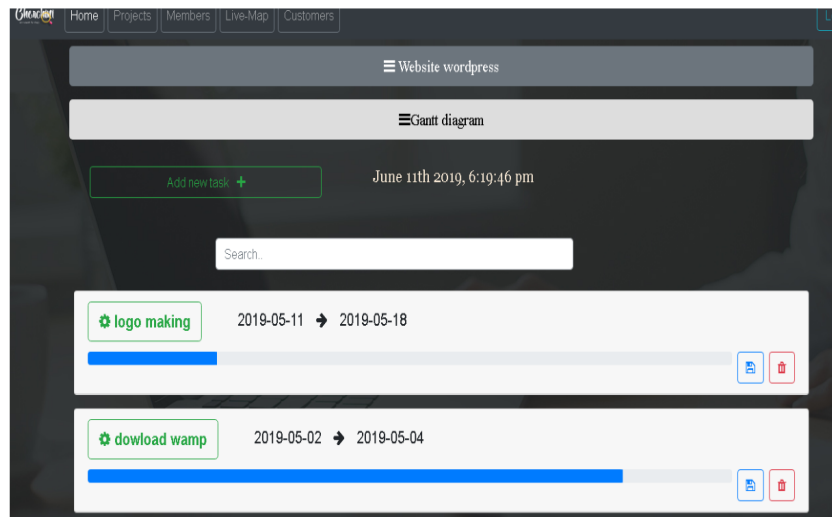


FIGURE 1.17 – Gestion de projet.4.

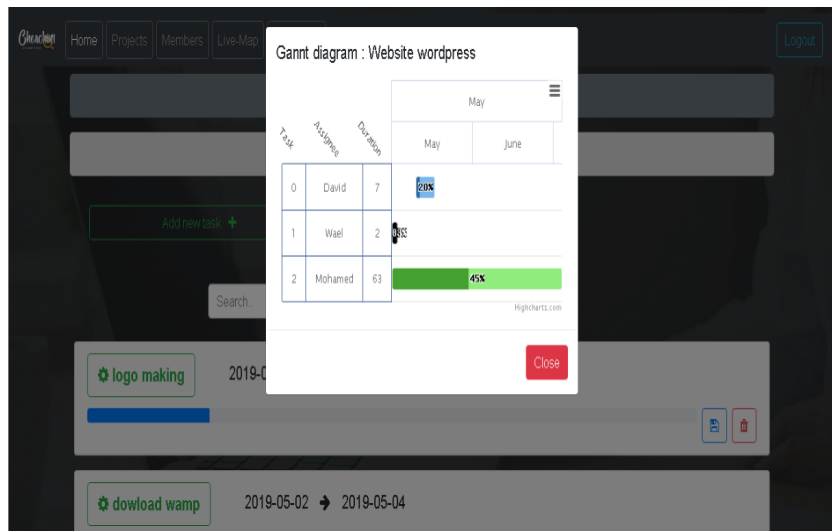


FIGURE 1.18 – Gestion de projet.5.

The screenshot displays a project management application interface. At the top, a header bar shows a green button labeled 'logo making' with a gear icon, followed by the date range '2019-05-11 → 2019-05-18'. Below this, a task card is visible with the following details:

- Name:** logo making
- Description:** add logo to webpage
- Select the new assignee:** A dropdown menu showing 'David Alaba'.
- Current assignee:** A grey button labeled 'David Alaba'.
- Status:** A dropdown menu showing 'To Do'.
- Price:** A field showing 'TND 75'.
- Start Date:** A date picker showing '11 May 2019'.
- End Date:** A date picker showing '18 May 2019'.

FIGURE 1.19 – Gestion de projet.6.

