

# Conception Générale

## Spécifications techniques et logicielles

---

|                      |   |                |                                 |
|----------------------|---|----------------|---------------------------------|
| <b>Client</b>        | IMERIR  | <b>Projet</b>  | A.R.E IMERIR Application Mobile |
| <b>Créé le</b>       | 19/12/2016  | <b>MAJ</b>     | Jj/mm/aaaa                      |
| <b>Auteur</b>        | BOUILLON - SIRE                                       | <b>Version</b> | 2.1                             |
| <b>Destinataires</b> | Chef de projet  |                |                                 |
| <b>Fichier</b>       | Conception_Generale_A.R.E_IMERIR_BOUILLON_SIRE_V1.odx |                |                                 |

| Version | Modifié le | Par  | Modifications |
|---------|------------|------|---------------|
| 1       | 19/12/2016 | SIRE |               |
| 1.1     | 21/12/2016 | SIRE |               |
| 2       | 08/03/2017 | SIRE |               |
| 2.1     | 03/05/2017 | SIRE |               |
| 2.2     | 29/05/2017 | SIRE |               |

### Sommaire

|                               |
|-------------------------------|
| I - CONTEXTE                  |
| II – DESCRIPTION GENERALE     |
| III – CONCPETION PRELIMINAIRE |
| IV – ARCHITECTURE TECHNIQUE   |
| V– REGLES DE CODAGE           |
| VI – PROGRAMME DE TEST        |

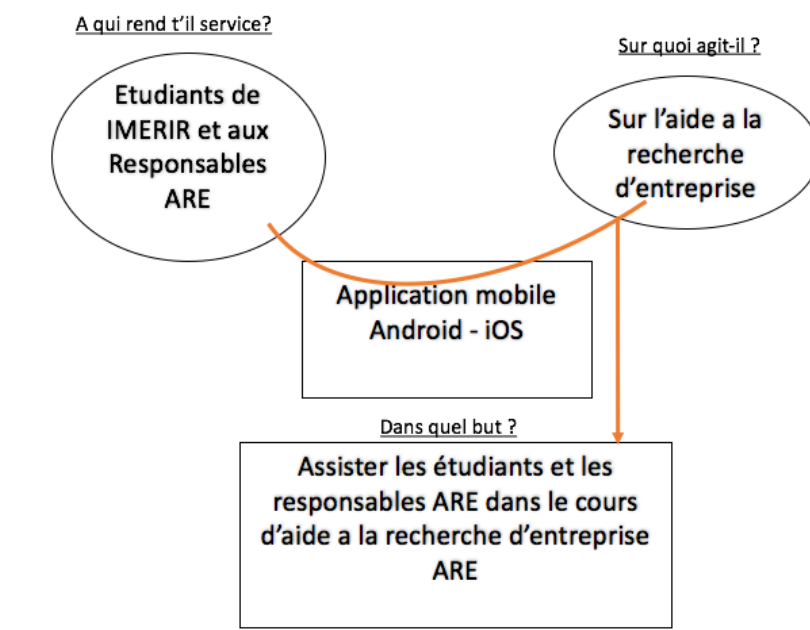
# I - Contexte

---

Ce document a pour but de regrouper toutes les spécifications techniques et logicielles que nous utiliserons pour avoir une application mobile iOS et Android permettant aux responsables IMERIR de l'ARE de déposer des offres d'emploi, des documents exemples et d'avoir un suivi des étudiants. Les étudiants pourront consulter les offres, les documents n'importe quand pour les aider dans leur recherche d'emploi.

L'application mobile réalisée proposera plusieurs fonctions telles que la consultation, la publication d'offres d'emploi, de documents, fonctions de suivi... L'objectif principal sera de proposer un accès à toutes ces fonctions qu'importe l'endroit où se trouve un utilisateur tant qu'il dispose d'un appareil compatible et d'une connexion internet.

Voici notre bête à cornes qui est un outil d'analyse fonctionnelle du besoin qui formule le besoin sous forme de fonctions simples que devra remplir l'application.



Ressources disponibles :

- ❖ Proposition Technique Commerciale
- ❖ Cahier des Charges Fonctionnel
- ❖ Documentation API
- ❖ Aussi disponible sur /doc depuis notre serveur
- ❖ Maquette application Android
- ❖ Veille Technologique « Interfaces Graphiques » et Etat de l'art « applications de recherche d'emploi »
- ❖ Note de cadrage
- ❖ Manuel d'utilisation

# II - Description générale

---

## Enoncé du besoin

Le but de cette application est de proposer un outil mobile qui sera utile aux Etudiants ainsi qu'aux responsables ARE pour appuyer le cours d'Aide à la Recherche d'Entreprise.

Cette application sera disponible seulement pour les imeriens et IMERIR et pourra être utilisée partout, le but est de pouvoir rapidement vérifier les offres disponibles peu importe l'endroit où se situe l'étudiant, les responsables ARE eux pourront déposer des offres que les entreprises leurs envoient et aussi avoir un suivi des étudiants.

Nous réalisons cette application dans le cadre d'un projet pour nos études afin d'obtenir de l'expérience dans le domaine du développement des applications mobiles et à terme de proposer un outil pratique et utilisé par les étudiants et l'école. Le résultat attendu sera de fournir une application native iOS et Android correspondant au cahier des charges défini.

