Université Mohammed Premier

Ecole Nationale des Sciences Appliquées d'Oujda

Génie Informatique

Rapport de stage d'application

Sujet : Etude, Conception et Développement d'une Application Android pour les petites annonces gratuites avec l'API Avito

Réalisé par :

Soumia Youbi

Encadré par :

M. Yasser Bouhamria

Remerciement

Au terme de ce travail, je tiens à exprimer ma profonde gratitude et mes sincères remerciements à mon encadrant M. Yasser Bouhamria pour tout le temps qu'il m'a consacré, ses directives précieuses, et pour la qualité de son suivi durant toute la période de ce stage.

Je tiens aussi à remercier vivement tous les membres de l'équipe Dream à qui je faisais partie durant ce stage, et à toute l'équipe d'AVITO pour leur accueil, le temps passé ensemble et le partage de leur expertise au quotidien.

Mes plus vifs remerciements s'adressent aussi à tout le cadre professoral et administratif de l'École Nationale des Sciences Appliquées d'Oujda.

Enfin, je remercie toute personne qui a contribué de près ou de loin à l'élaboration de ce travail.

Résumé

Ce document a pour but de présenter l'ensemble des travaux effectués dans le cadre d'un stage d'application. L'objectif du projet est de réaliser une application Android de petites annonces que tous les marocains peuvent utiliser pour acheter et vendre sur internet, en utilisant l'API Avito.

Ce projet a été divisé en trois parties : La première partie consiste à se documenter sur l'architecture de la plateforme d'Avito et à se familiariser avec son API ainsi qu'avec les technologies qui devront être utilisées pour réaliser ce projet. La deuxième concerne la réalisation de la nouvelle application mobile qui liste les annonces. Et finalement la dernière étape a été conçue pour l'insertion d'une annonce depuis notre application dans la plate-forme.

Abstract

This report is an outcome of the intership program that I have involved for two months in Avito. The main subject of this internship is creating an Android application for ads that helps Moroccans selling and buying on the net, all using Avito's existing API.

The project was divided into 3 parts: the first part concerned getting familiar with Avito's platform architecture, its API and the different technologies that must be used during the project. The second part was devoted to the conception and realization of the new Android app of ads. Finally, the last part is where we added the ad insert functionality to our app.

Table des matières

Introduction générale

Dans le cadre de ma formation de 2eme année du cycle d'ingénieur, filière génie informatique à l'école nationale des sciences appliquées d'Oujda, j'ai effectué un stage d'application au sein de la filiale Avito appartenant au Groupe Media Schibsted (Située à Casablanca). Au cours de ce stage dans le département de développement, j'ai pu mettre en pratique mes connaissances et mes compétences professionnelles.

Mon maitre de stage étant Yasser Bouhamria, j'ai pu apprendre dans d'excellentes conditions et ai bénéficié d'un soutien quotidien de qualité.

Mon stage au département de développement a consisté essentiellement à la réalisation d'une application Android de petites annonces gratuites en utilisant l'API déjà existante d'Avito.

Plus largement, ce stage m'a permis d'enrichir mes connaissances en développement mobile, et m'a donné l'opportunité de découvrir comment un organisme dans le secteur de l'e-commerce se développe, ses défis et son évolution au cours du temps.

I. Chapitre 1 : Présentation du groupe hôte Schibsted et sa filiale Avito.ma

Introduction

Pour mieux cerner le contexte du projet, nous nous intéressons à l'organisme Avito et le Groupe Media Schibsted (SCM) auquel il appartient.

1. Présentation du groupe Schibsted

Le Groupe Media Schibsted (SCM) est un conglomérat norvégien du secteur des médias avec une activité dans 30 pays, notamment en Norvège et en suède, et comptant plus de 7500 employées. Il est également propriétaire de 36 applications web commerciales dans 13 pays européens, deux pays asiatiques, cinq pays latino-américains pays et deux pays africains dont le Maroc avec Avito.ma.

Le Groupe Media Schibsted est organisé en :

- Schibsted Media Houses : Maisons de presse opérant au niveau international.
- Schibsted Marketplaces : dédié aux annonces en ligne, notant que le groupe est n ° 1 des petites annonces dans 18 pays, dont la France, la Suède et notamment Maroc.

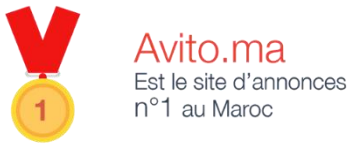
Schibsted Growth investit dans des entreprises numériques ambitieuses avec des modèles économiques évolutifs et innovants.

2. Présentation de la filiale Avito

En juin 2011, Schibsted Classified Media a lancée Bikhir.ma, qui est vite reconnu comme l'équivalent du site d'annonces français leboncoin.fr (filiale également de Schibsted).

Le 23 juillet 2014, Avito et Bikhir annoncent un contrat définitif de fusion. C'est cette joint-venture qui marque l'avènement du leader du marche d'annonces en ligne qu'est aujourd'hui Avito.ma. Cette fusion a donné lieu à une nouvelle identité visuelle de la marque et le nom retenu est Avito. Ceci est dû à la forte notoriété d'Avito (déjà en 2014, 95% des marocains connaissent avito.ma).

Aujourd'hui, Avito.ma est le premier site d'annonce au Maroc, avec plus de 6 millions de visiteurs uniques par mois et plus 400 millions de page vues par mois.



La mission revendiquée par Avito est la démocratisation de l'achat, la vente sur internet et l'amélioration du quotidien des internautes marocains. Effectivement, le marocain s'intéresse de plus en plus à l'achat et vente sur Internet ;

Après Hespress.com, Avito.ma est le deuxième portail web national le plus visité par les marocains. L'expertise accumulée en termes d'acquisition, de rétention et de monétisation de plateforme web est indéniable. 95% des marocains reconnaissent Avito et de plus en plus voit en la plateforme leur moyen de générer des revenus devenus indispensables.

Conclusion

Dans ce premier chapitre, nous avons commencé par une présentation générale du contexte du projet en précisant l'organisme d'accueil. Dans le prochain chapitre, nous allons présenter notre projet, sa problématique et son objectif.