1.5 Sprint 4

1.5.1 Conception

Le scénario « Authentification » Ce schéma présente le même scénario pour l'administrateur et un simple membre :

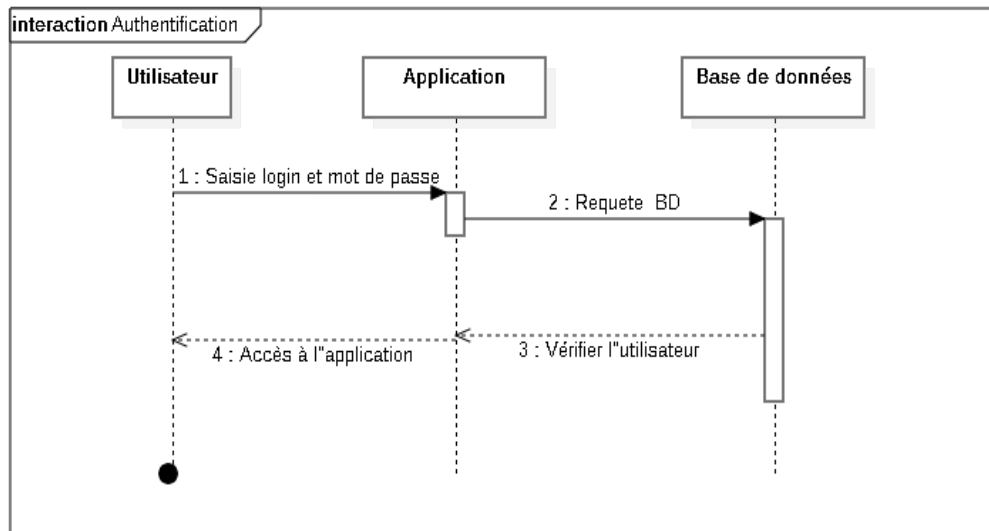


FIGURE 1.20 – Authentification.

1.5.2 Test

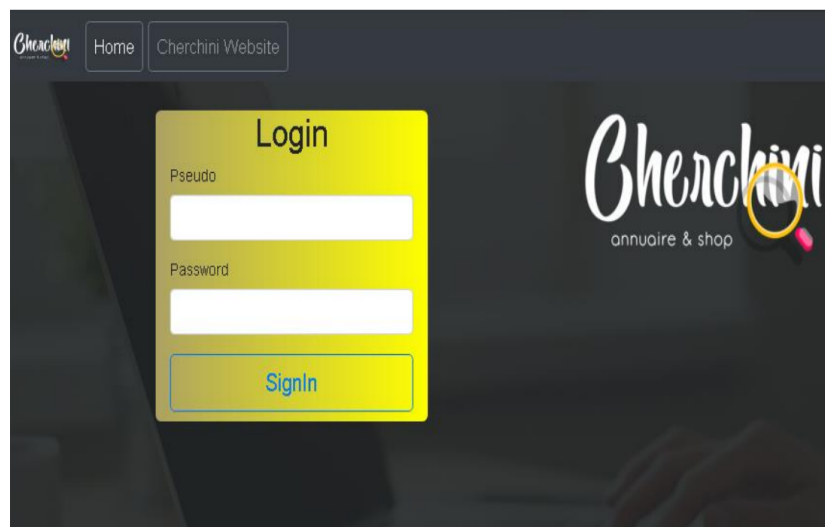


FIGURE 1.21 – Accueil.

