**Introduction :**

Nous avons préparé un projet qui consiste à concevoir et développer une application Android mobile pour transférer des fichiers.

Dans un premier temps, on va analyser les besoins concernant cette application, en indiquant : les besoins fonctionnels et non fonctionnels, une étude de l’existant et des solutions pour la réalisation. On exploite cette partie pour présenter l’entreprise, le cadre du stage et décrire de façon détaillée le sujet.

Ensuite, on va décrire la phase de conception, en présentant des diagrammes du langage UML comme : les diagrammes de cas d’utilisation générale et détaillé en précisant les acteurs impliqués dans cette application, le diagramme de classes, les diagrammes d’activités.

Enfin, on va décrire la phase de la réalisation et du codage de l’application en indiquant les outils informatiques utilisés tout au long du développement. On va présenter les interfaces inclus dans l’application et on va indiquer des techniques pour réaliser cette application mobile.

Notre travail consiste à concevoir et implémenter une application Android pour Transfer des fichiers, il s’agit d’une application mobile de transfert des fichiers à base de serveur web http qui sera réalisé avec trois technologies :

- la 1ère technologie consiste à développer l’application sur Android studio avec le langage Java.

-La 2ème technologie consiste à développer l’application avec la Framework Flutter, le langage Dart et IDE Android studio

-la 3ème technologie consiste à développer l’application avec la technologie ReactNative, le langage JavaScript.

Ce rapport se limte a la réalisation du projet avec la 1er technologie .

Après la réalisation du projet avec les trois technologies une étude comparative sera élaborer permettant d’énumerer les avantages et les inconvénients de chacune de technologie.

La réalisation avec chacune des technologie est programmé pendant 6 mois de travail

Cycle de vie de développement d’une application mobile :

Le « cycle de vie d'un logiciel » (en anglais software life-cycle), ou les étapes méthodes de développement, sélectionne et identifie toutes les étapes de développement d'un logiciel, dès sa conception en allant jusqu'à disparition. L'objectif était de définir des balises intermédiaires permettant la validation de la partie développement logiciel, c'est-à-dire si le logiciel répond aux besoins exprimés, et la vérification du processus de développement, et si les méthodes mises en œuvre respectant bien les contraintes prédéfinies auparavant.

**-Cycle en spirale :**

Dans notre application mobile nous travaille sur 4 Activités globales :

1. Connexion
2. Configuration
3. Administration
4. Files Transfer

Pour chaque Activité on passe par les 5 phases suivantes :

1. Conception Fonctionnelle :
2. Design des interfaces d’utilisateur :
3. Conception de l’architecture
4. Codage
5. Test

* Introduction des phases :

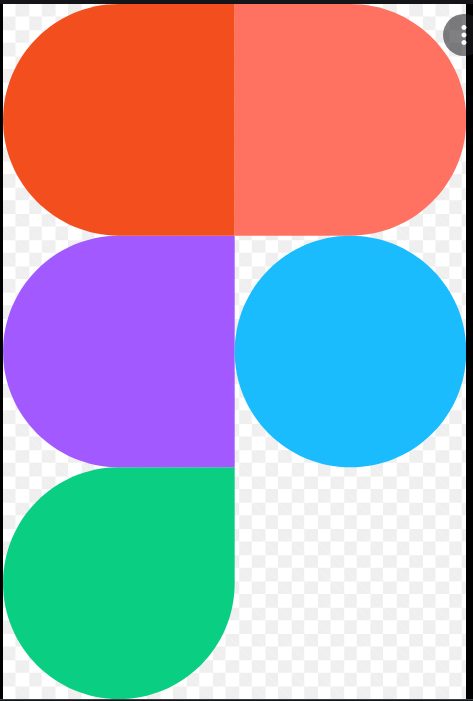
1.Conception Fonctionnelle :

Nous avons utilisé pour la partie conception fonctionnelle le langage UML

(Unified Modeling Language) est un langage de modélisation unifié permet de modéliser une application logicielle d'une façon standard dans le cadre de conception orienté objet.

-On a utlisé en particulier les deux diagrammes cas d’utlisation et diagramme d’activité

2.Design de l’interface utilisateur :

Nous avons utilisé pour la partie design « Figma.com» :

Figma est un éditeur de graphiques vectoriels et un outil de prototypage. Il est principalement basé sur le web, avec des fonctionnalités hors ligne supplémentaires activées par des applications de bureau pour macOS et Windows.

3.Conception de l’architecture

La méthodologie merise - +MCD de la BD

-MCD **:** Le **MCD** (**Modèle Conceptuel des Données**)est une représentation graphique de haut niveau qui permet facilement et simplement de comprendre comment les différents éléments sont liés entre eux à l’aide de diagrammes codifiés .