### Les acteurs principaux

- **Administrateur** : L'administrateur est l'acteur qui gère l'installation, la maintenance, les sauvegardes et les utilisateurs de l'application. C'est lui qui donne l'accès au téléchargement de l'application et qui permet à un responsable ARE de créer son compte.
- **Etudiant** : Un étudiant est un utilisateur de l'application. Il doit posséder un Smartphone compatible avec l'application. L'étudiant doit pouvoir se servir de l'application pour l'aider dans sa recherche d'entreprise.
- **Responsable ARE** : Un responsable ARE est un utilisateur de l'application. Il doit posséder un Smartphone compatible avec l'application. Le responsable doit pouvoir se servir de l'application afin d'assister au mieux les étudiants en recherche d'entreprise.
- **Compte utilisateur étudiant** : Un compte utilisateur étudiant est un compte qui permet à un étudiant de l'IMERIR d'utiliser l'application. Il ne peut être créé qu'avec une adresse mail IMERIR. Le compte étudiant ne peut pas accéder à toutes les fonctionnalités de l'application (comme publier une offre).
- **Compte utilisateur responsable ARE** : Un compte utilisateur responsable ARE est un compte qui permet à un responsable ARE d'utiliser l'application. Il ne peut être créé qu'avec une adresse mail IMERIR et un code spécifique donné par un administrateur à la création d'un compte responsable afin que les étudiants ne puissent pas en créer un. Le compte Responsable aura accès à toutes les fonctionnalités de l'application.
- **Message d'accueil** : Le message d'accueil est un message qui sera affiché à l'ouverture de l'application. Il est publié par les responsables ARE dans le but de motiver, informer les étudiants. Seuls les responsables peuvent publier un message d'accueil. Si un message d'accueil est mis à jour cela enverra une notification sur le Smartphone.
- **Offre d'emploi** : Une offre d'emploi est uniquement publiée par un responsable ARE, elle contient les informations nécessaires pour qu'un étudiant intéressé puisse postuler à l'offre. Une offre d'emploi peut être publiée, consultée, supprimée.
- **Document personnel** : Un document personnel est un document que les étudiants pourront charger dans leur profil (ex CV, Tableau de bord...). Ces documents permettront aux responsables d'aider au mieux les étudiants dans leur recherche. Les documents ne pourront être chargés seulement au format PDF.
- **Profil étudiant** : Le profil étudiant devra être complété par l'étudiant à la création de son compte et être tenu à jour. C'est ce qui permettra aux responsables ARE d'effectuer un suivi personnel afin

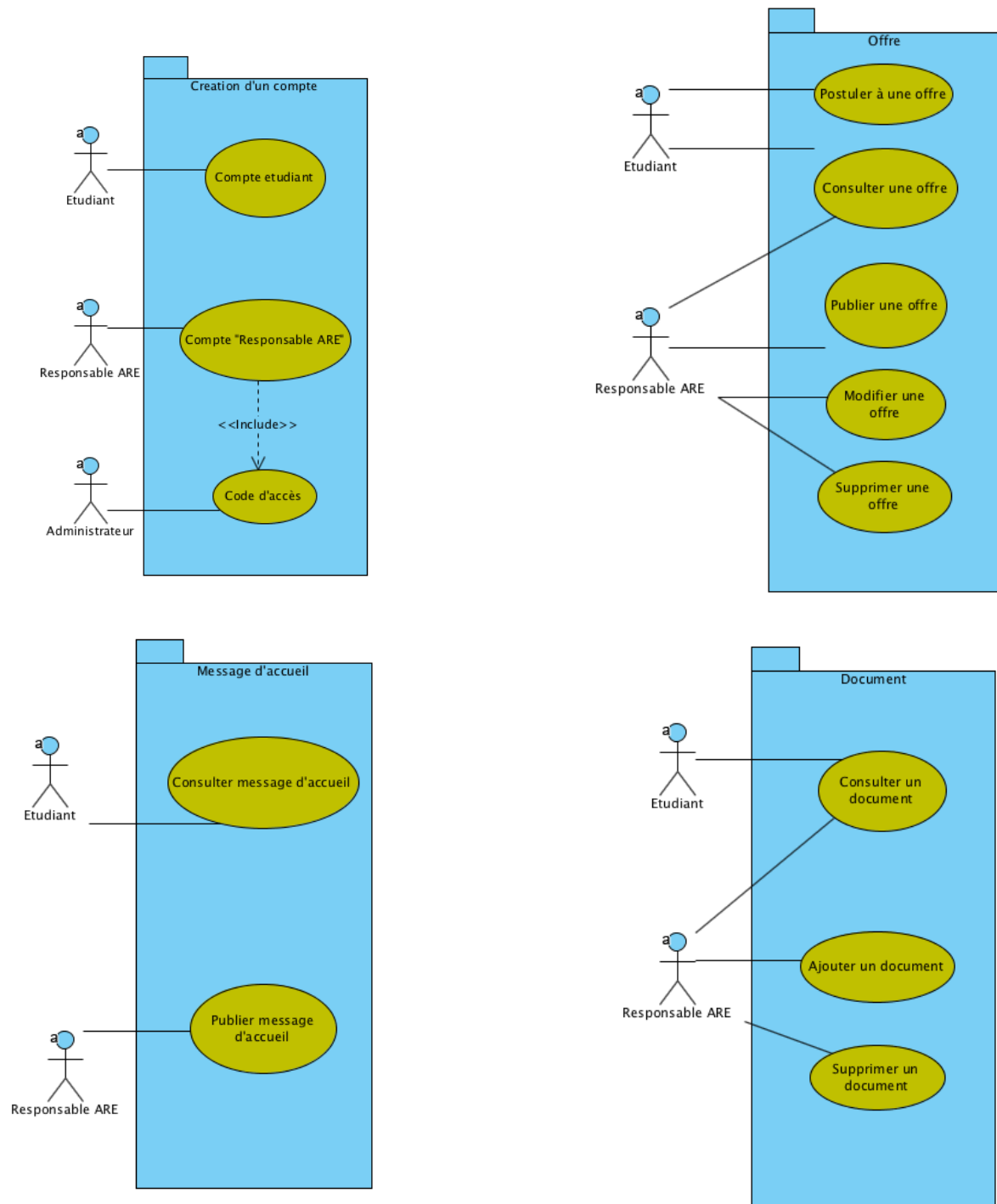
d'adapter au mieux sa méthode de suivi et d'aide. L'étudiant aura la possibilité d'y déposer ses documents personnels. Les profils étudiants seront consultables par les responsables.