1.6 Structure de l'application

Le schéma suivant présente la structure de notre application

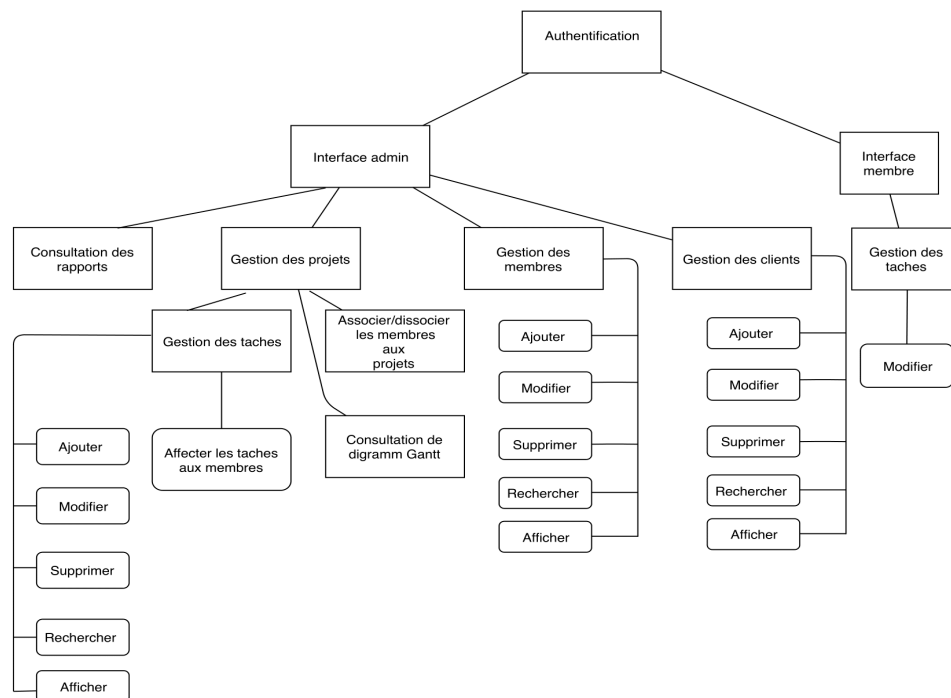


FIGURE 1.22 – Structure de l'application web.

1.7 Conception de la base de données

1.7.1 Modèle du liason des données

L'architecture de notre application nous implique à créer un modèle physique des données , et nous avons pas été besoin d'un model conceptuel logique ,puisque on a liée directement les données à l'application par le biais d'un pilote de connexion et pas par un ORM , et ceci est le diagramme Entités –Relations de la base de données qui est en interaction avec l'application web . Choix de la méthodologie de conception : La liaison par des ORM ,ou possède des avantages bienque des inconvenient , parmi ses avantages :

- **La portabilité** : ORM est utilisé pour que vous écriviez votre structure une seule fois et la couche ORM gérera l'instruction finale adaptée au SGBD configuré. C'est un excellent avantage, car une opération simple, telle que limit, est ajoutée sous la forme "limit 0,100" à la fin de l'instruction select dans MySQL, alors qu'elle est "select top 100 from table" dans MS SQL.
- **Imbrication de données** : en cas de relations, la couche ORM extraira automatiquement les données pour vous.
- **Langage unique** : vous ne connaissez pas le langage SQL pour traiter la base de données uniquement avec votre langage de développement. Ajouter revient à modifier : la plupart des couches ORM traitent l'ajout de nouvelles données (insertion SQL) et la mise à jour des données (SQL Update) de la même manière, ce qui facilite grandement l'écriture et la maintenance du code.
- **Imbrication de données** : en cas de relations, la couche ORM extraira automatiquement les données pour vous.

Et parmi les inconvenients de l'ORM on trouve :

- **La complexité des requêtes** : certaines couches ORM ont des limitations, en particulier lors de l'exécution de requêtes. Vous serez donc parfois obligé d'écrire en SQL brut.
- **Lenteur** : si vous comparez les performances entre l'écriture de SQL brut ou l'utilisation d'ORM, vous trouverez le brut beaucoup plus rapidement car il n'y a pas de couche de traduction.

- **Réglage** : si vous connaissez bien le langage SQL et votre SGBD par défaut, vous pouvez utiliser vos connaissances pour accélérer les requêtes, mais ce n'est pas la même chose avec ORM.
- **Configuration** : si vous travaillez dans un projet Big Data et que vous n'êtes pas satisfait de la performance, vous vous retrouverez en train d'étudier la couche ORM afin de pouvoir minimiser les occurrences du SGBD.