II. Chapitre 2 : Présentation du sujet de stage

Introduction

L'objectif de ce chapitre est de situer le projet dans son contexte général, à savoir la problématique qui à pousser à le réaliser et son objectif principal.

1. Problématique

La révolution des Smartphones a contribué fortement à l'arrivée des applications mobiles qui sont actuellement en pleine croissance, et développement des usages dans presque tous les domaines.

Il existe plusieurs catégories d'applications mobiles destinées à des segments de marchés différents, tel que les applications de petites annonces. Ces applications mettent en relation des particuliers et/ou professionnels qui diffusent une annonce, composée de texte et de photos, sur le site internet en question, pour proposer une vente, une location ou un service en échange d'une rémunération. Dans cette perspective, Avito, filiale du Groupe Media Schibsted, étant le site N°1 d'annonces gratuites au Maroc qui fournit un service permettant la vente et l'achat entre deux personnes en toute simplicité, sans procédure compliquée. Avito possède de plus une application Android reposant sur le même principe que le site et utilise une API REST pour le transfert de données, et pour se faire elle adopte la bibliothèque Volley comme bibliothèque qui offre d'excellentes fonctionnalités et réalise de plusieurs requêtes en même temps. L'objectif de mon projet est de réaliser la même application Android mais avec la nouvelle bibliothèque Retrofit qui est plus propre, simple et légère pour Android comme elle est assez puissante et plus rapide que Volley.

2. Objectif du projet

Le projet a pour objectif principal de réaliser une application Android pour les petites annonces avec l'API d'Avito, en utilisant la librairie Retrofit au lieu de Volley pour réaliser des appels web services REST.

Conclusion

Dans ce chapitre, nous avons mis notre projet dans son cadre général, à savoir la problématique du projet et ses objectifs, et nous allons spécifier dans le prochain chapitre les différents besoins auxquels doit répondre notre application

Chapitre 2 : Spécifications des besoins

Introduction

La phase de spécification des besoins est primordiale pour comprendre le contexte du système, elle permet de définir les besoins à différents niveaux d'abstractions.

Les fonctionnalités de l'application mobile devront être décrites sans ambiguïtés, ensuite analysées à travers l'introduction des acteurs et les diagrammes de cas d'utilisation relatifs à ces acteurs.

1. Besoins fonctionnels

Selon une enquête de Clutch réalisée en 2017, la principale raison pour laquelle les consommateurs utilisent une application mobile est la présentation du contenu d'une manière plus rapide, attrayante et ergonomique. Pour ceci, notre application doit satisfaire ces exigences et faciliter le contact entre l'acheteur et l'annonceur.

Nous présentons dans ce qui suit tous les besoins fonctionnels de l'utilisateur :

- Pouvoir visualiser toutes les annonces se trouvant au Maroc
- Consulter une annonce pour plus de détails.
- Contacter l'annonceur facilement en l'appelant ou en lui envoyant un message SMS.
- Créer un compte sur l'application et modifier ses informations.
- S'authentifier et se déconnecter de l'application.
- Enregistrer une annonce dans une liste d'annonces favorites.
- Recevoir en email des offres exclusives
- Insérer rapidement une annonce
- Ajouter jusqu'à six photos pour une annonce.

2. Besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement.

Les besoins non fonctionnels se résument en trois actions majeures :

- Ergonomie et souplesse : L'application mobile doivent offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles à l'écran du support tenu.

- Efficacité **:** L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur.
- Lisibilité : Le code doit être clair pour faciliter les futures améliorations.

Conclusion

Ce chapitre nous a permis de couvrir les différents besoins fonctionnels et non fonctionnels auxquels doit satisfaire notre application. Nous avons aussi détaillé ces besoins à travers des diagrammes de cas d'utilisation pour passer par la suite à la conception de notre application qui sera présentée dans le chapitre suivant.

III. Chapitre 3 : Architecture et conception

Introduction

Après avoir achevé la phase d'analyse et spécifications des besoins, nous entamons maintenant la phase de conception. Cette étape s'avère primordiale pour le déroulement du projet et permet de préparer le terrain pour l'étape de réalisation.

Pour ce faire, nous allons présenter une conception détaillé contenant des diagrammes de classes et de cas d'utilisation pour décrire le comportement du système et faire une représentation abstraite de ses objets. Ensuite nous déterminons la méthodologie suivie pour la réalisation, les tâches de construction et les technologies choisies pour réaliser cette application.

1. Architecture de la plateforme

Le diagramme ci-dessous représente l'architecture de la plateforme utilisée par Avito, détaillant ainsi le fonctionnement de l'application.

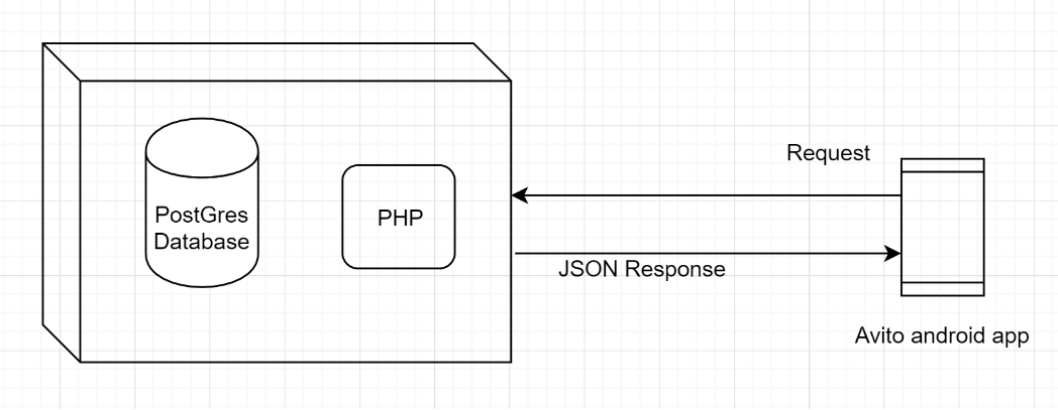


Figure 1: Architecture globale de la plateforme

L'application utilise le protocole http pour envoyer une requête vers un endpoint de l'API REST, qui la traite et renvois une réponse contenant le résultat en JSON.

