

RAPPORT DE STAGE

Elaboré par



Application mobile de transfert des fichiers à base de serveur Web Http

Réalisé au sein de

DEDICACES

Toutes les lettres ne sauraient trouver les mots qu'il faut...

Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance...

Aussi, c'est tout simplement que je dédie ce projet de fin d'études...

Remerciements

C'est avec une grande gratitude et reconnaissance que je réserve ces lignes qui s'adressent à toutes les personnes qui ont contribué à l'élaboration de ce projet de fin d'études et qui m'ont aidé lors de la rédaction de ce rapport.

Sommaire

Dédicaces

Remerciements

Table des figures

Liste des abréviations

1. Introduction

2. Présentation de la société

3. Etude de besoin du projet

3.1 Diagramme de cas d'utilisation global

3.2 Diagramme MCD globale

3.3 Architecture d'une application Android

3.3.1 Le composant <Activity>

3.3.2 Le composant <Fragment>

4. Cycle de vie de projet

4.1 Introduction des phases

4.1.1 Conception Fonctionnelle

4.1.2 Design de l'interface Utilisateur

4.1.3 Conception de l'architecture

4.1.4 Codage

4.1.5 Test

5. Cycle de développement de l'application

5.1 Activité « Connexion »

5.1.1 Conception Fonctionnelle

5.1.2 Design de l'interface utilisateur

5.1.3 Conception d'architecture

5.1.4 Codage

5.1.5 Test

5.2 Activité « Configuration »

5.2.1 Conception Fonctionnelle

5.2.2 Design de l'interface utilisateur.

5.2.3 Conception d'architecture

5.2.4 Codage

5.2.5 Test

5.3 Activité « Files Transfer »

5.3.1 Conception Fonctionnelle

5.3.2 Design de l'interface utilisateur

5.3.3 Conception d'architecture

5.3.4 Codage

5.3.5 Test

5.4Activité « Administration »

5.4.1 Conception Fonctionnelle

5.4.2 Design de l'interface utilisateur

5.4.3 Conception d'architecture

5.4.4 Codage

5.4.5 Test

6 Validation

7 Conclusion

Listes des figures

Figure 1. Modèle en spirale	
Figure 2. Diagramme de cas d'utilisation globale.....	
Figure 3. Diagramme MCD globale	
Figure 4. Cycle de vie d'une activité.....	

.

Listes des abréviations

UML: Unified Modeling Language

XML: eXtensible Markup Language

1. 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.

Il s'agit en particulier de développer une application de transfert des fichiers à base de serveur web http qui sera réalisé avec trois technologies différentes :

- La 1^{ère} technologie : consiste à développer l'application avec le langage Java sur Android studio.
- La 2^{ème} technologie : consiste à développer l'application avec le langage Dart, le Framework Flutter sur Android studio .
- La 3^{ème} technologie : consiste à développer l'application avec la le langage JavaScript et la technologie ReactNative.

Ce rapport se limitera à la réalisation du projet avec la 1^{er} technologie et dont la période est programmée durant toute la période du stage.

Après la réalisation du projet avec les trois technologies une étude comparative sera élaborée permettant d'énumérer les avantages et les inconvénients de chacune des trois technologies.

2. Présentation de la société

3. Etude de besoin du projet

Lors de cette étude on a réalisé en particulier ce qui suit :

1. Le diagramme de cas d'utilisation global
2. Le modèle conceptuel de donnée global (MCD).
3. La détermination de la technologie à adopter (Android SDK) .

3.1 Diagramme de cas d'utilisation global :

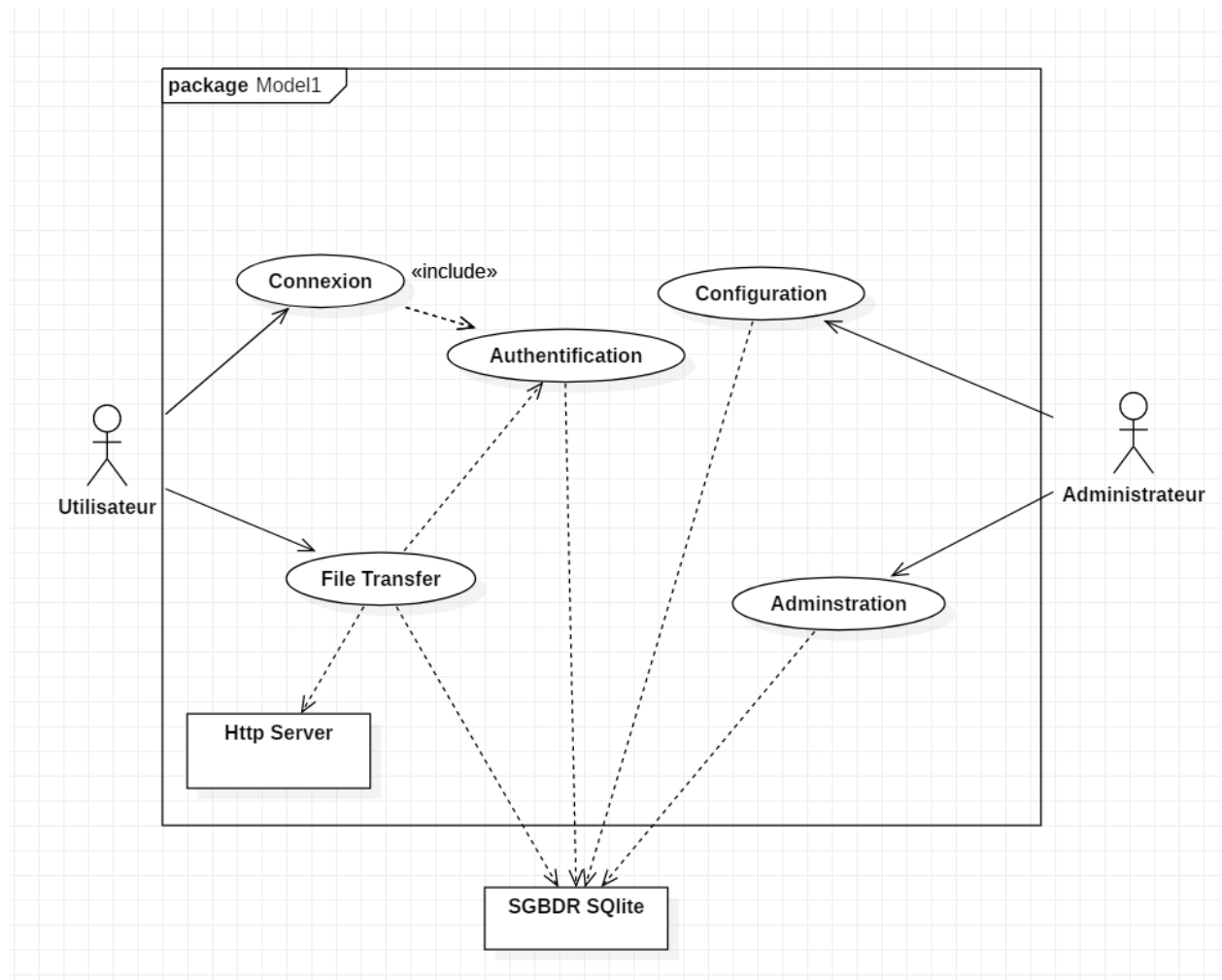


Figure1 : Diagramme de cas d'utilisation globale

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

- « Connexion » : permettant d'enregistrer et de connecter l'utilisateur.

- « Configuration » : permettant de mettre à jour les paramètres du serveur et de lancer/arrêter le serveur.
- « File Transfer » : permettant de se connecter vers un serveur distant et de télécharger/téléverser des fichiers
- « Administration » : permettant d'analyser la liste des utilisateurs et les transfert des fichiers.