1.7.2 Diagramme de classes

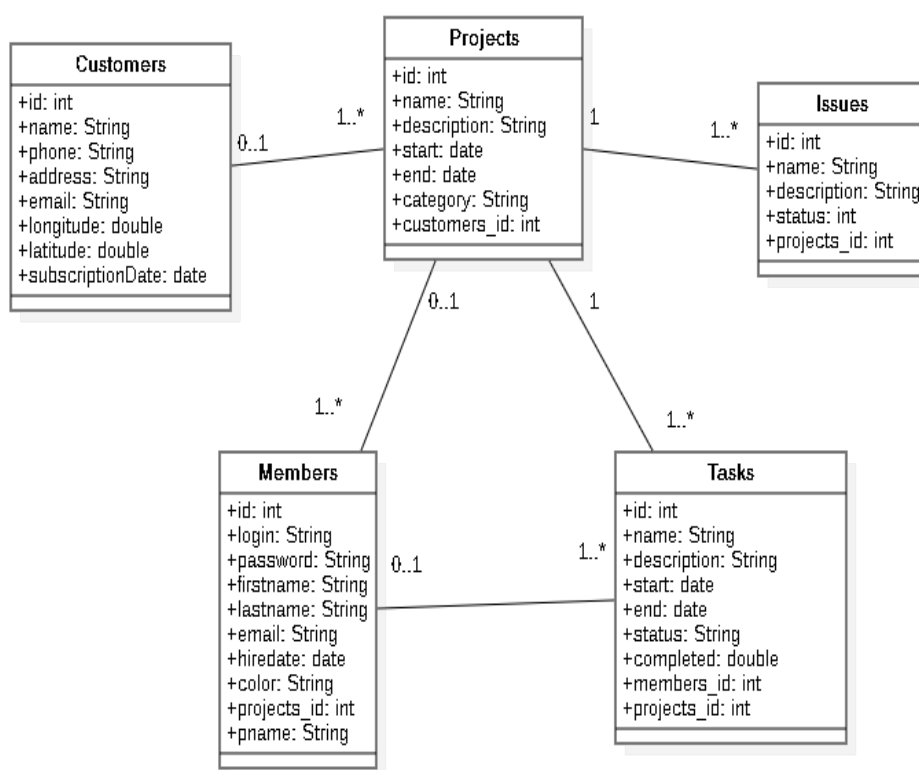


FIGURE 1.23 – Diagramme de classes.

Tableau explicatif

Nom relation	Entité E1	Entité E2	Relation(E1 :E2)	Relation(E2 :E1)
Projects -Tasks	Projects	Tasks	1 : N	1 : 1
			Non-identifiée	identifiée
Projects -Members	Projects	Members	1 : N	1 : 1
			Non-identifiée	Non-identifiée
Members-Tasks	Members	Tasks	1 : N	1 : 1
			Non-identifiée	Non-identifiée
Projects-Customers	Projects	Customers	1 : 1	1 : N
			Non-identifiée	Non-identifiée

TABLE 1.4 – Relations

1.8 Diagramme de déploiement

1.8.1 Diagramme de déploiement

- **Le Client** : C'est le navigateur web, il permet aux utilisateurs d'accéder au serveur, c'est d'interface à l'utilisateur.
- **Le serveur web** : C'est le serveur principal qui abrite les différents composants logiciels de notre application. Il assure la gestion des connexions et des requêtes du client ainsi que aussi la distribution et rendu (rendering) des pages EJS. Cet élément contient principalement un environnement d'exécution qui est le framework javascript Node js sur lequel est déployé l'application Web(Express JS).
- **La base de données** : Est exploité par le avec le serveur wamp. C'est le composant qui s'occupe du stockage et de la gestion des données. La communication des données entre l'application est la base de données est assurée par le pilote (driver) de mysql pour Node js.

