Introduction Générale

Dans un monde qui évolue rapidement, nous sommes constamment confrontés à de nouveaux défis dans toutes les filières. Pour suivre cette évolution exponentielle, il est essentiel de mener autant de projets que possible grâce à la recherche. Les centres de recherche sont dédiés à cet objectif. Ils regroupent un grand nombre d'employés de différents départements, travaillant dans des bureaux différents et réalisant des projets innovants, etc. La gestion de ces ressources humaines et matérielles est essentielle pour le bon fonctionnement du centre de recherche.

Comme ce travail est très délicat, une simple erreur peut avoir des conséquences néfastes : perte de temps, d'argent et d'énergie, etc. C'est pourquoi il est important d'informatiser ce travail. Nous avons donc décidé de concevoir une base de données pour garantir l'efficacité du travail, réduire les risques d'erreur, gagner du temps pour réaliser plus de projets, éviter la duplication des données et enfin protéger et sécuriser les données. Pour faciliter cette tâche, nous allons mettre en place une application qui recueille les nouvelles données.

Dans ce cadre, nous allons réaliser notre projet de fin d'année au sein de l'École Nationale Supérieure d'Ingénieurs de Tunis (ENSIT). Notre travail consiste à concevoir, créer, alimenter et interroger une base de données et développer une application web qui facilite les tâches web .

Notre rapport est constitué de quatre chapitres :

Chapitre 1 : Etude préalable .

Chapitre 2 : Etude étude détaillée .

Chapitre 3 : .

Chapitre 4 : .

chapitre 1 : Etude préalable

Introduction

Il est difficile d'effectuer un travail fiable tout en respectant les contraintes de temps et en garantissant l'efficacité des différentes opérations de gestion d'un centre, telles que l'ajout, la suppression et la récupération de données, sans une solution informatique.

Dans ce chapitre, nous allons expliquer le contexte de notre travail. Ensuite, nous présenterons l'état de l'art des moyens disponibles en effectuant une étude critique, puis nous décrirons le travail demandé. Enfin, nous représenterons les outils utilisés dans le cadre de ce projet.

1. Contexte du projet

Un centre de recherche a pour objectif la réalisation de projets de recherche en mobilisant plusieurs ressources matérielles et humaines.

En termes de ressources humaines, un centre de recherche recrute plusieurs employés qui sont caractérisés par des informations personnelles telles que l'ID, le nom, le prénom, etc., qui doivent être enregistrées dans la base de données du centre. De plus, chaque employé peut mener une étude et occuper un seul poste téléphonique.

Il convient également de noter qu'un centre de recherche comprend plusieurs départements, et que chaque département est doté de bureaux contenant des postes téléphoniques. Ces ressources sont interdépendantes avec des contraintes spécifiques, ce qui signifie que toute gestion effectuée sur une ressource peut affecter toutes les autres.

Dans ce contexte, nous envisageons de développer une application web permettant d'effectuer les gestions nécessaires avec souplesse et fiabilité.

2. l'état de l’art

3. Description du travail demandé

En premier lieu, nous souhaitons informatiser la gestion de notre centre de recherche afin de faciliter les tâches des administrateurs et de réduire les erreurs potentielles. Nous commençons donc par concevoir, alimenter et interroger une base de données pour la gestion des employés, des études, des projets de recherche, des postes téléphoniques, des bureaux et des départements. Le terme "gestion" englobe les actions suivantes : l'ajout, la suppression, la modification et la recherche des données.

En deuxième lieu, nous souhaitons offrir aux administrateurs la possibilité de gérer ces données à travers une interface graphique accessible via une application web.

Pour ce faire, nous avons choisi d'utiliser plusieurs outils .

4. les outils utilisés

pour l’ Environnement logiciel :

* git hub :

GitHub est un environnement logiciel très avantageux en raison de sa documentation et de sa capacité à présenter un travail. Pour notre part, nous avons choisi GitHub principalement pour l'un de ses avantages : faciliter le travail en groupe. En effet, lorsque plusieurs personnes collaborent sur un projet, il est difficile de suivre les révisions, c'est-à-dire de savoir qui a modifié quoi, quand et où ces fichiers sont stockés. GitHub résout ce problème en gardant une trace de toutes les modifications qui ont été poussées vers le référentiel. Tout comme l'utilisation de Microsoft Word ou de Google Drive, vous pouvez avoir un historique des versions de votre code afin que les versions précédentes ne soient pas perdues à chaque itération .