3.2 Diagramme MCD globale :

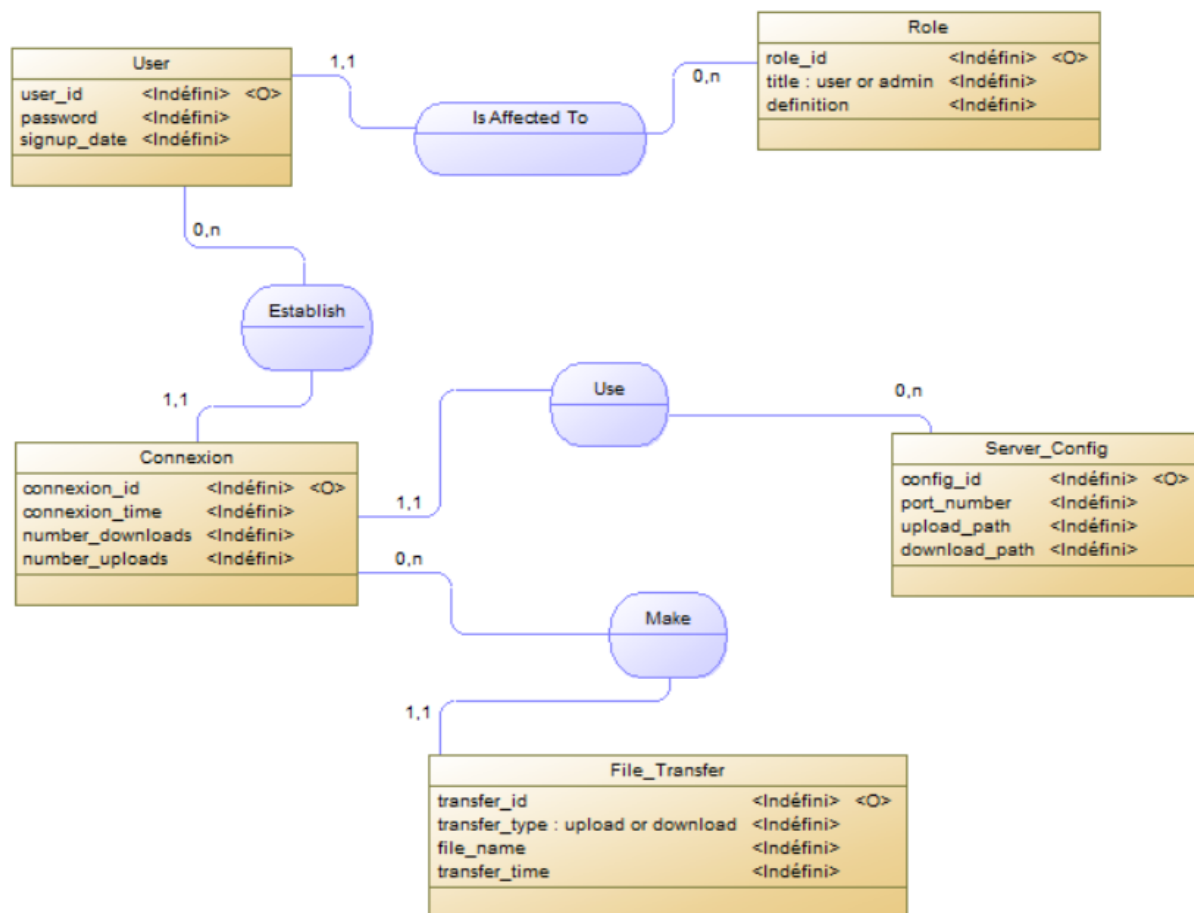


Figure2 : Diagramme MCD globale

On a les cinq tables suivantes :

- **Table « User »** : permettant d'enregistrer et d'authentifier les utilisateurs.
- **Table « Connexion »** : permettant d'enregistrer les connexions des utilisateurs.
- **Table « File_Transfer »** : permettant d'enregistrer les transferts des fichiers de type "download" et "upload".
- **Table « Server_Config »** : permettant de mettre à jour les paramètres de configuration du serveur qui sont : numéro du port, répertoire de téléchargement et répertoire de téléversement des fichiers.
- **Table « Rôle »** : permettant d'enregistrer les différents rôles à affecter aux utilisateurs qui sont "User" et "Admin".

3.3 Architecture de l'application

Une application Android comporte deux composants : « Activity » et « Fragment »

3.1.1 Le composant « Activity »

Faisons un peu de théorie. Une activité représente en gros ce que l'on voit à l'écran, pour la définir simplement, elle est généralement composée d'une classe, et d'un layout xml.

Une activité est la composante principale pour une application Android. Elle représente l'implémentation métier dans une application Android, permettant de gérer l'ensemble des vues et ressources. Une activité peut être avec ou sans interface utilisateur. Il est possible d'avoir plusieurs activités dans le même programme. Elle doit toujours être déclarée dans le fichier AndroidManifest.xml. Une activité n'est pas linéaire, elle est soumise à plusieurs événements. Chaque événement est représenté dans une méthode.

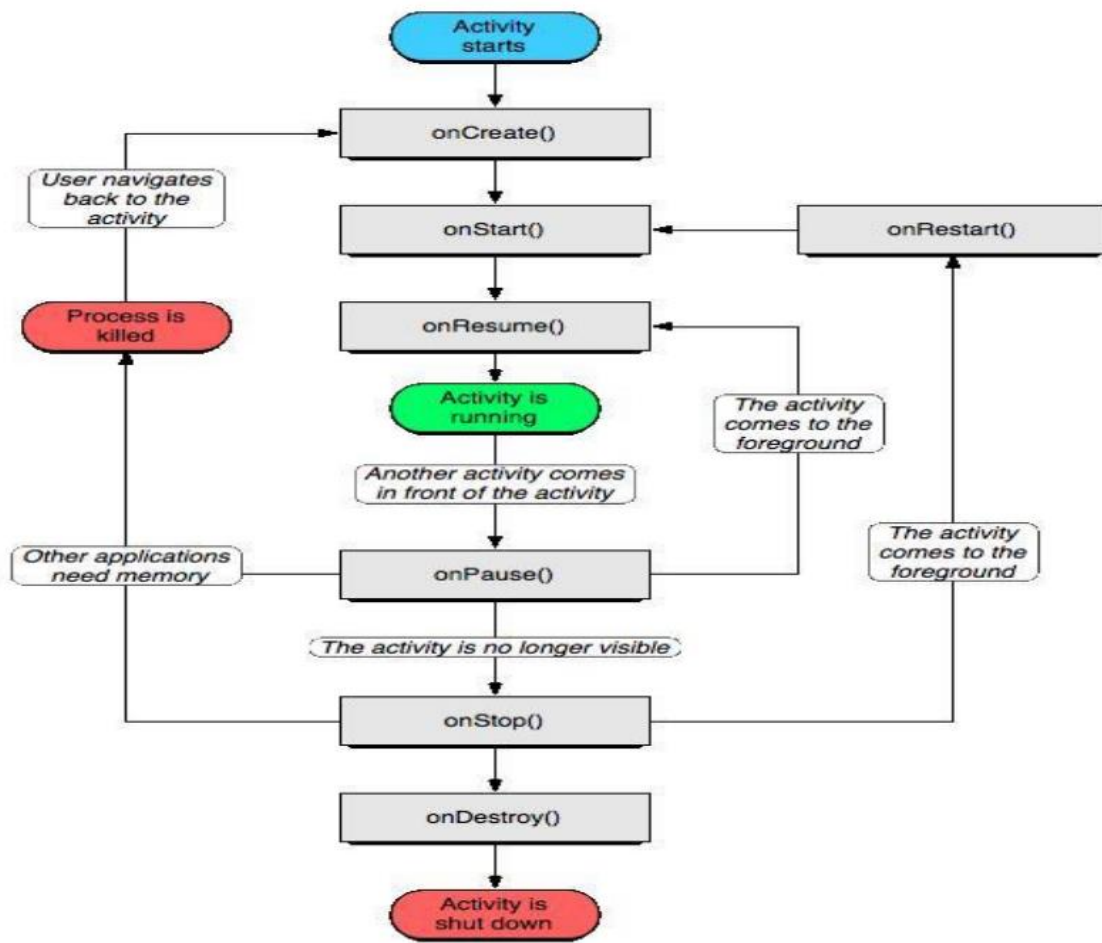


Figure3: Cycle de vie d'une activité

3.1.2 Le composante « *Fragment* »

Fragment est une partie d'une activité, qui contribue à sa propre interface utilisateur pour cette activité. Fragment peut être considéré comme une sous-activité. Ils sont utilisés pour une utilisation efficace de l'espace dans l'ensemble de l'écran des appareils.