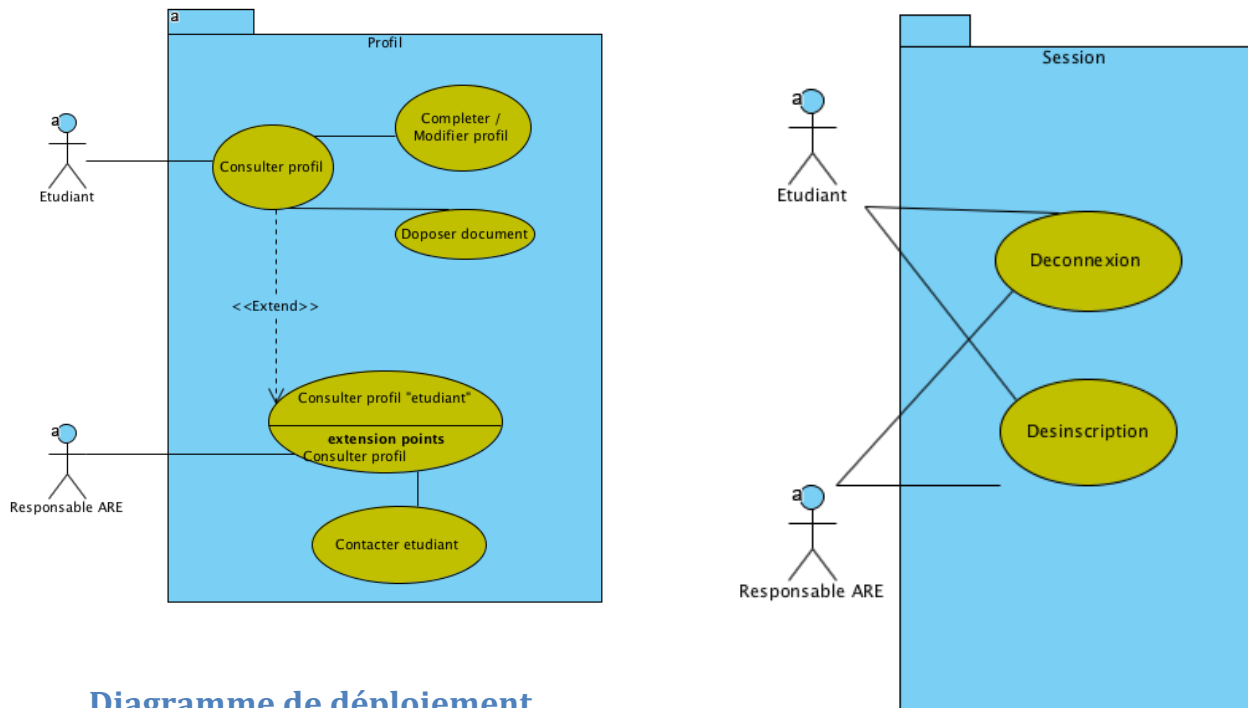
| <b>Fct</b>                                    | <b>Description</b>  |
|---|---|
| <b>FT1 :<br/>Publier une offre</b>            | Permettre à IMERIR ou à un professeur de l'école de publier des offres d'emploi ou de stage.<br>Stage, Offre d'apprentissage, Offre de professionnalisation, Offre d'emploi |
| <b>FT2 :<br/>Consulter une offre</b>          | Permettre à un étudiant de consulter des offres publiées en lien avec sa formation.   |
| <b>FT3 :<br/>Créer un compte (Prof/Elève)</b> | Deux types de comptes seront disponibles (prof ARE / IMERIR / Elève).   |
| <b>FT4 :<br/>Message Journalier</b>           | Professeurs ARE / IMERIR pourront ajouter des messages qui seront mis en avant au lancement de l'application (Encouragements/ conseils).<br>Message et photo                |
| <b>FT5 :<br/>Espace documentation</b>         | Un espace documentation pour que profs déposent des documents exemples (CV's / LM) et que les élèves peuvent consulter.   |
| <b>FT6 :<br/>Postuler offre</b>               | Permettre à un élève de postuler à une offre en envoyant un mail directement depuis le mobile en joignant une LM et un CV.  |
| <b>FT7 :<br/>Notifications</b>                | « Hey une nouvelle est dispo »  |
| <b>FT8 :<br/>Supprimer une offre</b>          | Une option dans les paramètres de l'offre pour les professeurs pour supprimer une offre d'emploi.   |
| <b>FT9 :<br/>Supprimer plusieurs offret</b>   | Une option dans les paramètres de la liste des offres pour les professeurs pour supprimer en même temps plusieurs offres d'emploi.  |
| <b>FT10 :<br/>Modifier une offre</b>          | Une option dans les paramètres de l'offre pour les professeurs pour modifier une offre d'emploi.  |
| <b>FT11 :</b>                                 | Une option dans le menu pour consulter son profil.  |
| <b>FT12 :</b>                                 | Dans sa page de profil permet d'enregistrer son CV, LM, tableau de bord et l'étiquette.   |
| <b>FT13 :</b>                                 | Les professeur / IMERIR de consulter les profils des étudiants.   |
| <b>FT14 :</b>                                 | Une option dans le menu pour se déconnecter de l'application en toute sécurité.   |
| <b>FT15 :</b>                                 | Dans sa page de profil dans les options permet à l'étudiant de supprimer son compte en intégralité.   |
| <b>FT16 :</b>                                 | Permet à un professeur / IMERIR de supprimer un ou plusieurs comptes en intégralité.  |
| <b>FT17 :</b>                                 | L'étudiants peut modifier quelques éléments de son profil.  |
| <b>FT18 :</b>                                 | Les professeur / IMERIR peuvent suivre les étudiants dans leurs recherches d'entreprises.   |
| <b>FT19 :</b>                                 | Les professeur / IMERIR peuvent filtre les étudiants en fonction de leurs sections.   |
| <b>FT20 :</b>                                 | Les professeur / IMERIR recherche le nom d'un étudiant.   |