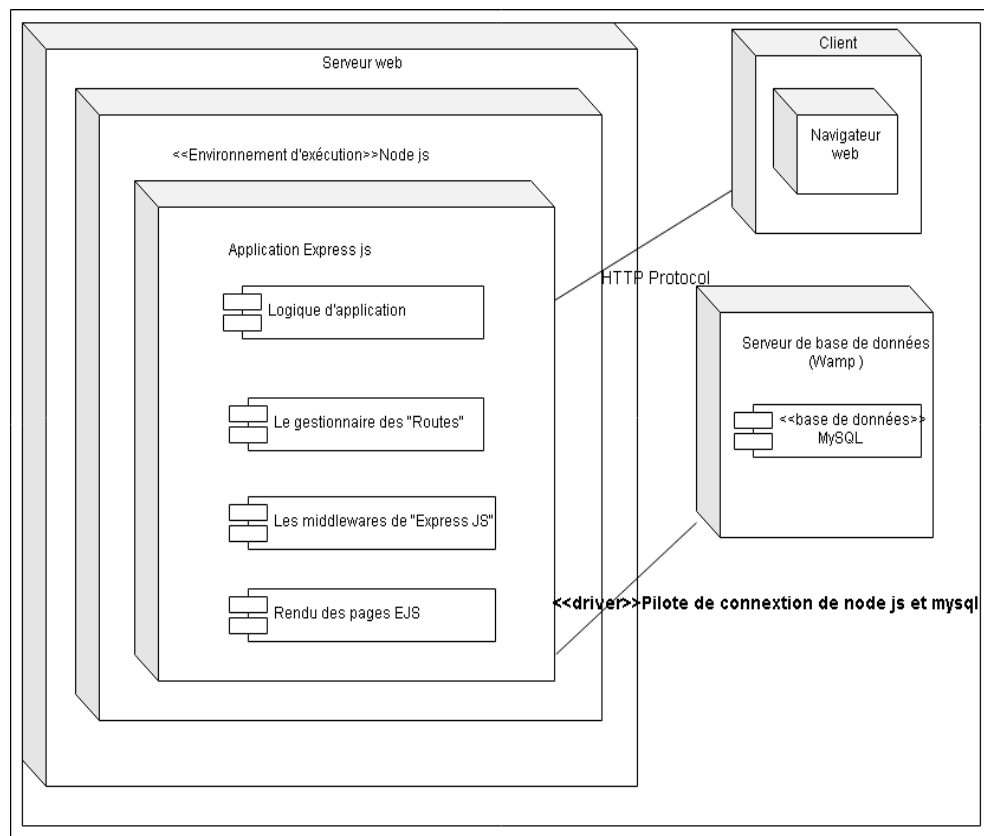
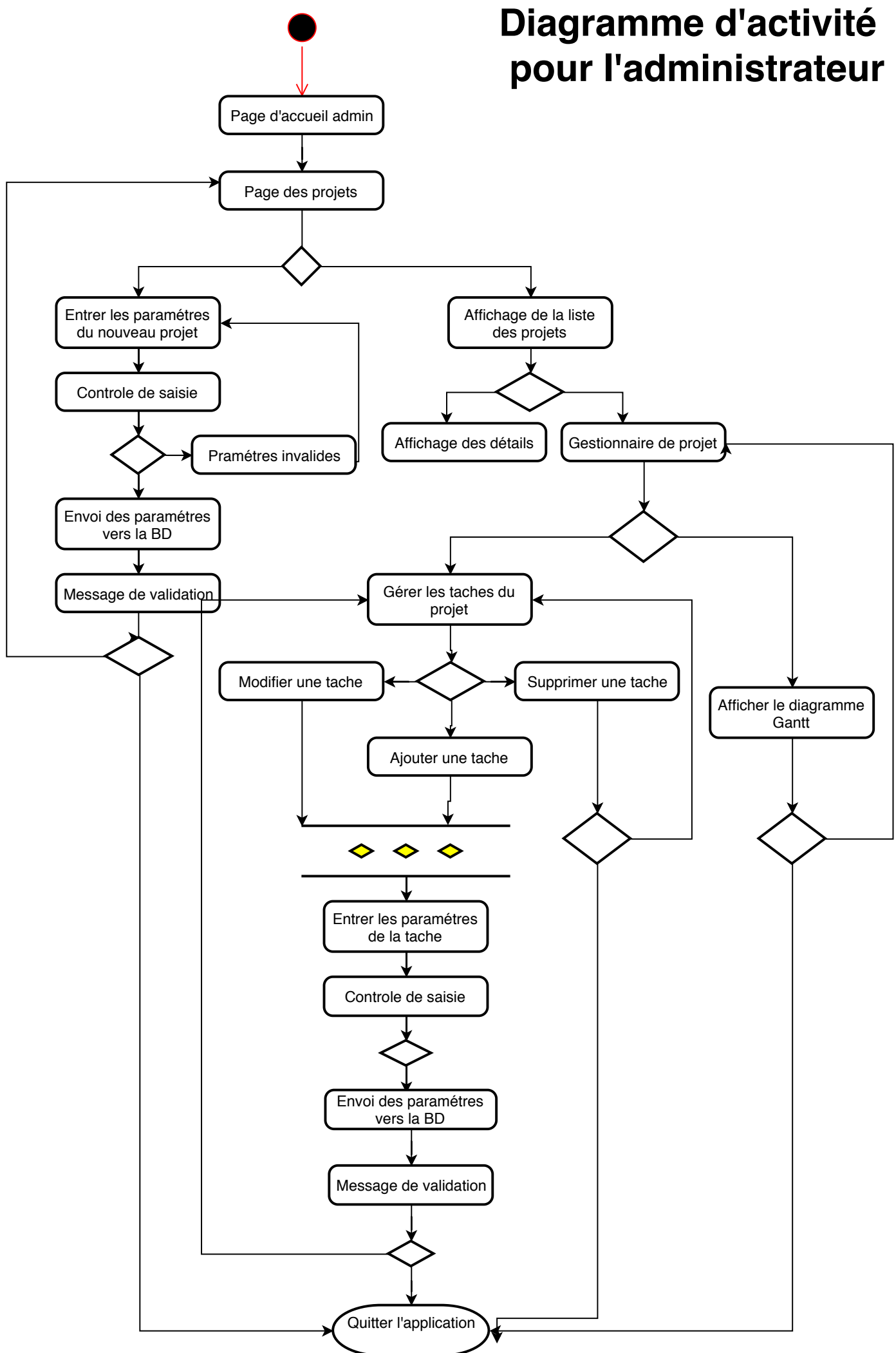