Une activité peut contenir 0 ou plusieurs fragments basé sur la taille de l'écran. Un fragment peut être réutilisé dans de multiples activités, de sorte qu'il agit comme un composant réutilisable dans les activités.

Un fragment qui ne peut pas exister indépendamment. Il convient toujours de participer à une activité. Où que l'activité peut exister avec tout fragment en elle.

Le cycle de vie d'un fragment est plus complexe du cycle de vie d'une activité parce qu'il contient plus d'états.

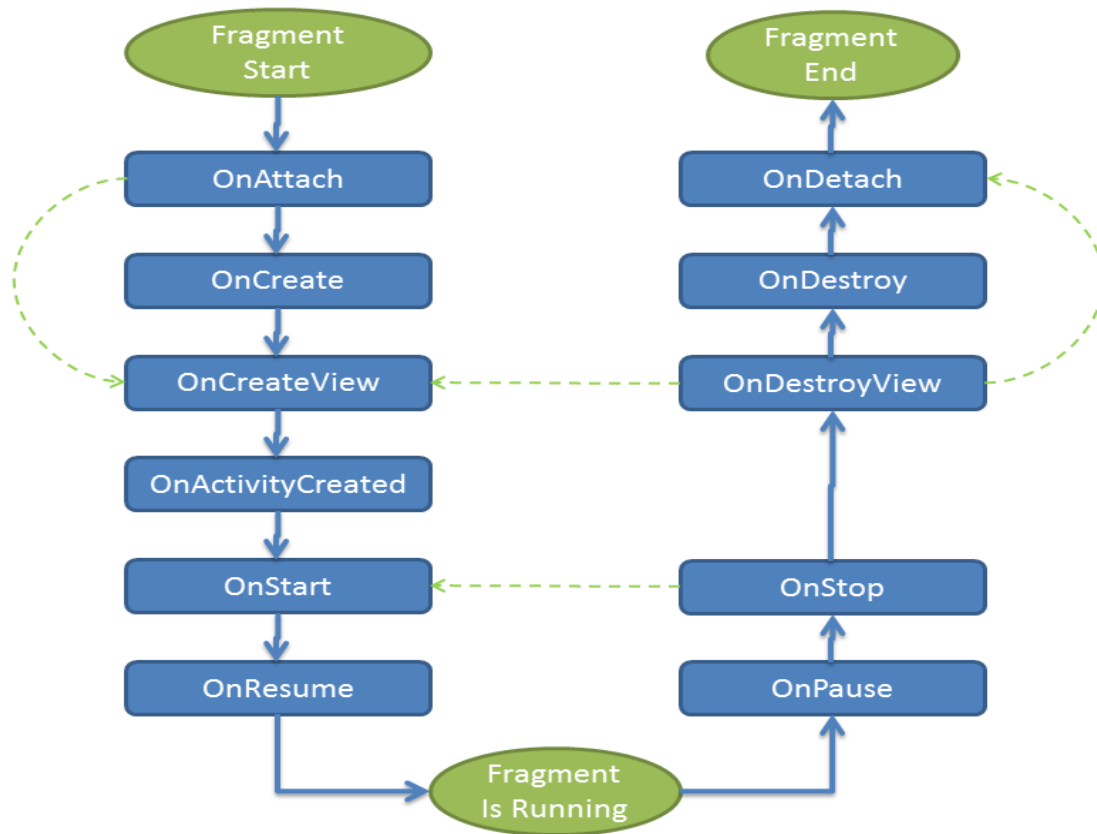


Figure 4 : Cycle de vie d'un fragment

4. Cycle de vie du projet

Le « cycle de vie d'un logiciel » ou les méthodes de développement sélectionne et identifie toutes les étapes de développement de logiciel, dès sa conception en allant jusqu'à livraison. 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 :

Le modèle en spirale (*spiral model*) est un modèle de Cycle de développement logiciel qui reprend les différentes étapes du cycle en V. Par l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet et dur. Le cycle en spirale met cependant plus l'accent sur la gestion des risques que le cycle en V.

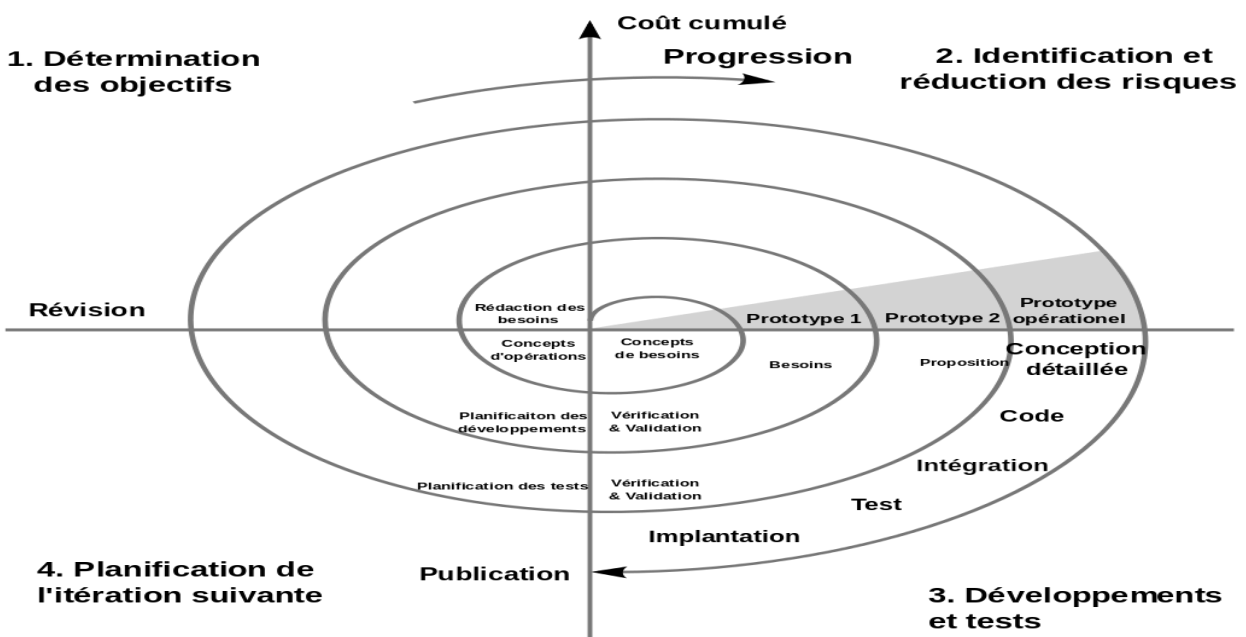


Figure 5 : Modèle en spirale

On commence d'abord par l'étude des besoins fonctionnelles :

L'**étude de besoins** est un élément déterminant dans la démarche projet. Elle permet de réaliser un diagnostic stratégique aidant ensuite à s'engager dans une réflexion de projet en cohérence avec le territoire.

D'après le diagramme de cas d'utilisation réalisé par l'étude de besoin on a identifié 4 activités principales pour notre application :

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

Pour chacune de ces activités on passe par les 5 phases suivantes :

1. Conception Fonctionnelle :
2. Design de l'interface utilisateur :
3. Conception de l'architecture
4. Codage
5. Test

4.1 Introduction des phases

4.1.1 Conception Fonctionnelle

Nous avons utilisé pour la partie conception fonctionnelle le langage UML (Unified Modeling Language) : c'est un langage de modélisation unifié permettant de modéliser une application logicielle d'une façon standard dans le cadre de conception orienté objet.

On a utilisé en particulier les deux diagrammes « cas d'utilisation » et diagramme « d'activité » .