2. Conception de l'application

i. Conception générale

Pour notre application mobile, nous avons choisi de travailler avec le Modèle-Vue-Contrôleur. C'est une architecture et une méthode de conception qui intervient dans la réalisation d'une application mobile ou web dont l'intérêt principal est la séparation des données, de l'affichage et des actions.

Nous avons choisi de travailler avec l'architecture MVC, car elle permet de bien séparer la logique de la présentation. La vue n'aura aucune logique

d'imbriquer. Aussi, étant donné que tout est très bien séparé, il est très facile d'ajouter et de modifier au code sans que le reste ne s'effondre.

Le schéma suivant résume la structure générique d'une architecture MVC :

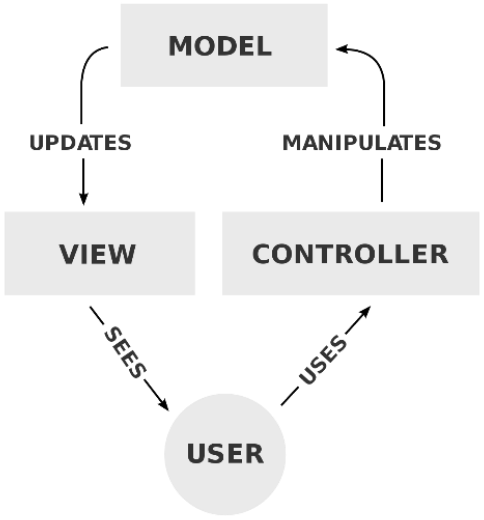


Figure 2 : Architecture du modèle-Vue-Contrôleur

- Le Modèle : représente le comportement de l'application.
- La Vue : correspond à l'interface avec laquelle l'utilisateur interagit.
- Le Contrôleur : prend en charge la gestion des évènements de synchronisation pour mettre à jour la vue ou le modèle.

ii. Conception détaillée

Diagramme de cas d'utilisation

Le Diagramme de cas d'utilisation représenté ci-dessous permet d'identifier les fonctionnalités que doit fournir le système et les possibilités d'interaction entre ce dernier et les acteurs.

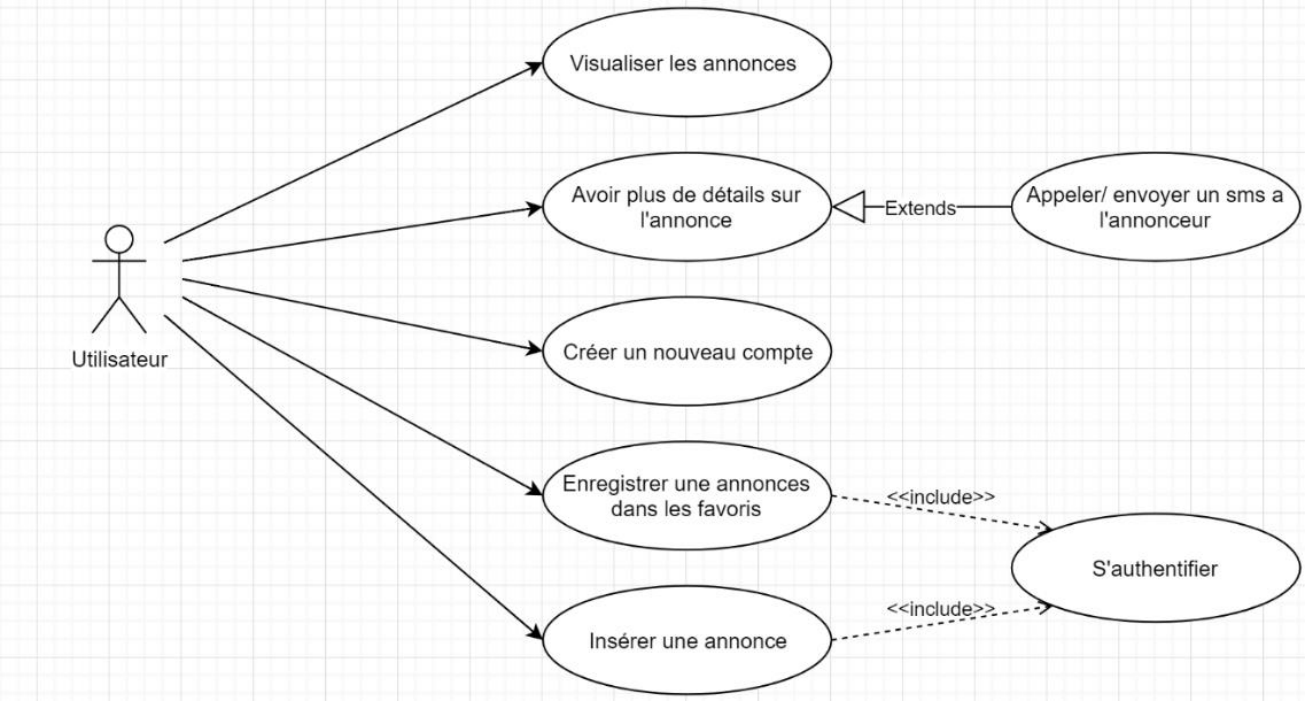


Figure 3 : Diagramme de cas d'utilisation

Diagramme de classe

Le diagramme de classes est considéré comme le plus important de la modélisation orienté objet. Il s'agit d'une vue statique du fait qu'on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classe permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier. Le diagramme de classe retenus à la fin de la conception est le suivant :