# III - Conception préliminaire

## Diagramme de séquences

Les diagrammes de séquence sont la représentation des interactions entre nos acteurs et le système





## Diagramme de déploiement

Voici un diagramme de déploiement en UML, qui représente l'utilisation de l'infrastructure physique par le système.

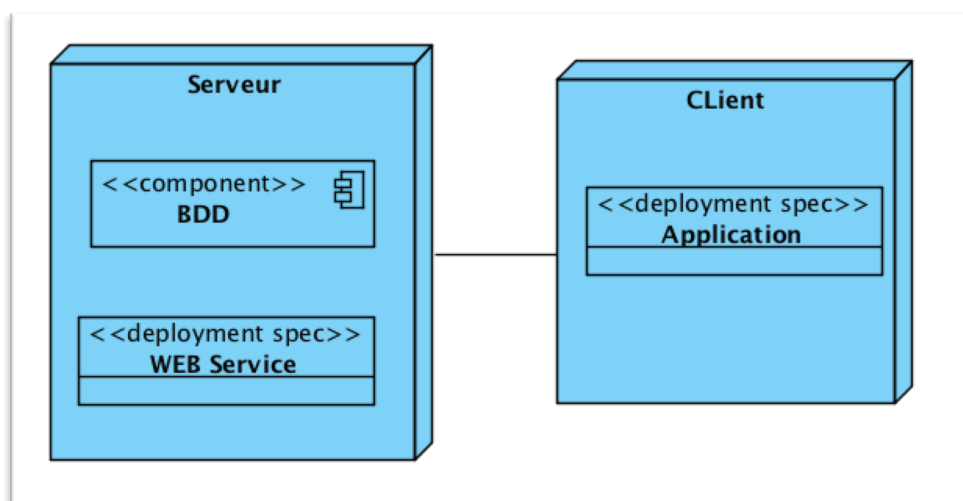
Notre système sera constitué de 2 machines :

Serveur :

Le serveur contient le Services web qui sera traitera avec le client sous le format REST/JSON ainsi que la base de données SQLite.