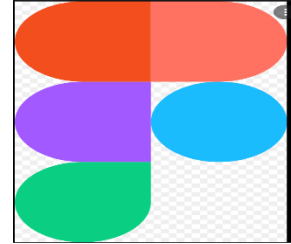
-def diagramme cas utilisation :

Def diagram d'activités :

4.1.2 Design de l'interface utilisateur

Nous avons utilisé pour la partie design la plateforme web « 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.



4.1.3 Conception de l'architecture

Pour la modélisation des données on a la méthodologie merise permettant de réaliser le modèle conceptuel des données et le modèle physique des données.

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.

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.

Nous avons utilisé le logiciel Power AMC pour modéliser le diagramme ER (entité-relation) permettant de générer le MPD de la base de données, c.-à-d. sa structure finale physique.

La méthode 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.

4.1.4 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.

Les technologies utilisées pour le codage de 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.



Android Studio 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. **G**it 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.



4.1.5 Test

5. Cycle de vie développement de l'application

5.1 Détermination des activités de l'application

Notre application comporte les 4 activités suivantes :

- Activité « Connexion »
- Activité « Configuration »
- Activité « Files Transfer »
- Activité « Administration »

5.1.1 Activité Connexion

Dans cette partie nous avons travaillé sur l'Activité Connexion de l'application Android de notre projet.

5.1.1.1 Conception Fonctionnelle

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

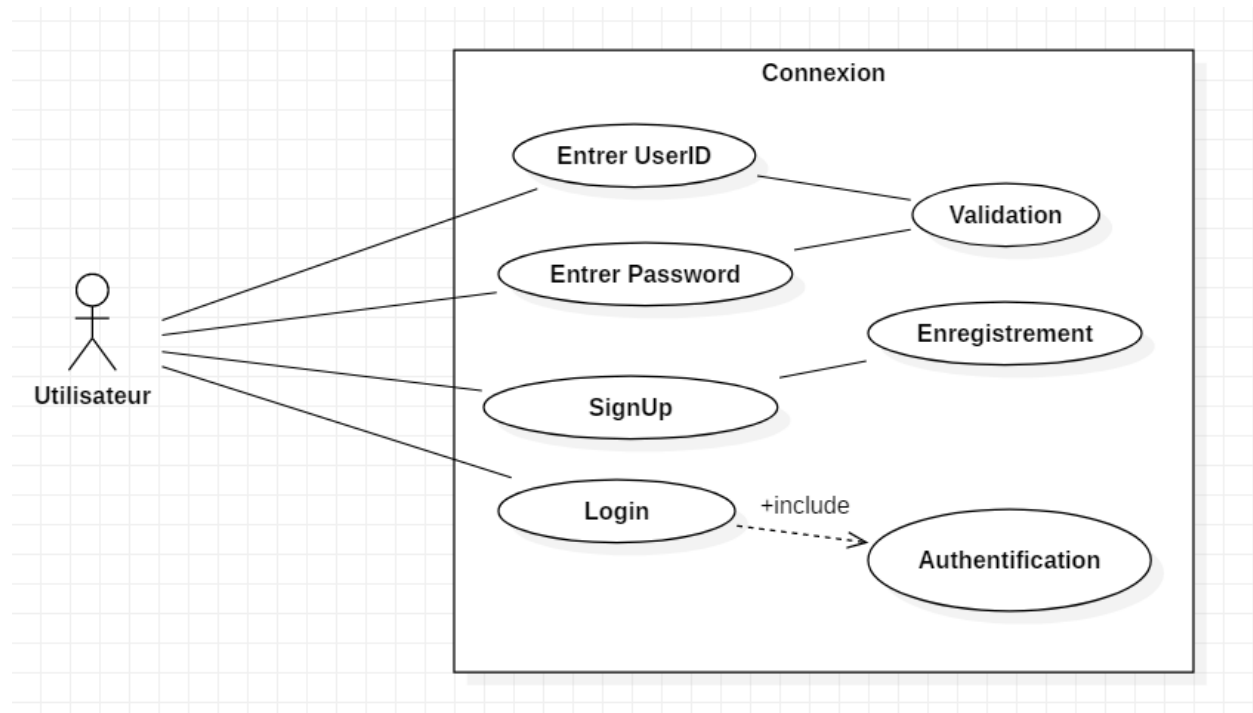


Figure : Diagramme cas d'utilisation

-le diagramme d'activité pour l'opération SignUp :

Dans cette partie l'utilisateur créer un compte.

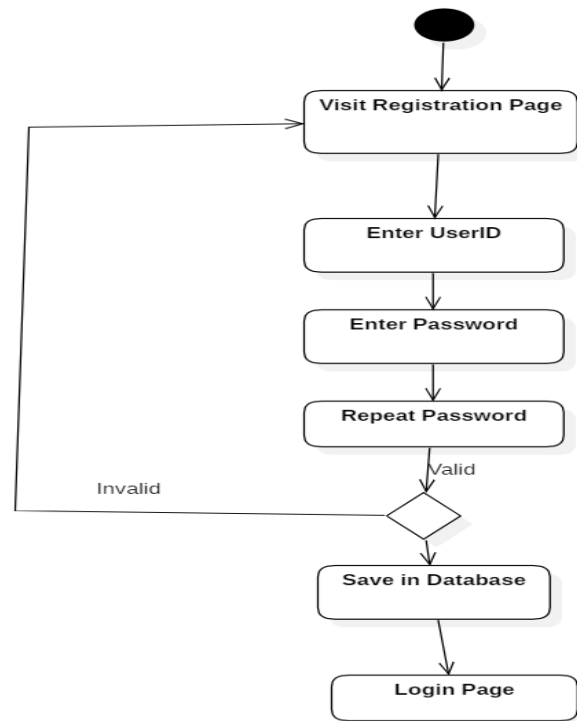


Figure : le diagramme d'activité pour l'opération SignUp :

-le diagramme d'activité pour l'opération Login :

Dans cette partie l'utilisateur saisie (userid) et (password) pour s'identifier a l'application.

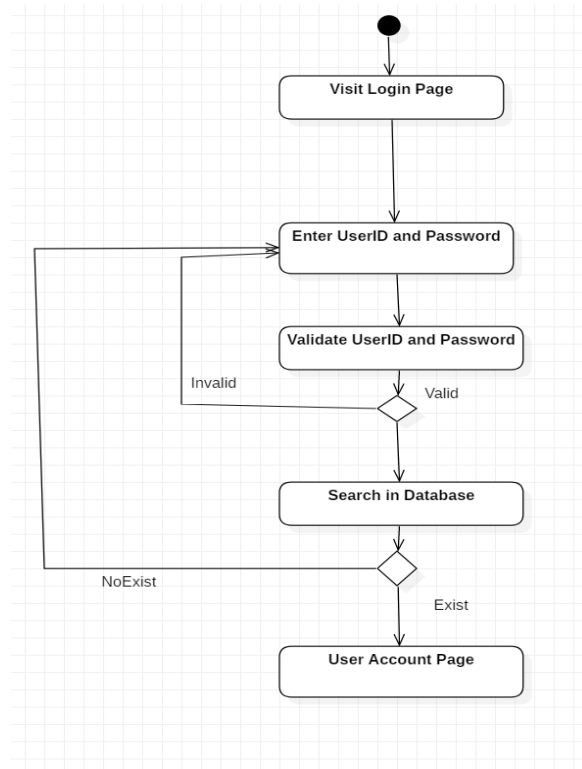


Figure : le diagramme d'activité pour l'opération Login :

5.1.1.2 Design

Pour lancer l'application il faut créer un compte dans l'activité Connexion lors de clique sur le bouton « Signup » il faut créer un « User ID » et « Password » selon les conditions Ci-dessous la figure de l'activité Connexion :

10:39

Connexion

User ID:

Password:

LOGIN SIGNUP

UserID must be:

- at least 1 digit
- at least 1 lower case letter
- only letters and numbers
- size between 8 and 15 characters
- no white spaces

Password must be :

- at least 1 digit
- at least 1 lower case letter
- at least 1 symbol
- at least 1 upper case letter
- size between 8 and 15 characters
- no white spaces

Figure :Activité Connexion

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

- au moins 1 chiffre
- au moins une lettre minuscule
- au moins une lettre majuscule
- n'importe quelle lettre
- pas d'espace.
- Taille entre 8 et 15 caractères.