FIGURE 1.24 – Diagramme de déploiement.

1.9 Diagramme d'activité

Le diagramme d'activité nous permet de décrire les traitements. Il est particulièrement adapté à la modélisation du cheminement de flots de contrôle et de flots de données. Il permet ainsi de représenter le comportement d'une méthode ou le déroulement d'un cas d'utilisation dans un graphe. Les diagrammes d'activités sont très proches des diagrammes d'états- transitions dans leur présentation, mais leur interprétation est différente. Une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles. Le passage d'une activité vers une autre est matérialisé par une transition. Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre. Dans la page suivante , nous présentons notre diagramme d'activité qui illustre le déroulement séquentiel des traitements accomplis par l'administrateur afin de créer et gérer un projet.

Diagramme d'activité pour l'administrateur



Ainsi , la figure suivante représente le diagramme d'activité pour un membre :

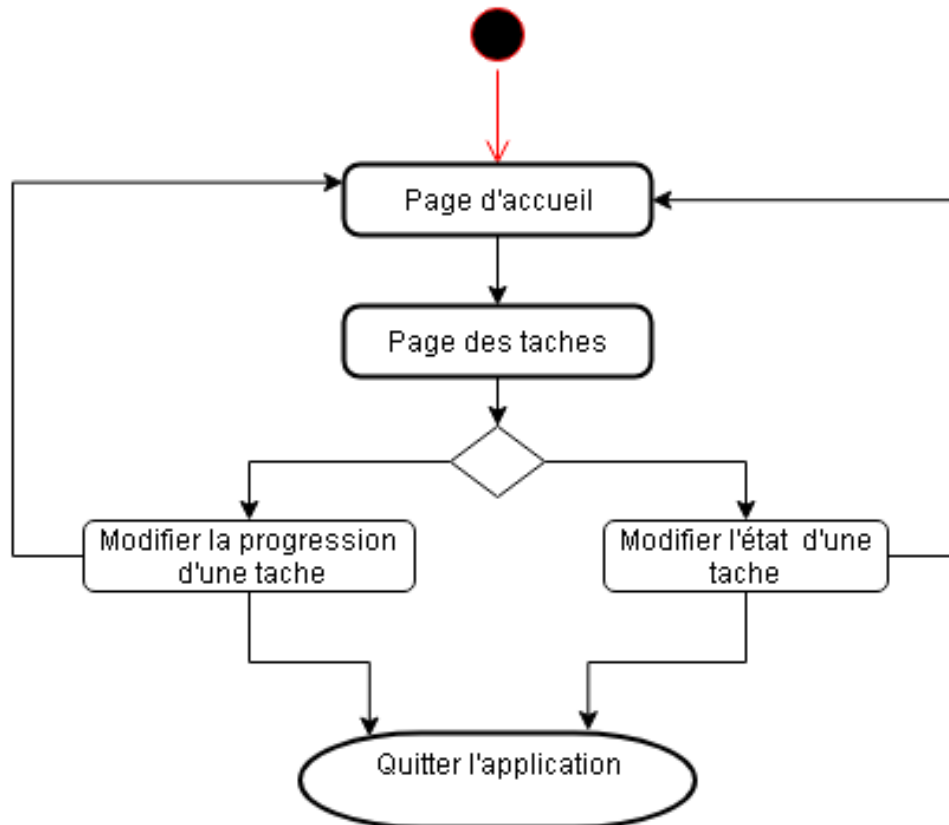


FIGURE 1.25 – Diagramme d'activité pour un membre.