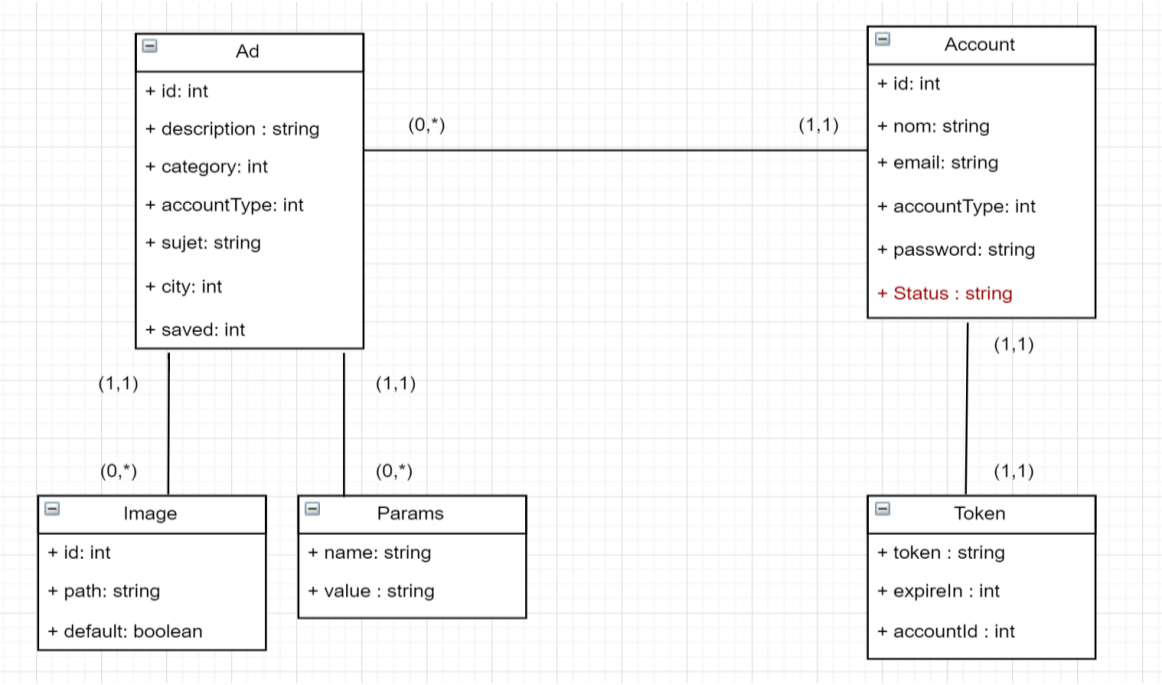


Figure 4 : Diagramme de classes

Conclusion

Dans ce chapitre, nous avons modélisé le fonctionnement de l'application afin d'avoir une vue globale et simplifiée du système. Nous avons aussi détaillé les différents modules de l'application ce qui nous a permis d'organiser le travail et d'avoir une idée claire sur le travail à réaliser. Ce travail est décrit plus précisément dans le chapitre qui suit.

IV. Chapitre 4 : Réalisation

Introduction

Cette partie est consacrée à présenter la méthodologie suivie pour la réalisation du projet, les outils et technologies utilisées, ainsi que quelques interfaces de l'application réalisé.

1. Méthodologie de travail

Dans cette partie, nous allons présenter le processus et la méthodologie suivie pour la réalisation de notre projet.

Le processus suivi est itératif incrémental. Ce dernier consiste de partir d'une idée, d'abord précisée itérativement, ensuite partiellement développée en un incrément fonctionnel, potentiellement livrable tel quel. Chaque incrément suivant étant susceptible d'affiner les fonctionnalités existantes, est également itératif.

Pour mieux gérer l'avancement du projet, nous avons adopté la méthode agile KANBAN. L'outil principal de cette méthode est un tableau blanc divisée en plusieurs colonnes, allant de la liste des tâches à effectuer à la liste des tâches terminées. Chaque tâche est modélisée par un post-it. Au fur et à mesure de l'avancement d'une tâche, son responsable la fait progresser de colonne en colonne pour marquer son état. Pour ceci, nous avons utilisé un outil de gestion de projet qui inspire par cette méthode : l'outil Trello

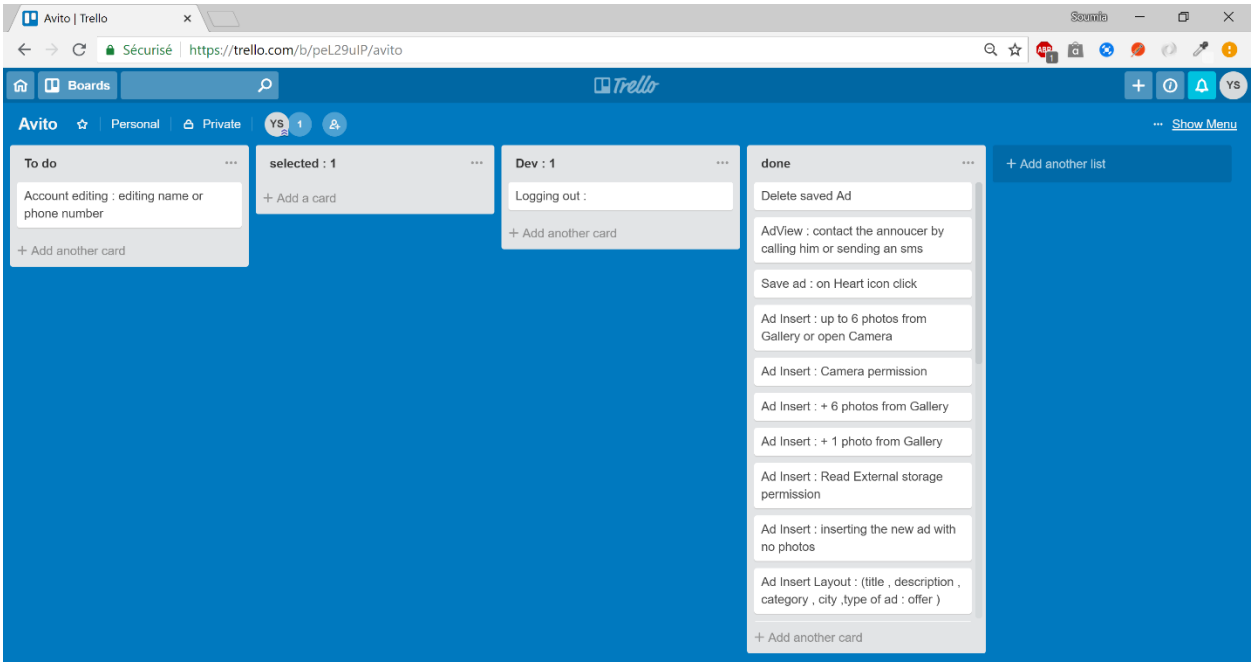


Figure 5 : la gestion du projet sur Trello

Les itérations fixées pour notre projet sont les suivantes :