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

- au moins 1 chiffre
- au moins une lettre minuscule
- uniquement des lettres minuscules et des chiffres
- pas d'espace.
- Taille entre 8 et 15 caractères.

5.1.1.3 Conception d'architecture

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.

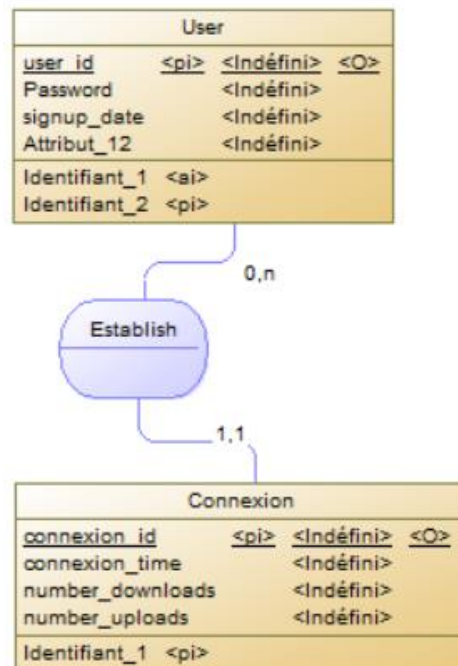


Figure : entité relation

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.

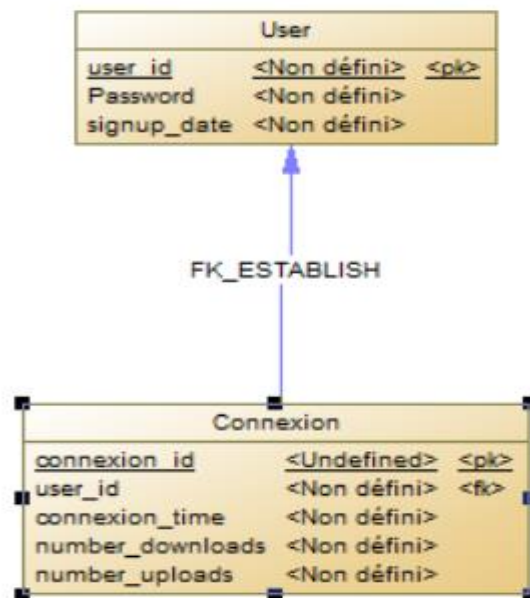
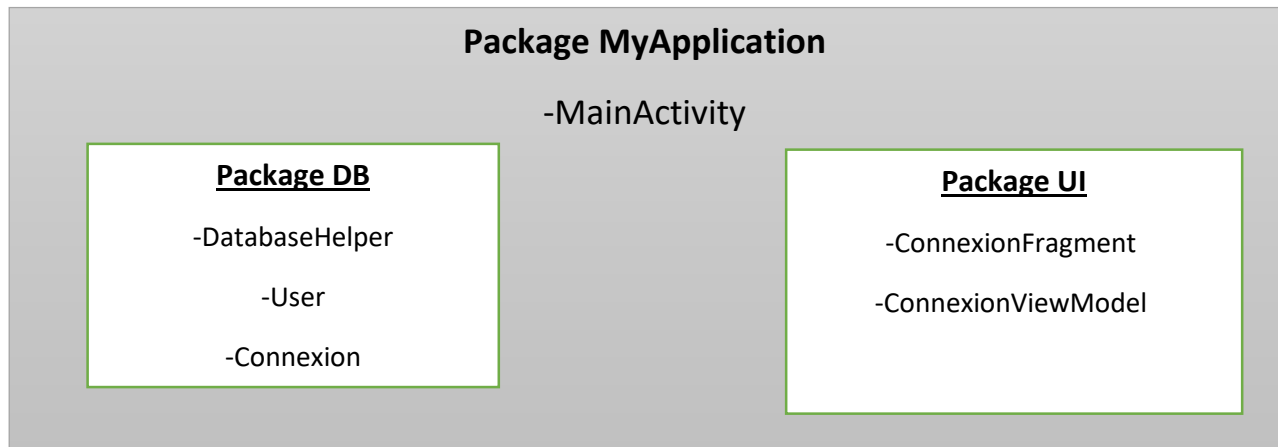


Figure :Diagramme MPD

5.1.1.4 Codage



-MainActivity : est la classe de démarrage 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 :

les expressions régulières pour la validation des champs (UserID et Password) :

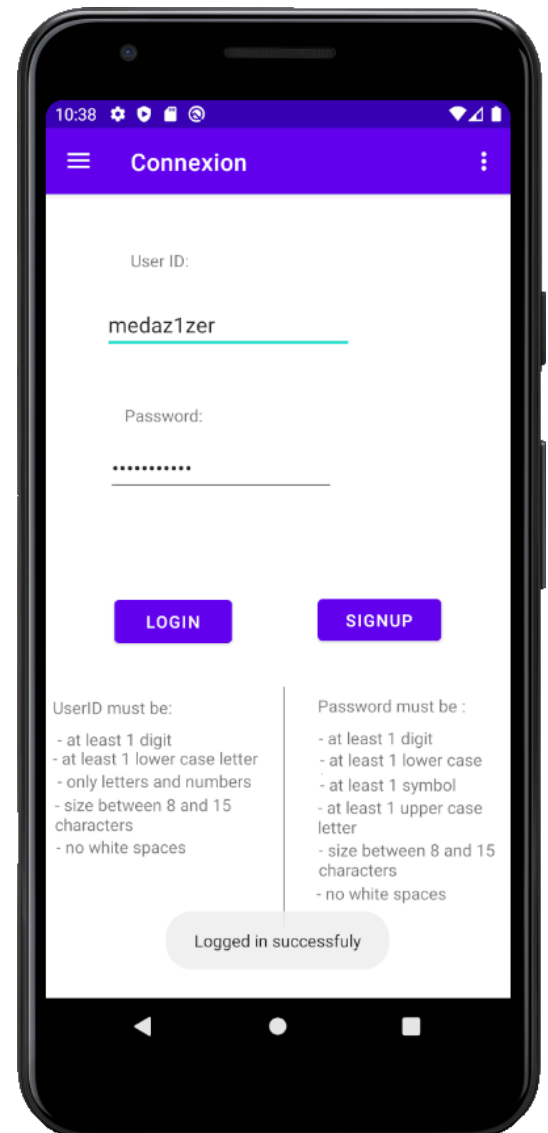
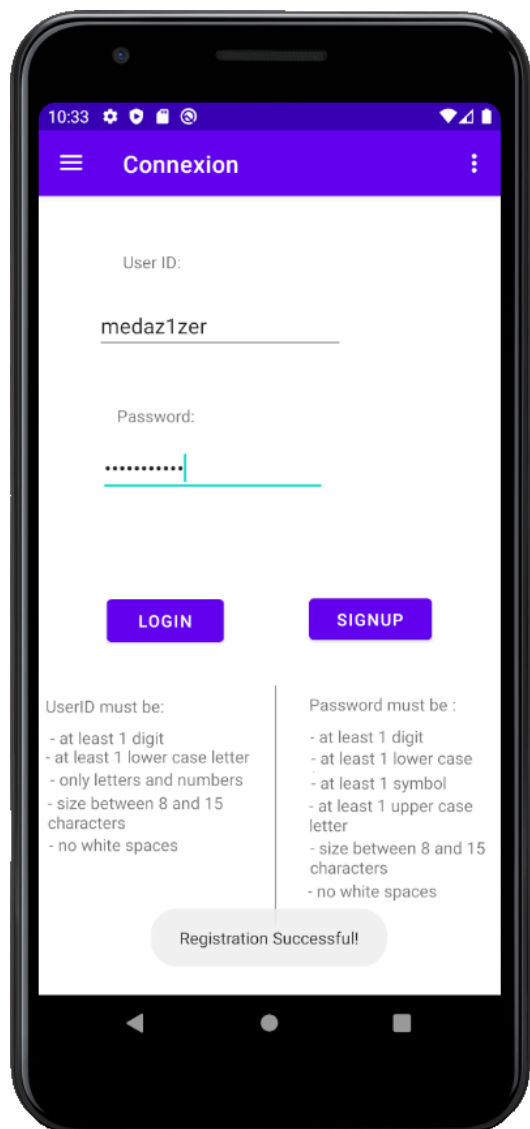
Pour le champ User ID :