* power AMC
* git bash :

Le premier avantage, qui a été la première raison de notre migration, est la gestion des branches. Nous pouvons désormais travailler sur plusieurs projets en parallèle sans nous marcher sur les pieds.

L'interface console et les algorithmes de fusion (merge) sont également très utiles. Lorsqu'un fichier a été modifié par plusieurs personnes en même temps, Git est capable de choisir un algorithme qui fusionne intelligemment les lignes du fichier qui ont été modifiées.

La rapidité est également un point fort de Git. Lorsque nous mettons à jour le code, les données sont empaquetées, compressées et les mises à jour sont fusionnées à la vitesse de la lumière.

* webstorm :

WebStorm est un environnement de développement intégré pour JavaScript et les technologies connexes. Comme les autres IDE de JetBrains, il rend votre expérience de développement plus conviviale, en automatisant les tâches répétitives et en vous aidant à gérer les tâches complexes avec facilité.

* intellij idea :

IntelliJ IDEA est un environnement de développement intégré (IDE) intelligent et prenant en compte le contexte qui permet de travailler sur une grande variété d'applications en Java et dans d'autres langages.

Il présente plusieurs avantages, dont une ergonomie en faveur du développeur. Chaque aspect d'IntelliJ IDEA est conçu pour offrir une expérience fluide dès la première utilisation. L'IDE fournit un accès rapide à toutes les fonctionnalités et à tous les outils intégrés qui peuvent être utiles dans le travail des développeurs, ainsi qu'à un large éventail d'options de personnalisation.

pour la conception :

* nous allons utiliser l’UML :

c’est un langage formel et normalisé clair et précis. encourage l’utilisation d'outils.De plus, c'est un support de communication performant. L'analyse devient plus facile. facilite la compréhension de représentations abstraites complexes.

pour l'implémentation du base de donné :

* On a utilisé mysql : MySQL est un système interactif dont le but est d'agir comme un gestionnaire de base de données, en utilisant le langage SQL pour fonctionner. Grâce à lui, les données peuvent être gérées, y compris leur introduction, leur accès et leur traitement.

Parmi les principales qualités de MySQL, on distingue les suivantes :

* Permet de créer tout type d'application.
* Possède des privilèges de haute sécurité.
* Capable de gérer de gros volumes de données.
* Il permet la réalisation de requêtes, auxquelles on répond rapidement.
* Il a une capacité de support technique élevée.
* Pour son fonctionnement, une grande quantité de ressources n'est pas nécessaire, ce qui se traduit par un faible coût.
* Sa structure fait intervenir des couches et des modules, ce qui lui confère une grande stabilité.
* Le processus d'importation et d'exportation de données est assez simple

pour le développement web :

* pour le front-end : nous allons utiliser angular framework JavaScript :

Avec une concurrence aussi extrême, l'exigence d'interactivité et **applications web conviviales** devient très important pour promouvoir le succès de l'entreprise. **Angular** sert de cadre idéal pour diriger l'industrie pour la production de solutions évolutives qui répondent aux besoins des gens.

Le principal avantage de **AngularJS** est qu'il permet aux développeurs de traduire des documents HTML statiques en contenu dynamique. AngularJS est devenu un choix de choix pour les applications Web, ayant une architecture **MVC** ce quiest le cas de notre projet , et des outils robustes, en particulier pour les applications d'une seule page.

* pour le back-end : nous allons utiliser spring boot framework Java :

Les **applications Spring Boot** sont une avancée majeure dans le développement Java et l’utilisation du framework [Spring](https://spring.io/).

Un des avantages que nous avons perçu dans l’utilisation de ces applications est la simplicité d’administration et de déploiement, et qui permet donc une mise en œuvre rapide d’un processus de déploiement avec le couple Ansible / Git.

Spring Boot lui-même est un cadre bien sécurisé et considéré à l'abri de la plupart des cyber-attaques car il fait partie de Java Spring. Le cadre est sécurisé contre la falsification de demandes inter sites, où un pirate peut manipuler tout le fonctionnement de l'application pour discuter avec les utilisateurs.