-MPD :Le MPD (**Modèle Physique des Données) :**L’étape de création du MPD est presque une formalité comparée à la création du MCD. En s’appuyant sur des règles simples (et qui fonctionnent à tous les coups), l’analyste fait évoluer sa modélisation de haut niveau pour la transformer en un schéma plus proche des contraintes des logiciels de bases de données. Il s’agit de préparer l’implémentation dans un [SGBDR](https://www.base-de-donnees.com/sgbd/).

-en particulier on a utilisé les modèles fonctionnelles et les modèles physiques :

Nous avons utilisé le logiciel Power AMC pour modéliser le diagramme

ER (entité-relation) servira à modéliser le MCD (modelé conceptuel des données) de la base de données, c.-à-d. la structure de la base de données qui va contenir les informations sur les utilisateurs et leurs connexions.

MERISE  :

MERISE (Méthode d’Etude et de Réalisation Informatique pour les Systèmes d’Entreprise) est une méthode d'analyse et de réalisation des systèmes d'information qui est élaborée en plusieurs étapes : schéma directeur, étude préalable, étude détaillée et la réalisation.

1. Codage

Nous allons présenter l'environnement matériel et logiciel utilisé pour le développement de la solution proposé tout en expliquant éventuellement nos choix techniques relatif aux langages de programmation et des outils utilisés. Enfin, nous allons donner une présentation des interfaces globales ainsi qu'une description du fonctionnement du système.

* Technologies utilisées dans l'application:

Android SDK (Software Development Kit) :

Android est un système d'exploitation mobile pour Smartphones, tablettes tactiles, PDA, smartwatches (version Wear) et terminaux mobiles.

**AndroidStudio IDE (Integrated Development Editor)**  : Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.

**Git VCS (version Control System): Git** est de loin le système de contrôle de version le plus largement utilisé aujourd'hui. **Git** est un projet open source avancé, qui est activement maintenu.

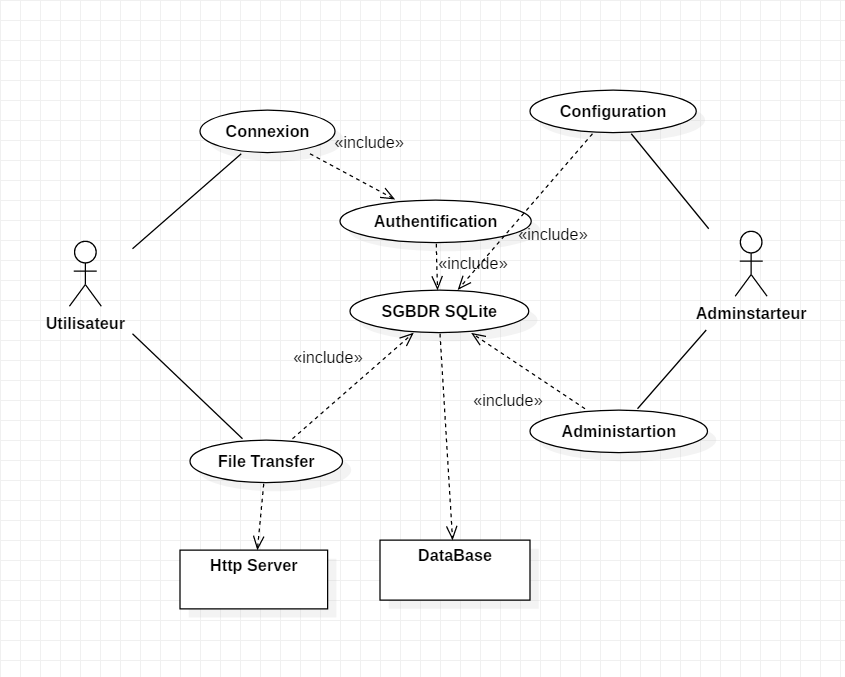


**SourceTree GUI (Graphical User Interface) : Une interface graphique Git qui offre une représentation visuelle de vos référentiels. Sourcetree est un client Git gratuit pour Windows et Mac.**

7.Test :

-Etude de besoin de l’application :

1/-Diagramme de cas d’utlisation globale :



D’après ce diagramme il y a 4 activités a prendre en considération par notre application mobile :