Conditions de Validation	Expression réguliers
Taille entre 8 et 15 caractères	<code>^.{8,15}\$</code>
au moins un chiffre	<code>^.*[0-9]+.*\$</code>
au moins une lettre minuscule	<code>^.*[a-z]+.*\$</code>
seulement des lettres et des chiffres	<code>^[a-z0-9]+\$</code>
pas d'espace	<code>^\\s+\$</code>

Pour le champ Password :

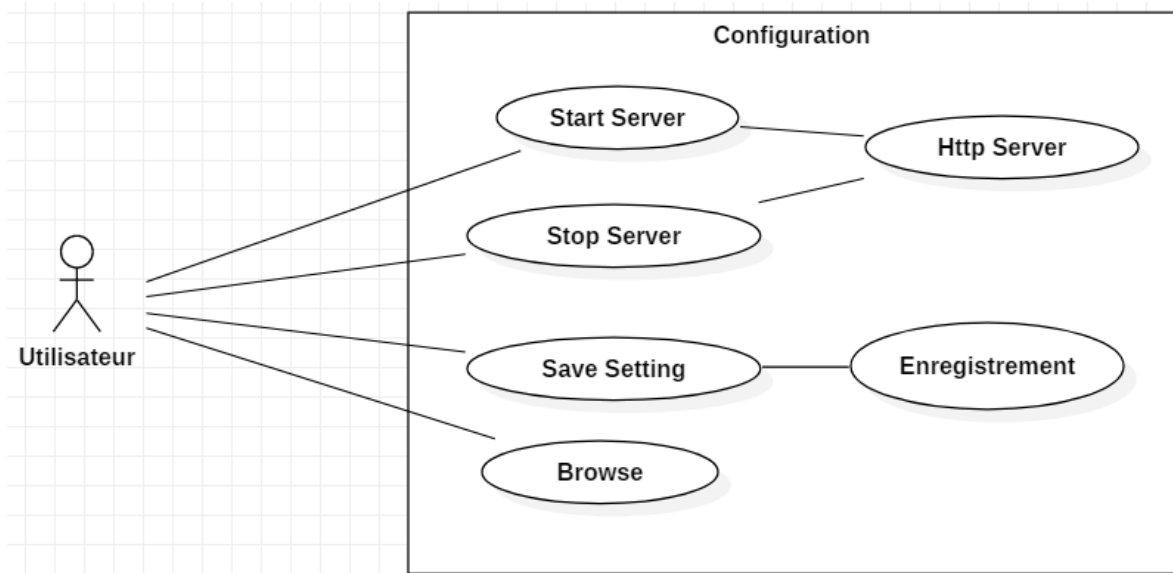
Conditions de Validation	Expression réguliers
Taille entre 8 et 15 caractères	<code>^.{8,15}\$</code>
au moins un chiffre	<code>^.*[0-9]+.*\$</code>
au moins une lettre minuscule	<code>^.*[a-z]+.*\$</code>
seulement des lettres et des chiffres	<code>^[a-z0-9]+\$</code>
pas d'espace	<code>^\\s+\$</code>
au moins une lettre majuscule	<code>^.*[A-Z]+.*\$</code>
n'importe quelle lettre	<code>^.*[@#\$\$%^&+=]+.*\$</code>

5.1.1.5 Test



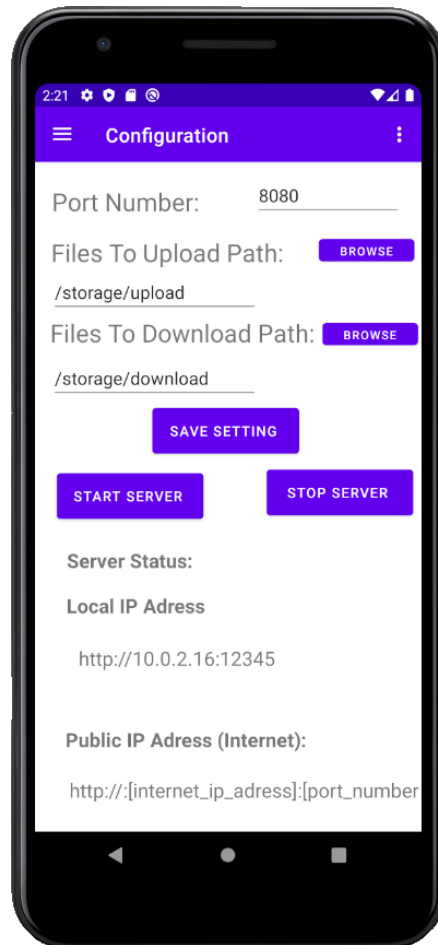
5.2 Activité Configuration

5.2.1 Conception Fonctionnelle



5.2.2 Design

Pour démarrer le serveur l'application il faut appuyer sur le bouton <StartServer> et pour l'arrêt il faut appuyer sur le bouton <StopServer>, Server Status affiche l'état de serveur et Local Ip adress permet d'afficher l'adresse ip . Le bouton <SaveSetting> permet d'enregistrer le numéro de port ainsi que d'ajouter dans la base de donnée. Ci-dessous la figure de l'activité Connexion :



5.2.3 Conception d'architecture

-Protocol tcp/ip : (Transmission Control Protocol/Internet Protocol) réunit les deux protocoles TCP et IP. Il s'agit donc d'une suite de protocoles associée au domaine d'Internet pour lequel elle facilite le transfert de données.

Présenté simplement, le protocole TCP/IP est un standard de communication entre deux processus. Il détermine et fixe les règles inhérentes à l'émission et à la réception de données sur un réseau. L'association des deux protocoles permet d'apporter des garanties de fiabilité dans le transfert des données. Avec le TCP/IP, vous êtes certain(e) que les informations envoyées arriveront bel et bien au bon destinataire. (+photo).

-Protocol http : HTTP signifie « **Hypertext Transfer Protocol** ». Ce protocole a été développé par Tim Berners-Lee au CERN (Suisse) avec d'autres concepts qui ont servi de base à la création du World Wide Web : le HTML et l'URI. Alors que le HTML (Hypertext Markup Language) définit comment un site Internet est construit, le HTTP détermine comment la page est transmise du serveur au client. Le troisième concept, l'URL (Uniform Resource Locator), fixe la façon dont une ressource (par exemple un site Internet) doit être adressée sur le Web.

-Protocol ftp : Un serveur FTP (File Transfer Protocol) est un logiciel utilisé dans le transfert de fichiers entre deux ordinateurs. Il est, avec le client FTP, l'une des deux composantes d'un transfert de fichiers via le langage FTP.

<u>HTTP</u>	<u>FTP</u>
Il ne prend en charge que la connexion de données.	Il prend en charge à la connexion de données et la connexion de contrôle
Il utilise le protocole de contrôle de transmission sur le port Tcp 80.	Il utilise le protocole de contrôle de transmission sur le port Tcp 20 et 21.
L'URL utilisant le protocole http commencera par HTTP	L'URL utilisant le FTP commencera par FTP

-Socket : Une socket est connue comme un type de logiciel qui agit comme un point d'extrémité qui fonctionne en établissant une liaison de communication réseau bidirectionnelle entre l'extrémité du serveur et le programme de réception du client. On l'appelle aussi souvent un point d'aboutissement dans un canal de communication bidirectionnel. Ces sockets sont réalisés et mobilisés en même temps qu'un ensemble de requêtes de programmation identifiées comme appels de fonction, qui est techniquement appelé interface de programmation d'application (API). Une socket est capable de simplifier le fonctionnement d'un programme car les programmeurs n'ont plus qu'à se soucier de manipuler les fonctions de la socket, ce qui leur permet de compter sur le système d'exploitation pour transporter correctement les messages sur le réseau.