5. Conclusion

Ce chapitre a pour but de clarifier les principaux axes de notre projet et de présenter les outils que nous utiliserons tout au long de celui-ci. Dans le chapitre suivant, nous effectuerons une étude détaillée en analysant minutieusement le cahier des charges. Nous présenterons également le diagramme de cas d'utilisation, les scénarios possibles, les diagrammes de séquence d'analyse, etc.

chapitre 2 : Etude détaillée

1. Introduction :

Ce chapitre va permettre de mieux expliquer le projet demandé. En effet, dans ce dernier, nous allons effectuer une analyse détaillée du cahier des charges. Ensuite, nous présenterons le diagramme de cas d'utilisation, les scénarios des cas d'utilisation, les diagrammes de séquence d'analyse et le diagramme de classe d'analyse. Nous terminerons en effectuant une implémentation de la base de données.

2. Analyse détaillé du cahier de charge :

Les acteurs :

Un acteur est une entité externe au système qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé. Dans notre cas, nous avons identifié trois acteurs :

* Un administrateur : il est responsable de la gestion de certaines données, ses pouvoirs sont restreints.
* Un directeur : il est le gestionnaire principal. Il a le droit de gérer toutes les données.
* Un internaute : c’est un simple visiteur du site. Dans le but de s'informer, il est garanti l'accès à certains projets de recherche effectués par le centre de recherche.

Les cas d'utilisation :

Les fonctionnalités garanties aux administrateurs et directeurs :

* Authentification : chaque administrateur ou directeur doit s’authentifier pour exercer les différentes tâches.
* Gestion des employés.
* Gestion des études.
* Gestion des projets de recherche.
* Gestion des postes téléphoniques.
* Gestion des bureaux.
* Le directeur a une fonctionnalité de plus :
* Gestion des départements.
* Les fonctionnalités garanties aux internautes :
* Authentification : ils doivent préciser leur identité.
* Consultation des projets de recherche valides.

3. Diagramme de use case :

En langage de modélisation unifié (UML), un diagramme de cas d'utilisation peut servir à résumer les informations des utilisateurs de votre système (également appelés acteurs) et leurs interactions avec ce dernier.

À travers ce diagramme, on peut représenter plus clairement les cas d'utilisation et les liens entre ces cas. Ces liens sont définis par les cardinalités entre les différentes entités (cas) et traduisent les modifications implicites impliquées par une gestion désirée des données. Ceci est représenté dans le diagramme par les deux mots-clés "extend" et "include".

Pour mieux expliquer, prenons le cas élaboré dans la figure ci-dessous où l'administrateur veut ajouter un employé : le cas d'utilisation "gérer les employés" est lié à l'authentification par le mot-clé "include". Ainsi, l'admin doit passer obligatoirement par l'authentification. Il est également lié aux cas d'utilisation "gestion d'étude" et "gestion des projets de recherche" par le mot-clé "extend". Ainsi, l'admin peut passer pour faire ses gestions ou non.