Le client :

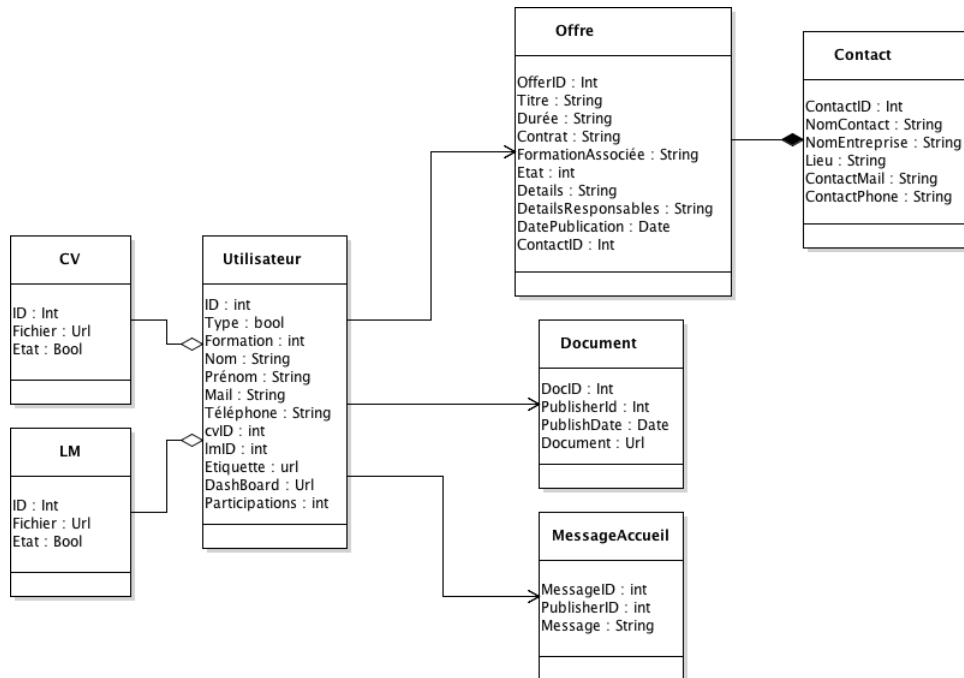
Le client devra au préalable télécharger l'application et avoir accès à une connexion internet pour pouvoir l'utiliser.



## Modèle conceptuel de données

Après avoir analysé les besoins du cahier des charges, nous avons mis en place le schéma relationnel de la base de données

Voici la composition de notre base de données qui est composé de sept tables



CV :

- ID : INTEGER – AUTOINCREMENT
- UrlCV) - STRING
- Etat INTEGER

Contact :

- ContactID : INTEGER – AUTOINCREMENT
- NomContact : STRING
- NomEntreprise : STRING
- Lieu : STRING
- ContactMail : STRING
- ContactPhone : STRING

Document :

- DocID : INTEGER – AUTOINCREMENT
- PublisherId : INT
- PublisherDate : STRING
- Document : STRING

LM :

- ID : INTEGER – AUTOINCRMENT
- UrlLM : STRING
- Etat : BOOL

MessageAccueil :

- MessageID : INTEGER - AUTOINCREMENT
- PublisherId : INT
- Message : TEXT

Offre :

- OfferID : INTEGER – AUTOINCREMENT
- Titre : STRING
- Duree : STRING
- Contrat : INT
- FormationAssociee : INT
- Etat : INT
- Details : STRING
- DetailsReponsable : STRING
- DatePublication : STRING
- ContactRef : INT

Utilisateur :

- id : INTEGER – AUTOINCREMENT
- type : BOOL
- Nom : STRING
- Prenom : STRING
- Mail : STRING
- Telephone : STRING
- Formation : INT
- cvID : INT
- ImID : INT
- Etiquette : STRING
- DashBoard : STRING
- Participations : INT
- Password : INT
- IsValid : BOOL

Table CV et LM : Les deux tables sont identifiées par un identifiant unique, le fichier sous format (URL) et d'un état pour vérifier si le fichier est présent/existant.

Table Utilisateur : Cette table correspond à toutes les personnes qui se sont enregistrées dans l'application, on y trouve un identifiant unique pour chaque utilisateur, un champ Type pour savoir si c'est un étudiant ou un Responsable ARE ainsi que tous les champs personnels de l'utilisateur.

Table Contact : Correspond à l'entreprise qui propose l'offre comme une carte de visite

Table Document : Cette table correspond au document disponible dans la page « Documents » de l'application.

Table Message d'accueil : Cette table correspond au message d'accueil qui est affiché une fois connecté à l'application.

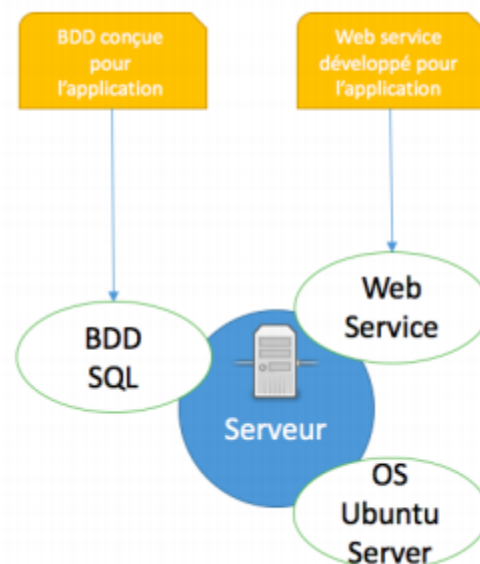
Table Offre : Cette table correspond à toutes les offres disponibles en détails dans la page « Offres » de l'application.



# IV - Architecture technique

Le projet demandant une interaction avec une base de données :

- Une partie serveur contenant une BDD
- Un web service qui commandera les échanges avec le client.