5.2.4 Codage

Les permissions d'utilisateurs d'Android :

Chaque application Android s'exécute dans un bac à sable à accès limité. Si votre application doit utiliser des ressources ou des informations en dehors de son propre bac à sable, vous pouvez déclarer une autorisation et configurer une demande d'autorisation qui fournit cet accès. dans notre application nous avons utilisé les permission suivantes :

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
```

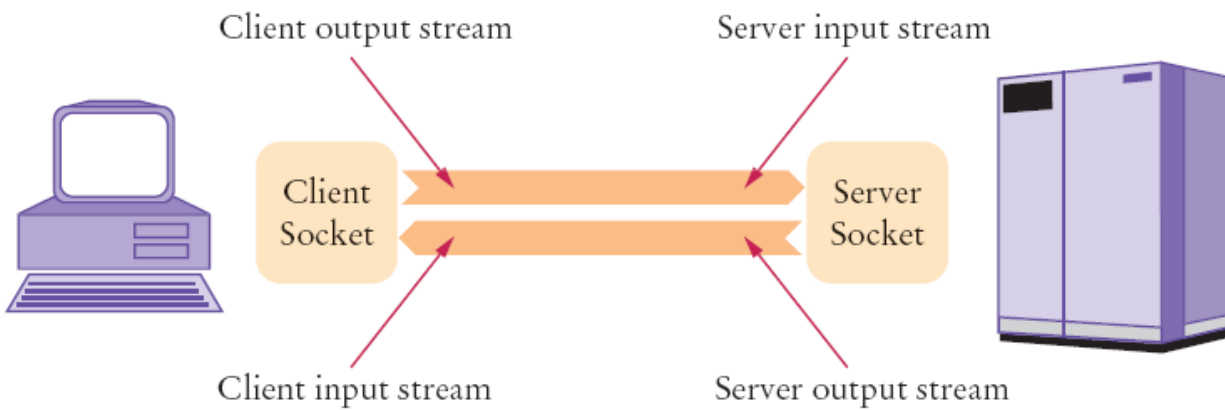
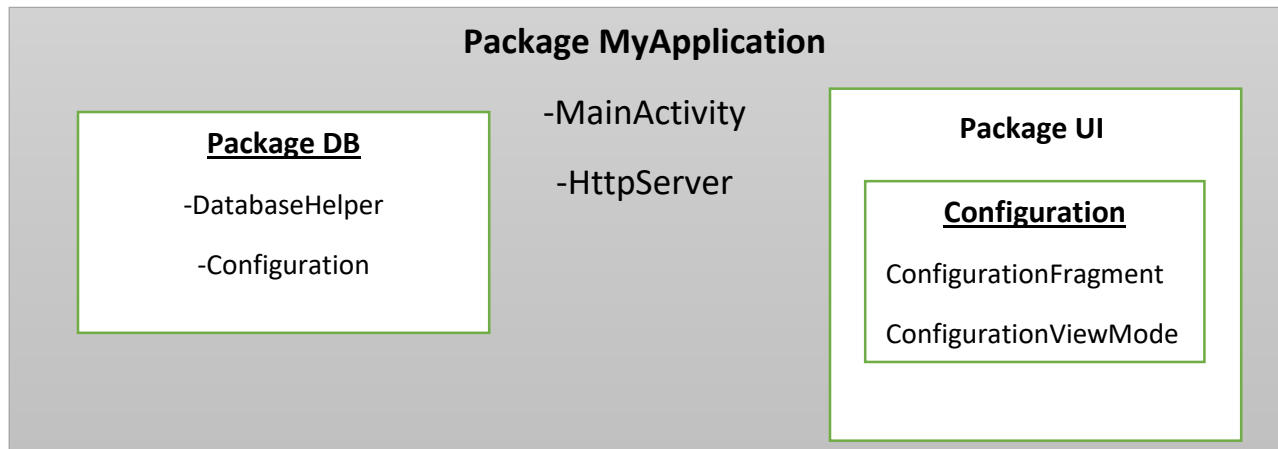
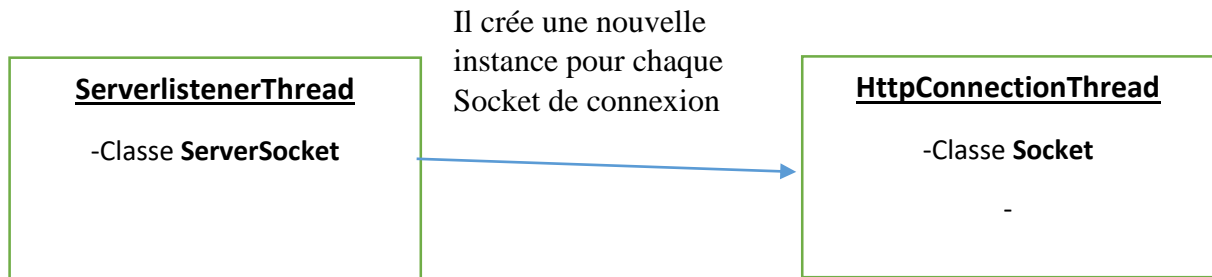


Figure 5 Client and Server Sockets

-On a une seule instance de type `ServerlistenerThread` permettant de faire fonctionner le serveur http en arrière plan.

-Pour chaque nouvelle connexion on crée une nouvelle instance de type `HttpConnectionThread` permettant de gérer les communications via la Socket de connexion.



Package com.fst.myapplication

Android SDK

- Activity
- Fragment
- AndroidManifest.xml

Java JDK

Java.io

- InputStream
- OutputStream

Java.net

- ServerSocket
- Socket

5.2.5 Test

le serveur est en marche : (Server Status : Running !)

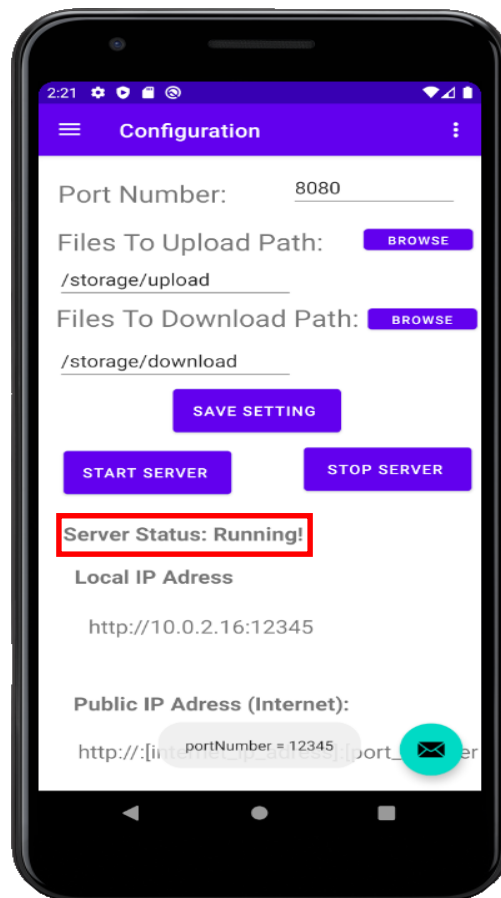
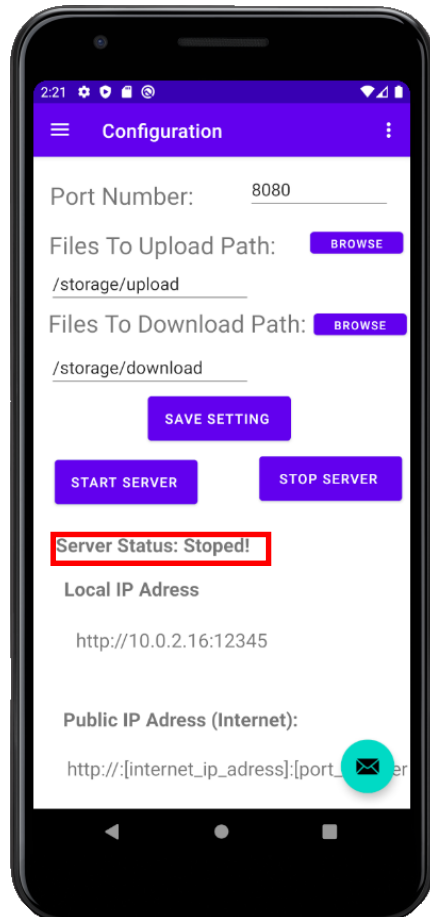


Figure : serveur en marche