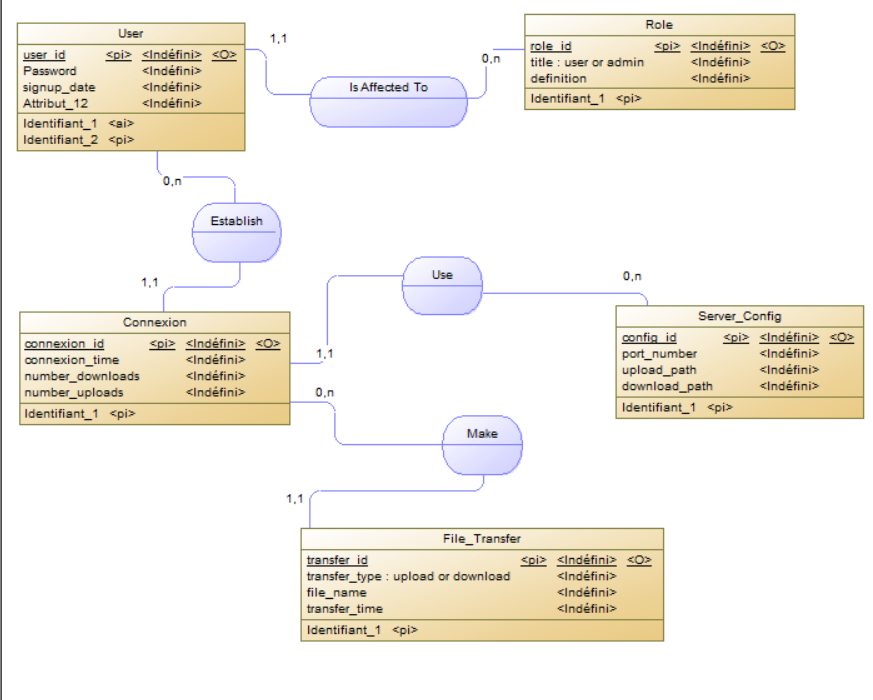
-Connecion

-Adminstration

-Configuration

-File Transfer

2/Diagramme MCD globale :



3/-Architecture d’une application Android  :

-Activity def : +cycle de vie d’une activity dans app android

-Fragment : def : +cycle de vie fragment d’une app

Le modèle de projet choisi génère le code pour une application avec une activité. Une activité est un composant d’une application Android qui représente un écran avec lequel l’utilisateur va pouvoir interagir. Un composant Android est représenté par une classe Java. Cependant, un composant n’est pas instancié directement dans le code de l’application. C’est lors de l’exécution que le système prend en charge la création et le cycle de vie d’un objet de cette classe. Dans le cas d’une activité, un objet correspondant est instancié par le système quand il est nécessaire d’afficher l’écran à l’utilisateur au lancement de l’application.

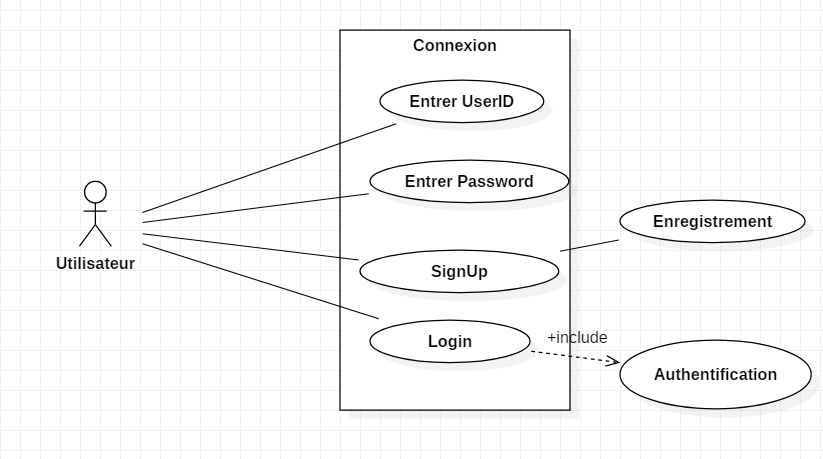
Pour notre application, le générateur de projet a créé une classe <MainActivity> dans le répertoire Java.

**Activité Connexion :**

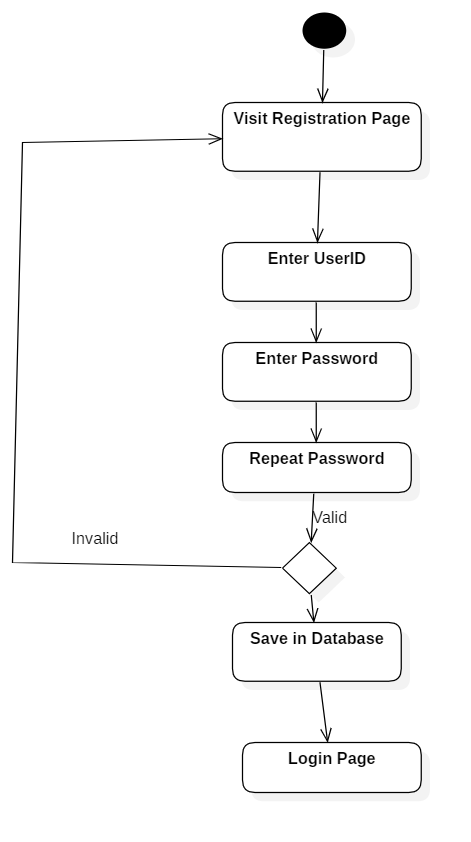
Dans cette partie nous avons travaillait sur l’Activité Connexion de l’application Android de notre projet.