## Web Service

Un Web Service (ou service Web) est une application accessible via un site Web, application qui fournit des données à d'autres systèmes potentiellement hétérogènes comme par exemple d'autres sites web ou des appareils mobiles notamment.

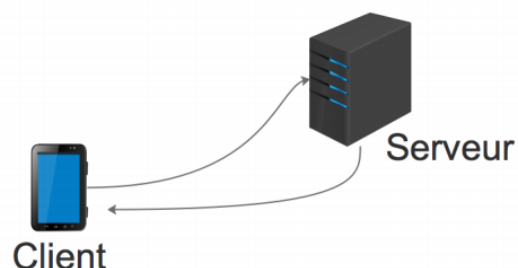
Nous choisirons de faire ce Web Services avec le langage de programmation en Python avec le micro Framework « FLASK » (<http://flask.pocoo.org>, <http://flask.pocoo.org/docs/0.11/quickstart/>, <https://www.tutorialspoint.com/flask/index.htm>), qui a l'avantage d'être très simple à prendre en main.

## Base de données

Nous utiliserons la librairie SQLite pour la création de la base de données, qui permet grâce au langage SQL de récupérer ou de scotcher des données de façons structurée. SQLite est rapide, flexible et léger.

Pour qu'un client puisse échanger avec un serveur, les données envoyées au Web Services seront au format « JSON » (Java Script Object Notation), qui repose sur une notation simple, compacte, facile à lire.

NOUS NOUS SOMMES RENDUS COMPTE QUE SQLITE DANS UN VIRTUAL ENVIRONNEMENT (Ce comment fonctionnent les serveurs distants) est instable pour l'écriture du coup on va tout modifier pour passer la base de données sur un SGBDR (MySQL).



## Le protocole HTTP

### La requête

Quand le client « demande une action », il envoie au serveur une requête.

Nous allons détailler nos requêtes par rapport au projet.

Qui regrouperont plusieurs éléments :

- Le chemin
- La méthode pour faire appel à chaque élément
- Le type de requête (GET, POST, PUT, DELETE)

|                         | Chemin                         | Méthode               | Requête | Commentaire   |
|-------------------------|--------------------------------|-----------------------|---------|---|
| Connexion               | /mobile/users/{mail}           | GetUserByEmail        | GET     | Récupère les informations d'un utilisateur grâce à son adresse mail   |
| Inscription             | /mobile/users/                 | AddNewUser            | PUT     | Poste toutes les informations entrées dans les formulaires sur la base de données. Ajoute un utilisateur.     |
| PAGE PRINCIPALE         | /mobile/offers/                | GetOffers             | GET     | Récupère les offres d'emploi disponibles sur la BDD   |
| DERNIÈRES OFFRES        | /mobile/welcomeMessages/       | GetLastWelcomeMessage | GET     | Récupère le dernier message d'accueil disponible  |
| MESSAGE D'ACCUEIL       | /mobile/contact/{id}           | GetContactById        | GET     | Récupère les informations du contact associé à une offre  |
| Consulter une offre     | /mobile/offers/{offerID}       | GetOfferById          | GET     | Récupère toutes les données de l'offre d'emploi sélectionnée sur la page principale                           |
|                         | /mobile/contact/{id}           | GetContactById        | GET     | Récupère les données d'un contact associé à l'offre d'emploi sélectionnée sur la page principale              |
| Supprimer une offre     | /mobile/offers/{offerID}       | DeleteOfferById       | DELETE  | Supprime l'offre d'emploi en utilisant de son ID  |
| Modifier une offre      | /mobile/offers/{offerID}       | ModifyOfferById       | POST    | Modifie l'offre d'emploi en utilisant de son ID   |
| Consulter les documents | /mobile/documents/             | GetDocuments          | GET     | Récupère toutes informations disponibles sur les documents disponibles dans la base de données                |
|                         | /mobile/users/{id}             | GetUserById           | GET     | Récupère les données d'un utilisateur grâce à son id affiché sur les infos du document pour afficher son nom. |
| Consulter un document   | /mobile/documents/{id}         | GetDocumentById       | GET     | Récupère les données d'un document grâce à son ID   |
| Supprimer un document   | /mobile/documents/{documentID} | DeleteDocumentById    | DELETE  | Supprime le document en utilisant de son ID   |