Itération 1 :

Cette première partie a pour objectif d'étudier l'existant et la conception globale du système d'Avito. L'étude de l'existant consiste à connaitre tous les éléments du système, à savoir le côté fonctionnel de la plate-forme, son architecture ainsi que les technologies utilisées pour la mise en œuvre de ces éléments. Pour se faire, nous avons pris le temps de se documenter, d'étudier l'architecture du projet ainsi de fixer les technologies et outils pour réaliser le nôtre. Durant cette itération, nous avons aussi effectué des premiers essaies d'implémentation, ce qui nous a permis d'avoir une idée plus précise des diffèrent composant d'Avito et de la communication entre ses éléments. Une documentation est faite sur des nouvelles librairies et qui seront utilisé pour ce projet, parmi lesquelles on cite la librairie Retrofit qui réalisera les appels web services REST, et la librairie Fresco qui fera le chargement et l'affichage d'images l'application.

Itération 2 :

Cette itération a été divisé en deux parties.

Dans la première partie, nous avons réalisé les taches suivantes :

- Affichage de la liste infinie d'annonces avec possibilité de la rafraichir
- Consulter une annonce pour plus d'information sur celles-ci.
- Création d'un compte sur l'application
- Authentification et Déconnection
- Ajout d'une annonce dans une liste des favoris

Ces taches ont été réalisées après avoir charger les données depuis l'API. Ces données sont récupérées de type JSON et transformées ensuite en objet java POJO (Plain Old Java Object).

La deuxième partie de cette itération a été consacrée au design des interfaces graphiques, en les rendant plus ergonomique et convivial pour les utilisateurs.

Itération 3 :

Dans la troisième itération, nous avons rajouter une nouvelle fonctionnalité à notre projet qui est l'insertion des annonces. Avant d'entamer la partie du code, nous avons étudier à nouveau les éléments nécessaires pour insérer dans chaque catégorie une annonce, et vue la complexité et la diversité des composants de la plate-forme Avito, nous avons décidé d'éliminer les catégories Immobilier et Véhicule. L'utilisateur, authentifiée sur l'application, peut insérer une annonce avec plusieurs photos sélectionnées depuis sa galerie ou capturées depuis sa caméra, un titre, une description, un prix et une ville.

2. Outils et technologies utilisées

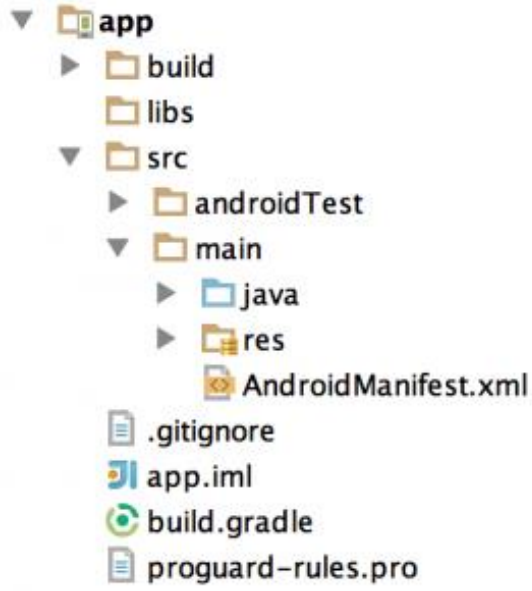
Cette partie est réservée aux outils et langages de développement utilisées durant la réalisation de l'application. Le choix de ces outils répond aux exigences techniques.

Android Studio

Tout d'abord au niveau des outils, une application est développée à l'aide du SDK Android et de l'IDE Android studio. Elle est codée à l'aide de deux langages, l'interface se développe en XML et le reste de l'application en JAVA. Aucun OS spécifique requit. L'IDE Android Studio fonctionne sur tous les OS qui supportent une machine virtuel java.

Les projets ont changé d'architecture depuis le passage à Android Studio, ou l'on utiliserait plusieurs projets sous Eclipse, avec des dépendances entre

chacun, nous parlerons maintenant d'un seul Projet, contenant plusieurs modules. Ces modules ont la forme suivante :



Nous y trouvons le dossier **res/** et le fichier **AndroidManifest.xml**, les sources (fichier. Java) ont maintenant été déplacées dans un dossier **java/**.

Le fichier **build.gradle** sert de configuration pour le nouveau moteur de production nommé **Gradle**, qui sera utilisé pour construire notre application afin de la déployer sur notre smartphone ou sur le Play Store.

REST API

Le protocole REST (**RE**presentational **S**tate **T**ransfer) constitue un style architectural et un mode de communication fréquemment utilisé dans le développement de services Web. Comme il est très souple car il ne consomme pas autant de bande passante, ce qui rend son utilisation plus pratique sur Internet, et il est indépendant vis-à-vis de la plateforme sur laquelle ils sont déployés.

Pour ce projet, nous avons opté pour **Retrofit 2**, qui est un client REST développé par le groupe Square. Cette librairie est basée elle-même sur le

client REST OKHttp et a pour objectif de simplifier la gestion des appels réseaux avec une interface, une instance de cette interface appelée service et un appel de type Call.

Nous définissons dans l'interface ce que nous appelons des endpoints. Pour une url de base est spécifié dans le service et une url relatif à chaque endpoint. De plus, pour chaque endpoint nous spécifions le type de requête il effectue en utilisant l'une des annotations suivantes : @GET, @PUT, @UPDATE, @DELETE, @HEAD, @PATCH.