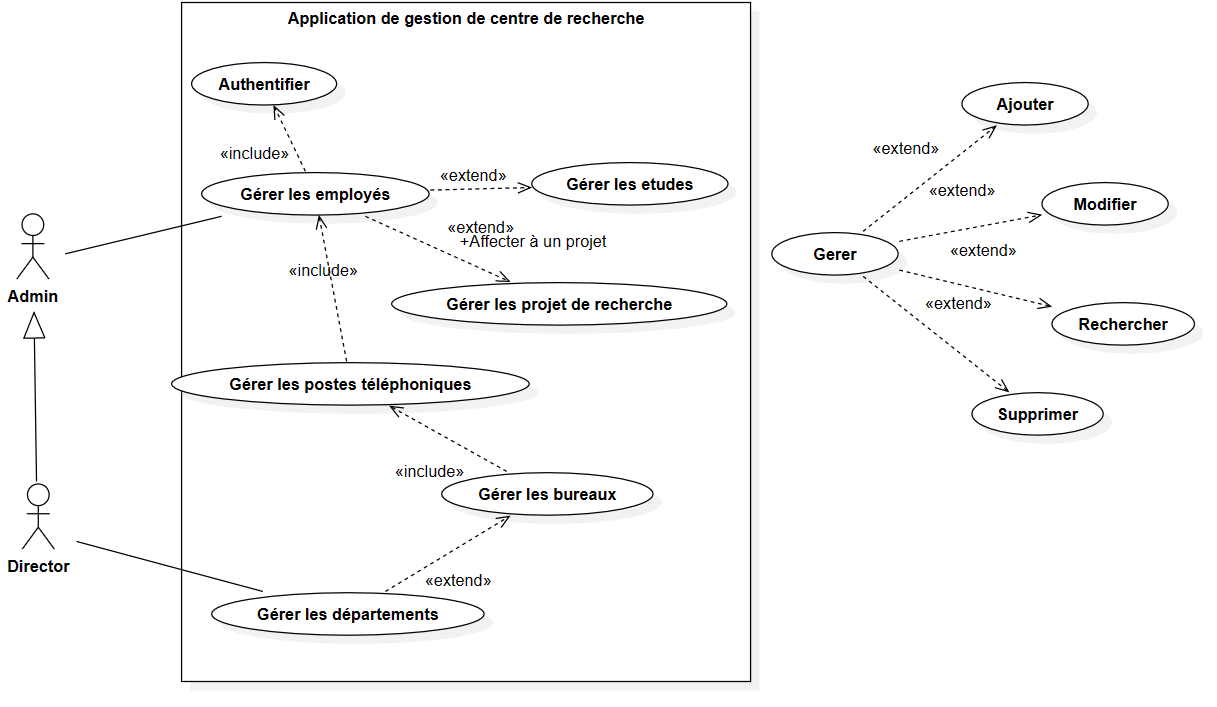


figure 1 diagramme use case

4.scénario des cas d'utilisation :

un scénario de cas d’utilisation présente le déroulement logique d'une procédure : les cas d’utilisation présentés précédemment dans le diagramme de cas d’utilisation .

on a représenté le scénario par un tableau contenant :

**le titre**

**l’acteur** : celui qui a le droit d’effectuer la gestion désirée .

représenter un cas d’utilisation selon 3 types de scénario

**Scénario nominal:**C’est le scénario idéal »(tout se passe bien) pour le cas d’utilisation.On décrit un enchaînement d’actions (Acteur Système) qui conduisent au bon déroulement du cas d’utilisation. On parle également de scénario de base, scénario normal.

**Scénario alternatif**:Partant du scénario nominal, on étudie chaque point de l’enchaînement et s’il existe une variante, on propose «une nouvelle façon de dérouler le cas d’utilisation ».

**Scénario d’erreur :**

**Préconditions**:Les conditions nécessaires pour déclencher le cas d’utilisation

**Post conditions:**Les conditions de sortie du cas d’utilisation (état du système après réalisation du cas d’utilisation)

En prenant l'exemple du scénario de gestion des postes téléphoniques - figure 2 -, le titre dans la première colonne met en évidence la tâche souhaitée : la suppression d'un poste téléphonique. L'acteur désigné est l'administrateur, qui a le droit d'effectuer cette tâche, mais ce droit est soumis à des conditions spécifiques, appelées les préconditions. Ainsi, l'administrateur doit être authentifié et le poste téléphonique qu'il souhaite supprimer doit déjà exister dans la base de données.

Dans le cas nominal, l'administrateur cherche le poste téléphonique qu'il souhaite supprimer, puis il appuie sur le bouton de suppression et confirme la suppression à travers une boîte de dialogue pop-up.

Dans le cas alternatif, on traite un autre scénario où la suppression n’est pas effectuée. Dans ce cas, une boîte de dialogue pop-up apparaît pour l'informer qu’il y a un problème .

Enfin, on trouve la postcondition qui implique l'affichage de la page d'accueil suite aux événements effectués par l'administrateur.