|   | Chemin                   | Méthode              | Requête | Commentaire   |
|---|--------------------------|----------------------|---------|---|
| CONSULTE<br>R LES<br>PROFILS<br>ÉTUDIANTS | /mobile/users/{type}     | GetUsersByType       | GET     | Retrouve tous les utilisateurs d'un type (Etudiant ou responsable ARE)  |
|   | /mobile/CV/{id}          | GetCVByID            | GET     | Retrouve les informations d'un CV grâce à son ID                        |
|   | /mobile/LM/{id}          | GetLMByID            | GET     | Retrouve les informations d'une LM grâce à son ID                       |
|   |                          |                      |         |   |
| CONSULTE<br>R UN<br>PROFIL<br>ÉTUDIANT    | /mobile/users/{id}       | GetUserByID          | GET     | Retrouve un utilisateur grâce à son ID                                  |
|   | /mobile/CV/{id}          | GetCVByID            | GET     | Retrouve les informations d'un CV grâce à son ID                        |
|   | /mobile/LM/{id}          | GetLMByID            | GET     | Retrouve les informations d'une LM grâce à son ID                       |
|   | /mobile/CV/              | ChangeCVStatus       | POST    | Modifie le statut d'un CV si modifié par le responsableARE              |
|   | /mobile/LM/              | ChangeLMStatus       | POST    | Modifie le statut d'une LM si modifié par le responsableARE             |
| AJOUTER<br>UNE OFFRE                      | /mobile/offers/          | AddNewOffer          | PUT     | Ajoute une nouvelle offre d'emploi avec les données complétées          |
|   | /mobile/users/{id}       | GetUserByID          | GET     | Récupère l'ID de l'USER pour l'insérer dans le PUT de l'offre           |
| AJOUTER<br>UN<br>DOCUMENT                 | /mobile/documents/       | AddNewDocument       | PUT     | Ajoute un nouveau document  |
|   | /mobile/users/{id}       | GetUserByID          | GET     | Récupère l'ID de l'USER pour l'insérer dans le PUT du document          |
| AJOUTER<br>UN<br>MESSAGE<br>D'ACCUEIL     | /mobile/welcomeMessages/ | AddNewWelcomeMessage | PUT     | Ajoute un nouveau message d'accueil                                     |
|   | /mobile/users/{id}       | GetUserByID          | GET     | Récupère l'ID de l'USER pour l'insérer dans le PUT du message d'accueil |

## La réponse

L'erreur 500 désigne une erreur survenue sur le serveur.

L'erreur 401 désigne que l'accès à cette page web n'est pas autorisé

L'erreur 404 signifie que la page demandée ne correspond à aucune route existante.

L'erreur 400 signifie que la requête du client est mal formée (par exemple s'il n'envoie pas toutes les données d'un formulaire)

L'erreur 200 signifie que tout s'est bien passé

# V – Règles de codage & Convention de nommage

---

## 1. Règles de codage

Les règles de codage sont un ensemble de règles à suivre pour uniformiser les pratiques de développement logiciel, diffuser les bonnes pratiques de développement et éviter les erreurs de développement "classiques" au sein d'un groupe de développeurs.

- ❖ Le nommage et l'organisation des fichiers du code source
- ❖ Le style d'indentation
- ❖ Les conventions de nommage, ou règles de nommage
- ❖ Les commentaires et documentation du code source
- ❖ Recommandations sur la déclaration des variables
- ❖ Recommandations sur l'écriture des instructions, des structures de contrôle et l'usage des parenthèses dans les expressions.

## 2. Convention de nommage

Une convention de nommage dans la programmation informatique est un ensemble de règles destinées à choisir les identifiants logiciels (noms des éléments du programme) dans le code source et la documentation.

- ❖ Nommage des programmes :
- ❖ Nommage des fichiers :
- ❖ Gestion des versions :

# VI – Programme de test

---

Utilisation de la plateforme Git Hub une fois les tests validés.

Lien du projet :

<https://github.com/mbouillon/ApplicationARE>

## Les tests :

Tests par modules/écrans.

### I. LoginActivity

Ouverture de l'application sur l'écran login activity pour une première utilisation

Tests A réaliser :

Tentative de connexion avec bon login mot de passe -> Envoi sur la Main Activity

Tentative de connexion avec mauvais identifiants -> Affiche un message d'erreur

Tentative de connexion avec un compte non validé -> Propose le renvoi d'un email de validation

Click du bouton Inscription Etudiant IMERIR -> Envoi sur l'activité Inscription Etudiant

Click du bouton Inscription Responsable ARE -> Envoi sur l'activité Inscription Responsable ARE

Sélection de la check box se souvenir – se connecter et se déconnecter -> L'adresse email est pré remplie sur le champ

Ne pas sélectionner la check box se souvenir – se connecter et se déconnecter -> l'adresse email n'est pas pré remplie.

### II. RegisterResponsableActivity

Remplir les champs et s'inscrire -> Envoi sur la login activity et réception d'un mail de confirmation de compte sur l'adresse mail renseignée

Ne pas remplir un champ -> Erreur il faut remplir tous les champs

Adresse mail et confirmation mail non égales -> Message d'erreur adresse mail doivent être identiques.

Mot de passe et verif mot de passe non identiques -> Message d'erreur mot de passes doivent être identiques.

Adresse email non imerir -> Message d'erreur l'adresse email doit être imerir.com

Mot de passe d'accès non correspondant affichage du message d'erreur le mot de passe est incorrect

### III. RegisterStudentActivity

Remplir les champs et s'inscrire -> Envoi sur la login activity et réception d'un mail de confirmation de compte sur l'adresse mail renseignée

Ne pas remplir un champ -> Erreur il faut remplir tous les champs

Adresse mail et confirmation mail non égales -> Message d'erreur adresse mail doivent être identiques.

Mot de passe et verif mot de passe non identiques -> Message d'erreur mot de passes doivent être identiques.