Les requêtes de type POST et PUT doivent contenir un corps qui spécifie l'entité à traiter, ainsi ils utilisent la balise @Body en paramètre.

D'autres annotations importantes existent comme :

@Path : elle permet d'avoir des Url dynamiques qui seront résolues lors de l'exécution.

@Header : elle permet de rajouter des paramètres dans le header de la requête.

Ci-dessous figure l'endpoint pour récupérer les annonces depuis l'api :

```
33
34  public interface GetDataService {
35      @Headers({
36              "Accept: application/json",
37              "content-type: application/json",
38              "lang: fr"
39      })
40
41      // Get Ads
42      @GET("api/v1/ads")
43      Call<AdList> getAllAds();
```

Postman

Postman est un outil pratique pour tester rapidement une Api. Il permet de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en **Collections**. Cette classification permet notamment de regrouper des

requêtes de façon « fonctionnelle » (par exemple enchaînement d'ajout d'item au panier, ou bien un processus d'identification).

Postman assure également la gestion des **Environnements**, qui permet de contextualiser des variables et d'exécuter des requêtes ou des séries de requêtes dans différents configurations (typiquement : dev, recette, prod).

Fresco

C'est un système de chargement et d'affichage d'images pour les applications Android. Il charge les images à partir du réseau ou du stockage local, et affiche un placeholder jusqu'à ce que l'image soit chargée. Fresco dispose de deux niveaux de cache ; un en mémoire et un autre en stockage interne.

Fresco stocke les images dans un emplacement spécifique de la mémoire Android, ce qui permet à l'application de s'exécuter plus rapidement et réduit le risque de tomber sur l'erreur « OutOfMemoryError ».

Fresco soutient également:

- Streaming des fichiers progressifs JPEGs
- Affichage des GIFs et WebP animés
- Personnalisation étendue du chargement et de l'affichage des images.

GitHub

GitHub est le plus grand espace de stockage de travaux collaboratifs au monde. Il est centré vers l'aspect social du développement. En plus d'offrir l'hébergement de projets avec Git, le site offre de nombreuses fonctionnalités habituellement retrouvées sur les réseaux sociaux comme les flux, la possibilité de suivre des personnes ou des projets ainsi que des graphes de réseaux pour les dépôts.

2. Taches de réalisations

Cette partie contiendra les taches réalisées au niveau de l'application de notre projet.

- La liste des annonces :

Une fois l'application lancée, La première interface que l'utilisateur aperçoit contient la liste infinie d'annonces se trouvant dans tout le Maroc. Dans cette liste, juste les informations qui intéressent les utilisateurs le plus sont affichées, nous citons : le titre de l'annonce, son prix et la ville d'où le vendeur publie l'annonce.

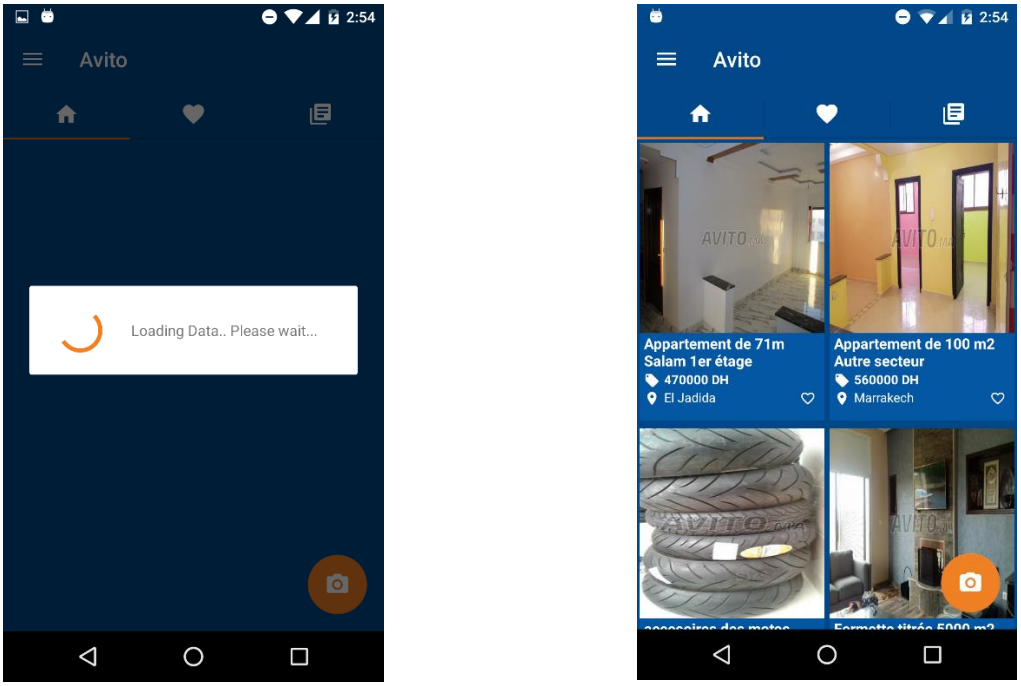
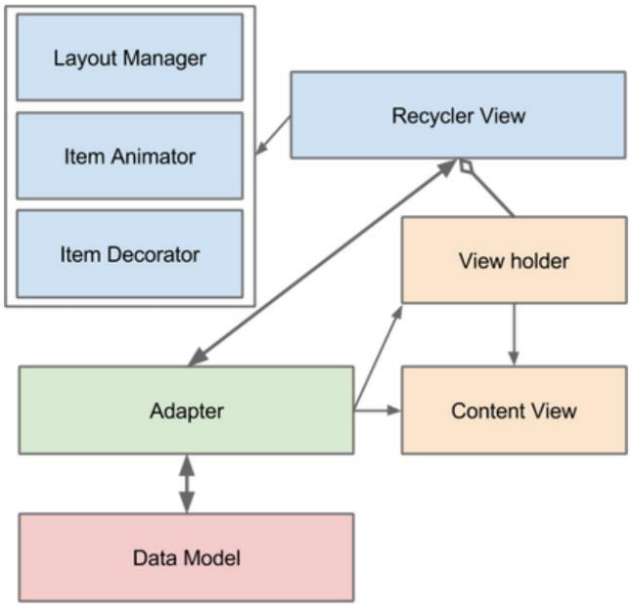


Figure 6 : Liste des annonces

Pour la représentation des annonces, nous avons opter pour le RecyclerView. Ce dernier offre une liberté de customisation du contenu de la liste en question et augmente à la fois la performance de l'application en termes de consommation de temps et de mémoire.