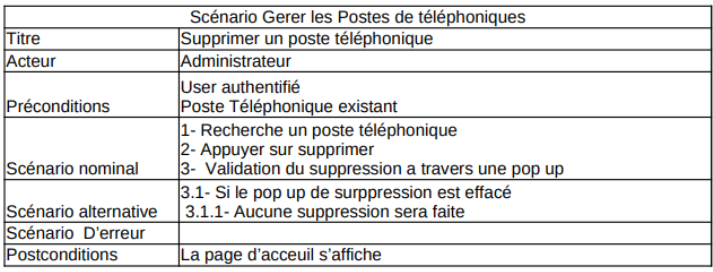


figure -2-

scénario de gestion des poste téléphonique cas du suppression

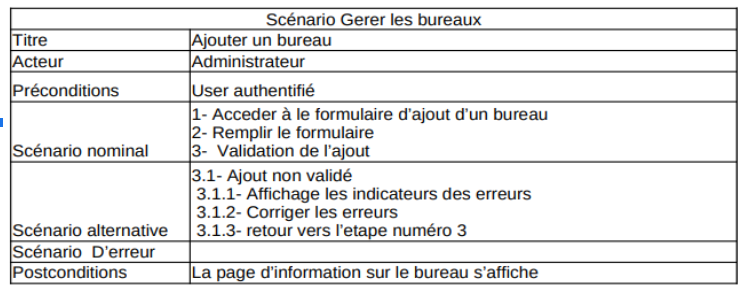


figure -3-

scénario de gestion des bureaux cas d’ajout

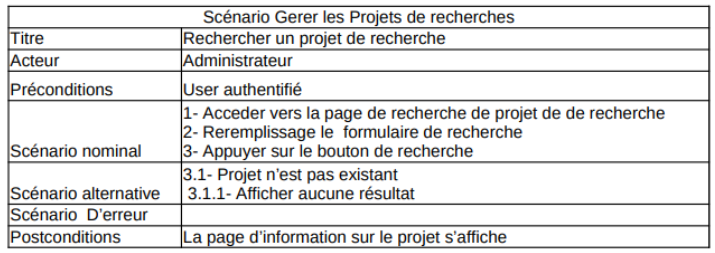


figure -4-

scénario de gestion des projets de recherche cas de recherche

5.diagramme de séquence analyse :

Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Ces diagrammes sont utilisés à la fois par les développeurs logiciels et les managers d'entreprises pour analyser les besoins d'un nouveau système ou documenter un processus existant. Les diagrammes de séquence sont parfois appelés diagrammes d'événements ou scénarios d'événements.

Prenons l'exemple de la figure 5. Ce diagramme de séquence système met en évidence un administrateur qui interagit avec l'application de gestion de recherche de projets .

Dans ce scénario, l'administrateur demande l'accès à la page de recherche de projets . Le système garantit alors l'accès et affiche un formulaire de recherche. Une fois authentifié, l'administrateur remplit le formulaire avec les données nécessaires , puis appuie sur le bouton de recherche.

Ensuite, deux cas peuvent se produire. Si le système trouve des résultats correspondant à la recherche, il les affiche à l'administrateur. Sinon, le système affiche un message informant qu'aucun résultat n'a été trouvé.

Le diagramme de séquence système permet de visualiser les interactions entre l'administrateur et le système, ainsi que les étapes chronologiques de l'utilisation de l'application de gestion de recherche de projets cas de recherche.