```
Emulator Pixel_3a_API_react And  com.fst.myapplication (30086)  Verbose  log
2021-09-13 19:21:44.832 30086-30086/com.fst.myapplication D/Log: portNumber:12345
2021-09-13 19:21:44.832 30086-30086/com.fst.myapplication D/Log: Server is running on port: 12345
```

le serveur est en arrêt : (Server Status : Stopped !)



Emulator Pixel_3a_API_react And com.fst.myapplication (30086) Verbose log

```

2021-09-13 19:21:44.832 30086-30086/com.fst.myapplication D/log: portNumber:12345
2021-09-13 19:21:44.832 30086-30086/com.fst.myapplication D/log: Server is running on port: 12345
2021-09-13 19:28:38.408 30086-30361/com.fst.myapplication D/log: Server exception: Socket closed
2021-09-13 19:28:38.408 30086-30361/com.fst.myapplication D/log: ServerSocket closed!

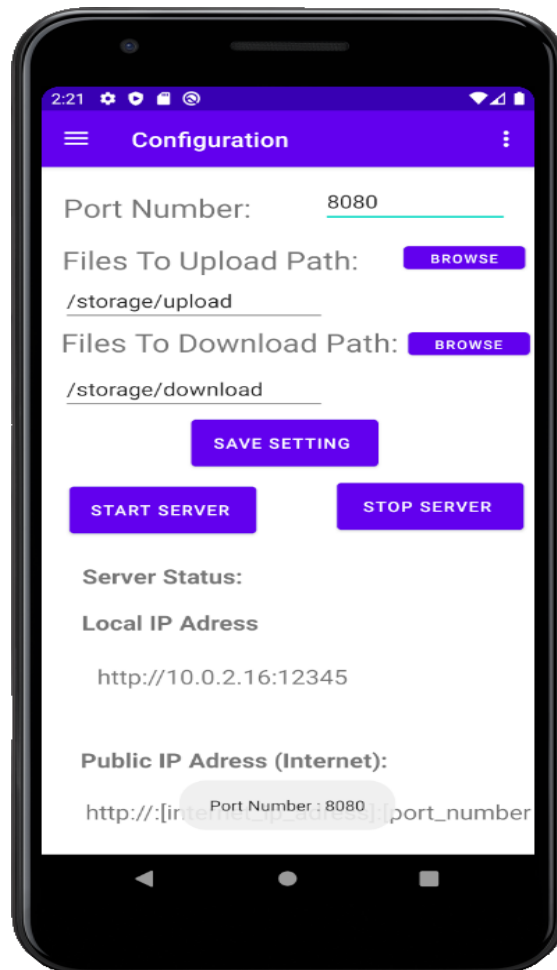
```

Le Button
<SaveSetting> : permet d'ajouter une ligne dans la table de configuration ou bien mettre à jour d'une ligne existante dans la base de données.

Ajouter une ligne dans table de configuration avec le numéro de port ajouté 8080 :

Table: Configuration

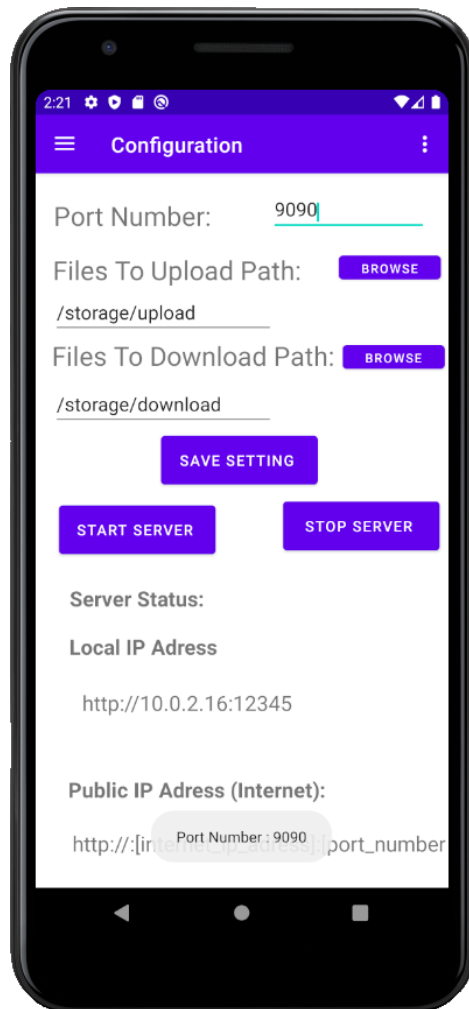
	config_id	port_number	uploads_path	download_path
	Filter	Filter	Filter	Filter
1	1	8080	/storage/upload	/storage/download



Mettre à jour la table de configuration avec le nouveau numéro de port ajouté 9090 :

Table: Configuration

	config_id	port_number	uploads_path	download_path
	Filter	Filter	Filter	Filter
1	1	9090	/storage/upload	/storage/download



5.3 Activité File Transfer

5.3.1 Conception Fonctionnelle

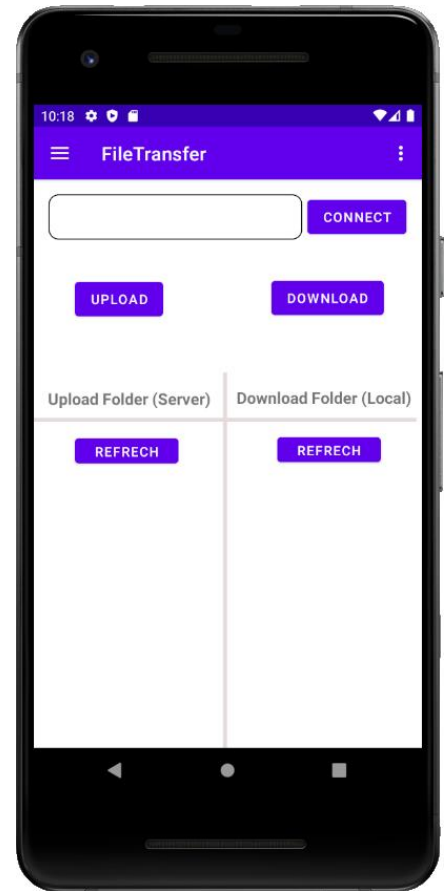
5.3.2 Design

L'utilisateur se connecte à un serveur distant une fois le serveur est connecté le Server Status est en marche (Server Status : is Running !) .

L'utilisateur permet de télécharger des fichiers à partir de <Upload Server> , il sélectionne le fichier à télécharger puis en cliquant sur le bouton <Download> une progression de téléchargement est affichée sous le bouton , une fois le téléchargement est terminé , il faut appuyer sur le bouton < Refresh> pour vérifier que le téléchargement est bien téléchargé dans le répertoire de téléchargement local de l'appareil utilisateur .

Pour le téléversement des fichiers dans le serveur distant :

L'utilisateur sélectionne les fichiers à téléverser depuis le répertoire <Download Local> puis en cliquant sur le bouton <Upload> une progression de téléversement est affichée sous le bouton , une fois le téléversement est terminé les fichiers seront téléversés dans <Upload Server>



5.3.3 Conception d'architecture

5.3.4 Codage

5.3.5 Test

5.4 Activité Administration

5.4.1 Conception Fonctionnelle

5.4.2 Design

5.4.3 Conception d'architecture

5.4.4 Codage

5.4.5 Test

6 Validation

7 Conclusion