La figure ci-dessous illustre l'architecture et le fonctionnement d'un RecyclerView :



- **Adapter , Data Model , View Holder et Content View :** Ces classes travaillent ensemble pour préparer un seul index du RecyclerView. L'Adapter interagit avec le Data Model et l'attribut au View holder qui lui est responsable de convertir le Content View en code JAVA.
- **Recycler View, Layout Manager, Item Animator et Item Decorator :** Une fois la seule ligne de l'index est prête, le RecyclerView l'envoit vers les 3 classes Layout Manager, Item Animator et Item Decorator. L'item Animator et Item Decorator s'occupent de la manière dans sera affiché la ligne en question. Tandis que le Layout Manager s'occupe de la manière dans sera organisé l'ensemble des index.

Dans notre cas, nous avons utilisé un GridLayoutManager pour afficher deux annonce par la ligne.

L'application charge 35 annonces dans la liste, en scrollant vers le bas de liste, une fois arrivé vers la fin, l'application recharge 35 autres annonces dans la même liste, et ainsi de suite.

L'utilisateur peut aussi rafraichir la liste des annonces en effectuant un geste vers le bas.

Figure 7 : rafraichissement de la liste des annonces

- Consulter une annonce :

Pour avoir plus de détails sur une annonce, il suffit de cliquer dessus dans la liste. Les informations qui s'afficheront sont les suivant :

- Un Slider contenant toutes les photos insérées par l'annonceur, réalisé avec la librairie AndroidImageSlider
- Le titre de l'annonce
- La catégorie de l'annonce
- Le prix
- Une description sur l'annonce

- Un bouton pour appeler l'annonceur et un autre pour lui envoyer un message SMS.

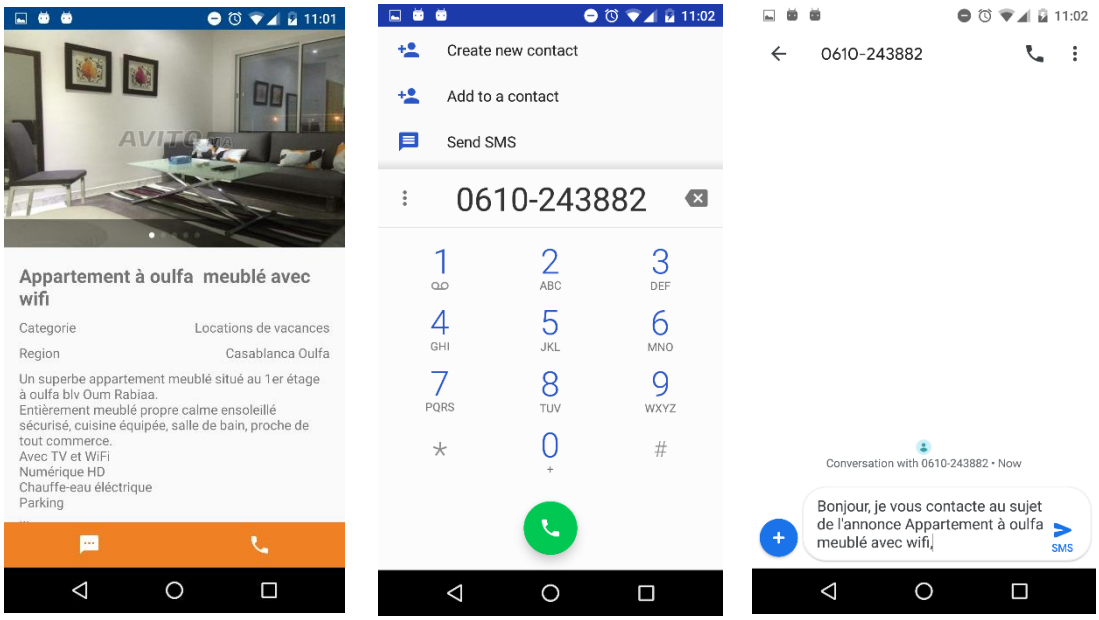


Figure 8 : Détails sur une annonce

- <u>Créer un compte sur Avito :</u>

L'utilisateur peut créer un compte sur Avito, en remplissant les champs suivant :

- Le nom
- La ville
- L'email
- Un mot de passe
- Le numéro de téléphone
- Le choix de masquer ou afficher le numéro de téléphone
- Le type du compte

Les vérifications d'email suivantes sont lancées avant de créer un nouveau compte :

- Vérifier si l'email entré est conforme à la forme général des emails (c'est à dire de la forme xxx@xxx.xx)
- Vérifier si l'email existe.
- Vérifier s'il y a un compte déjà créé avec le même email, ceci en envoyant une requête qui vérifie l'état de l'email (compte disponible ou existe déjà)