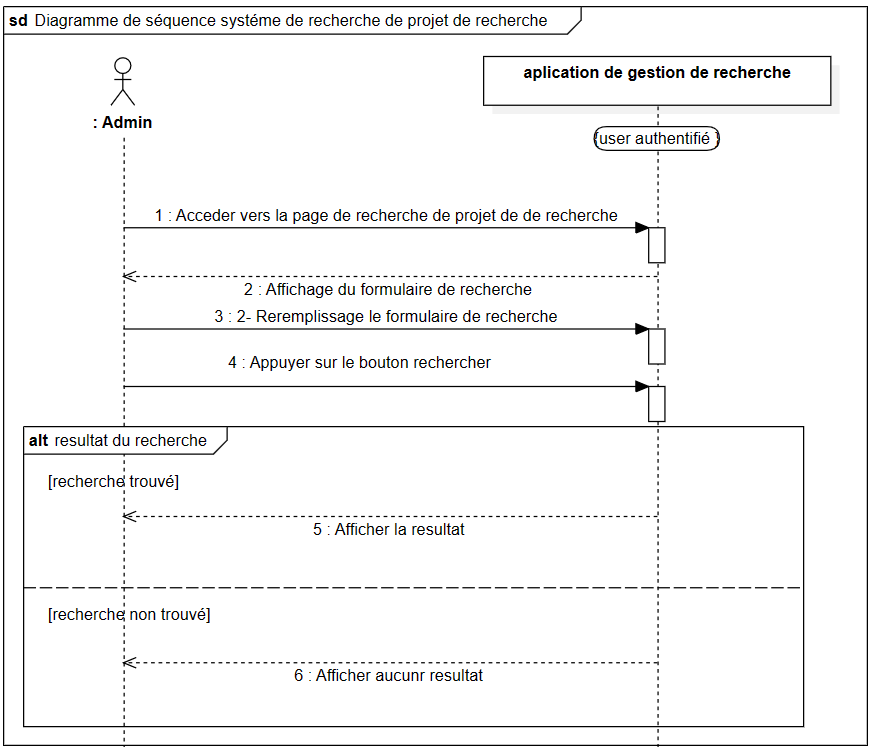


figure -5-

diagramme de séquence système de recherche d’un projet de recherche

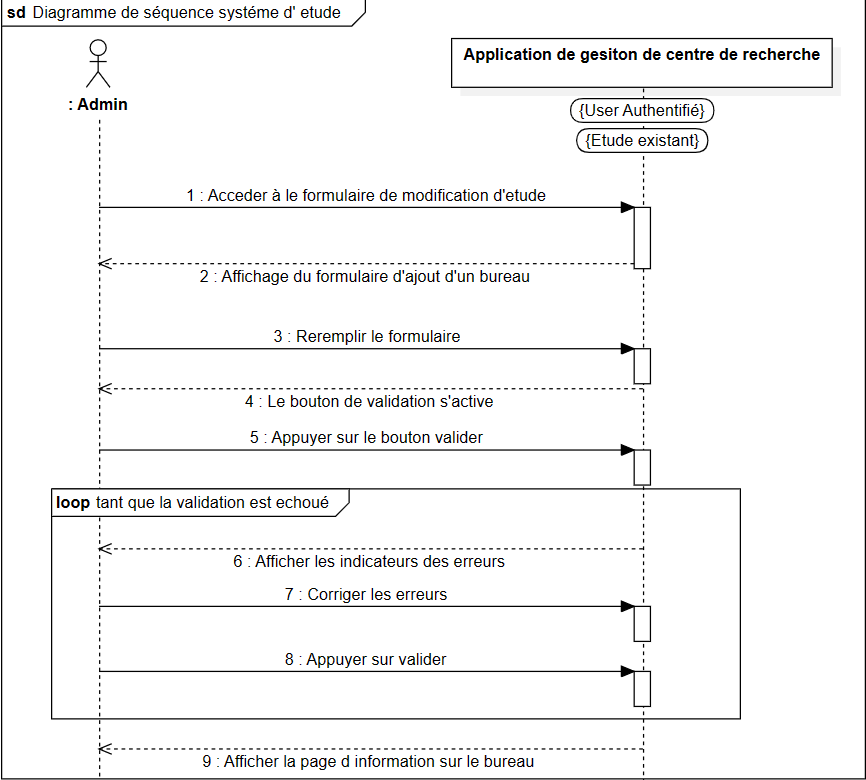


figure -6-

diagramme de séquence système d'étude

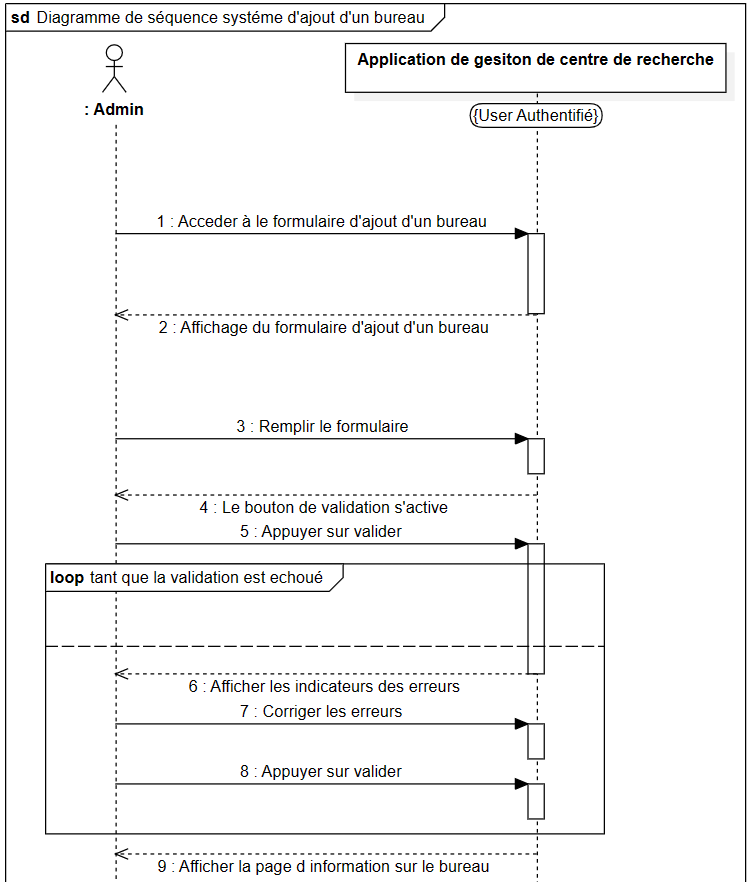


figure -6-

diagramme de séquence système d’ajout d’un bureau

6.diagramme de classe analyse et extraction du modèle :

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d’un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.

Le diagramme de classe Figure-7- présente 6 classes : prenons l'exemple de l'entité "Département". Cette entité a 3 attributs : un "NumDep" comme clé primaire, un "IntituléDep" et un "BudgetDep".

Un département mène un et un seul projet de recherche. Ceci est traduit par l'association "mener" et la cardinalité 1,1.

Un département est dirigé par un et un seul employé. Ceci est traduit par l'association "diriger" et la cardinalité 1,1.