1. Conception Fonctionnelle UML:

-Diagramme cas d’utilisation pour l’activité Connexion :



-Diagramme d’activité pour chaque cas d’utilisation :

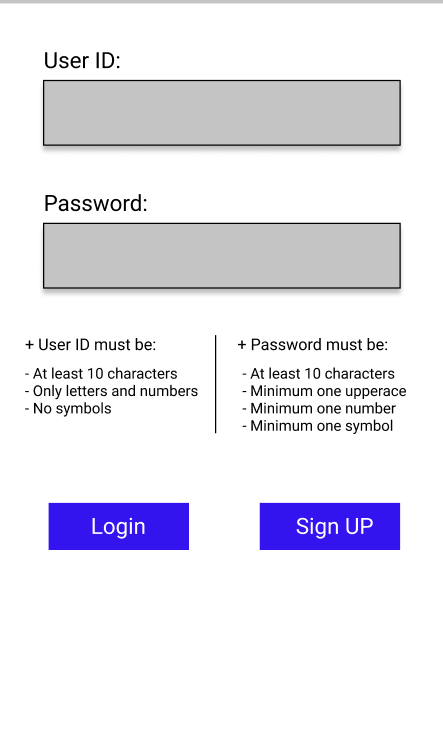
\*UML pour la page de « SignUp » :

Dans cette partie l’utilisateur créer un compte.

\*UML pour la page de Connexion (Login) :

Dans cette partie l’utilisateur saisie (userid)

Et (password) pour s’identifier a l’application.

1. Design :

Nous avons utilisé pour la partie design « Figma.com» :

Figma est un éditeur de graphiques vectoriels et un outil de prototypage. Il est principalement basé sur le web, avec des fonctionnalités hors ligne supplémentaires activées par des applications de bureau pour macOS et Windows.

Pour la création de mot de passe sécurisé dans l’application il doit comporter :

- au moins 10 caractères

-au moins une majuscule

-au moins un chiffre

-au moins un symbole.

Pour la création d’un nom utilisateur il faut respecter les règles suivantes :

- au moins 10 caractères

-seulement des lettres et des chiffres

-sans symbole.

1. Conception d’architecture:

-MCD (table user and table connexion)

-MPD

* Modélisation de Base de données avec la méthodologie merise :
* Diagramme entité relation :

Table User : Enregistrer les utilisateurs.

Cette table contient les champs : User-id, Password et Signup\_date.

Table Connexion : Enregistrer les connexions d’un utilisateur dans la base donnée, chaque utilisateur faire la connexion est enregistré dans la table de connexion. Cette table contient les champs : Connexion\_id, Connexion\_time, number\_downloads et number\_uploads.



## Diagramme MPD (Multi Protocol Driver) :

## le modèle physique des données consiste à implanter une base de données dans un SGBDR. Le langage utilisé pour ce type d'opération est le SQL.

Table User : Enregistrer les utilisateurs.

Cette table contient les champs : User-id, role\_id, password et Signup\_date.

Table Connexion : Enregistrer les connexions d’un utilisateur dans la base donnée, chaque utilisateur faire la connexion est enregistré dans la table de connexion. Cette table contient les champs : Connexion\_id, User, config\_id, Connexion\_time, number\_downloads et number\_uploads.

## C:\Users\salim\Pictures\Screenshots\Screenshot (1554).png

1. **Codage** :

-les classes DatabaseHelper , Connexion.java, mainActivity

-interface : Connexionfragment , viewmodel.

**Package MyApplication**

-MainActivity

**Package DB**

-DatabaseHelper

-User

-Connexion

**Package UI**

-ConnexionFragment

-ConnexionViewModel

-**MainActivity** : est la classe de démarage de l’application .

-**Le Pacakage db**: contient les classes suivantes :

-DatabaseHelper : la création de la base de donnée , la requête pour la création des tables <User> et <Connexion> ainsi la création de la méthode <adduser> et <addConnexion>

-User : Création de la table User avec les champs <user\_id>,<password> et <signup\_date> ainsi que le constructeur et getter et setter dans la classe <User>

-Connexion : Création de la table Connexion avec les champs <connexion\_id>,<connexion\_time>,<connexion\_downloads>,<connexion\_uploads> et <user\_id> ainsi que le constructeur et getter et setter .

-**Le Pacakage Ui** : contient les classes suivantes :

-ConnexionFragment : détecteur des évènements des clicks des buttons (Login) et (Signup).

-ConnexionViewModel :