### IV. MainActivity

Ouverture sur offersListFragment, possibilité de changer de fragments grâce à la Bottom bar -> Navigation entre offersList DocumentList et Profil

Click sur le menu à droite de la toolBar -> Affiche les options Déconnexion et Quitter l'application

**Conception Général**

Click sur déconnexion -> Déconnecte de l'app en renvoyant l'utilisateur sur la login activity si l'utilisateur rouvre l'app il reste sur la LoginActivity.  
Click sur Quitter l'application -> Quitte simplement l'app si l'utilisateur la rouvre il tombe sur la MainActivity.

#### V. MainActivity.OffersListFragment

Connecté avec un compte Responsable :

- Liste de toutes les offres d'emploi disponibles dans la BDD visible
- Dernier message d'accueil avec le nom de l'auteur visible
- Bouton ajouter offre et message disponible
- Click sur le message d'accueil -> Envoi vers la messages list activity
- Click sur une offre -> Envoi vers la OfferDetailActivity
- Click sur bouton ajouter offre -> Envoi vers la AddOfferActivity
- Click sur bouton ajouter message -> Envoi vers la AddMessageActivity
- Long click sur une offre -> Affiche une popup proposant de la supprimer ou de l'éditer.
- Click sur supprimer -> Supprime l'offre en question
- Click sur editer -> Envoi vers EditOfferActivity en remplissant les champs avec les données de l'offre
- Swipe du haut vers le bas -> Recharge les données offres messages et utilisateurs

Connecté avec un compte Etudiant CDSM

- Les offres destinées à tout le monde ou uniquement aux CDSM sont visibles
- Dernier message d'accueil avec le nom de l'auteur visible
- Bouton ajouter offre et message non disponible
- Click sur le message d'accueil -> Envoi vers la messages list activity
- Click sur une offre -> Envoi vers la OfferDetailActivity
- Long click sur une offre -> ne fait rien

Connecté avec un compte Etudiant CDPIR

- Les offres destinées à tout le monde ou uniquement aux CDPIR sont visibles
- Dernier message d'accueil avec le nom de l'auteur visible
- Bouton ajouter offre et message non disponible
- Click sur le message d'accueil -> Envoi vers la messages list activity
- Click sur une offre -> Envoi vers la OfferDetailActivity
- Long click sur une offre -> ne fait rien

Connecté avec un compte Etudiant UPVD

- Les offres destinées à tout le monde ou uniquement aux UPVD sont visibles
- Dernier message d'accueil avec le nom de l'auteur visible
- Bouton ajouter offre et message non disponible
- Click sur le message d'accueil -> Envoi vers la messages list activity
- Click sur une offre -> Envoi vers la OfferDetailActivity
- Long click sur une offre -> ne fait rien

#### VI. MainActivity.DocumentListFragment

Connecté avec un compte responsable

- Visibilité de la liste de tous les documents
- Visibilité du bouton ajouter document
- Click long sur un document affiche une popup proposant la suppression
- Click sur le bouton supprimer -> Supprime le document
- Click sur document ouvre la PDF Activity qui affiche le document dans une WebView

Connecté avec un compte étudiant

- Visibilité de la liste de tous les documents

- Long click ne fait rien

- Le bouton ajouter document n'est pas visible

- Click sur document ouvre la PDF Activity qui affiche le document dans une WebWiew

#### VII. MainActivity.ProfilFragment

Connecté avec un compte responsable

- Liste de tous les étudiants triables par nom et/ou par formation

- Click sur étudiant ouvre l'activity profil et affiche tous les détails de l'étudiant

Connecté avec le compte étudiant

- Affiche le profil de l'étudiant

- Possibilité d'ajouter un CV une lettre de motivation, une étiquette et un tableau de bord (cf cours ARE)

#### VIII. AddMessageActivity

2 entités, un edit text et un bouton ajouter message, 255 caractères possibles

Si les 255 caractères dépassés alors impossible d'écrire plus.

Bouton ajouter message envoie le message a la BDD et renvoie vers la mainActivity ou l'on voit le nouveau message ajouté

Si aucun texte dans le champ -> Affiche erreur champ vide

#### IX. AddOfferActivity

Aucun champ rempli -> Message d'erreur aucun champ rempli

Tous les champs OK -> Ajoute l'offre et renvoie vers la main activity ou l'offre est visible

Formation non renseignée, alerte l'offre sera visible par tous les étudiants voulez-vous continuer ?

Type contrat non renseigné -> Erreur veuillez renseigner le type du contrat si non connu veuillez renseigner non connu

#### X. ModifyOfferActivity

Les champs doivent être remplis avec les données de l'offre à modifier

Aucun champ rempli -> Message d'erreur aucun champ rempli

Tous les champs OK -> Modifie l'offre et renvoie vers la main activity ou l'offre est visible

Formation non renseignée, alerte l'offre sera visible par tous les étudiants voulez-vous continuer ?

Type contrat non renseigné -> Erreur veuillez renseigner le type du contrat si non connu veuillez renseigner type non connu

#### XI. MessagesListActivity

Connecté avec un compte responsable

- Visibilité de la liste de tous les messages

- Long click sur un message -> Alerte voulez-vous supprimer ce message ?

- Click sur supprimer -> Supprime le message de la BDD

Connecté avec le compte Etudiant

- Visibilité de la liste de tous les messages

- Long click sur un message -> Rien ne se passe