À un département, un et un seul bureau est affecté. Ceci est traduit par l'association "affecter" et la cardinalité 1,1.

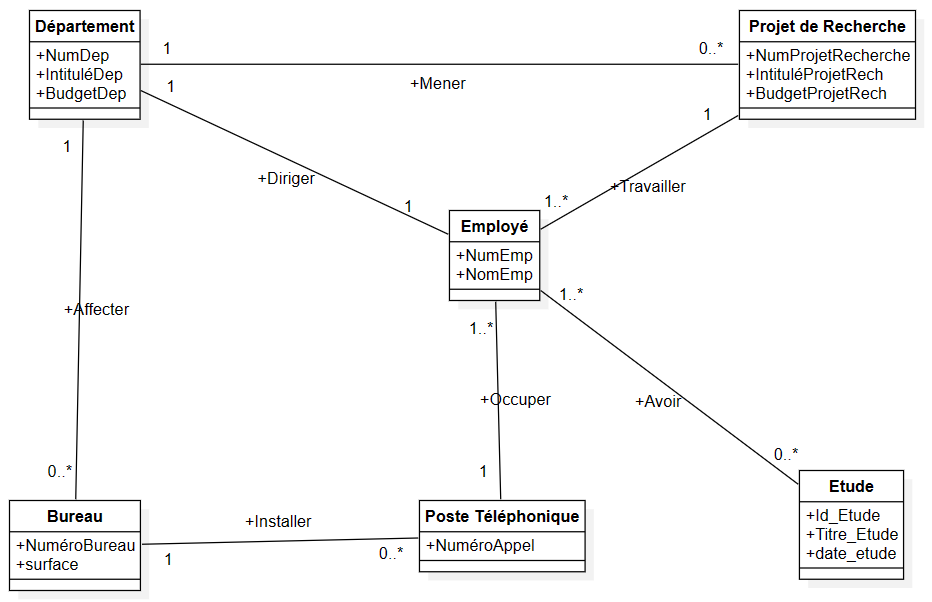


figure -7-

diagramme de diagramme de classe analyse

À travers ce diagramme, on a pu générer les schémas relationnels qui vont être présentés sous forme de tables et implémentés ultérieurement.

\*\*\*\*\*

7.implémentation du base de données :

Comme le montre l’image ci-dessous on a pu distinguer 5 tables qu’on a implémenté dans la base de données en utilisant mysql comme outil logiciel .

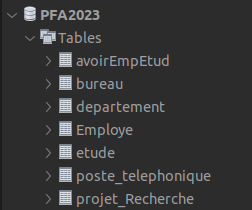


figure -8-

implémentation du base de donnée

\*\*\*\*\*

9. conclusion :

Après la conception analytique , nous allons nous intéresser dans le dernier chapitre à présenter le diagramme de package , diagramme de classe conception , les maquettes et le diagramme de séquence objet .