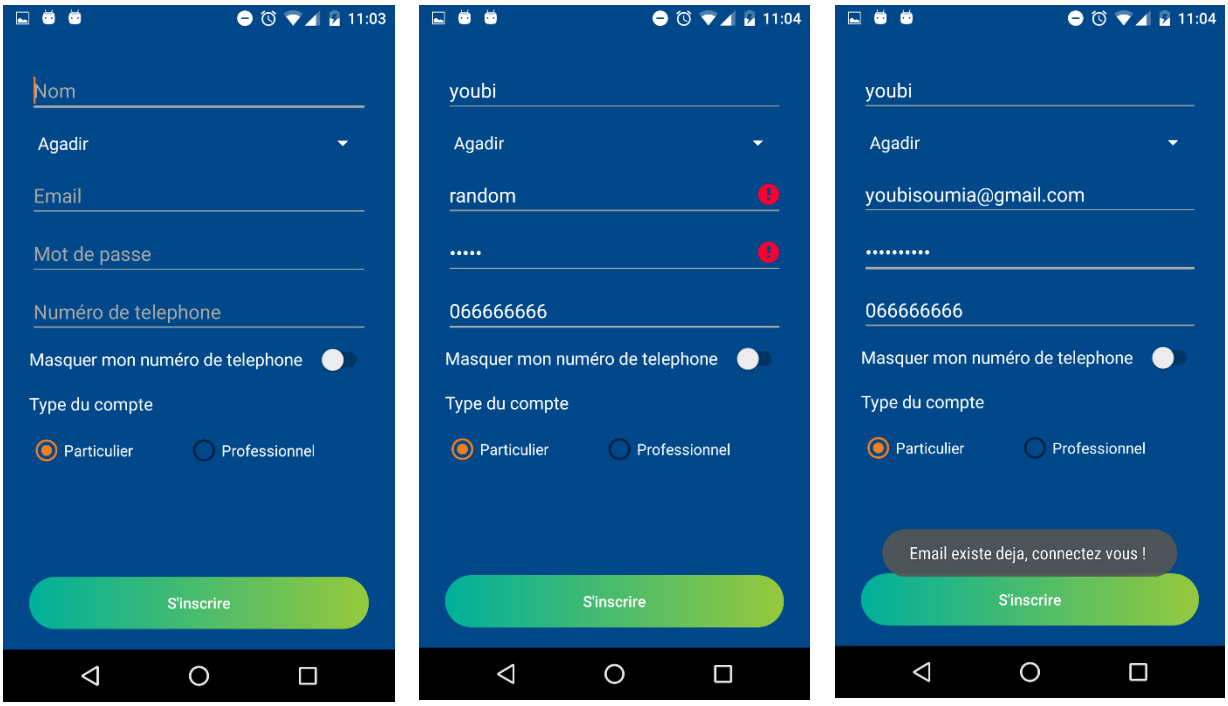


Figure 9 : Inscription sur l'application

- Authentification et déconnection :

Le système doit permettre à l'utilisateur qui a déjà fait l'inscription de passer à l'étape de l'authentification. L'utilisateur peut se connecter avec son email et un mot de passe. Et Il peut de même se déconnecter.

Ci-dessous l'interface d'authentification, et celle de « mon compte » ou l'utilisateur authentifié peut mettre à jour son compte ou se déconnecter

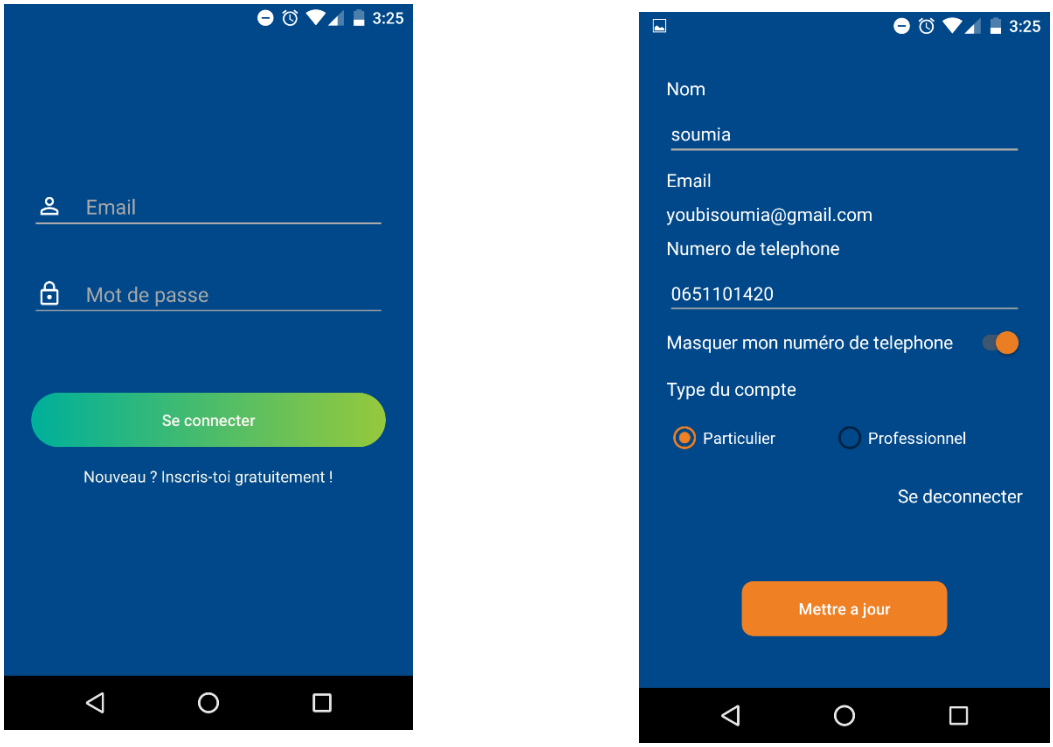


Figure 10 : Interfaces Login et Mon compte

- Enregistrer / Supprimer une annonce dans la liste des favoris :

L'utilisateur authentifiée, peut enregistrer une annonce dans sa liste des favoris ou en supprimer en cliquant sur l'icône en forme de cœur. La liste des favoris se trouve dans le deuxième onglet comme suit :

Figure 11 : Liste des annonces favorites

- Insertion d'une annonce :

L'insertion adoptée dans notre application est de deux étapes (3 si l'utilisateur n'est pas connecté). Dans la première étape, Nous avons utilisé la librairie **Matisse** qui permet de sélectionner jusqu'a 6 photos depuis la galerie de photos, ou les capturer depuis la camera. Et pour ceci, nous avions besoin de demander les permissions pour accéder au stockage externe et à la camera.

Dans la deuxième étape, l'utilisateur fait entrer les informations sur l'annonce.

Les interfaces d'insertion d'annonces :

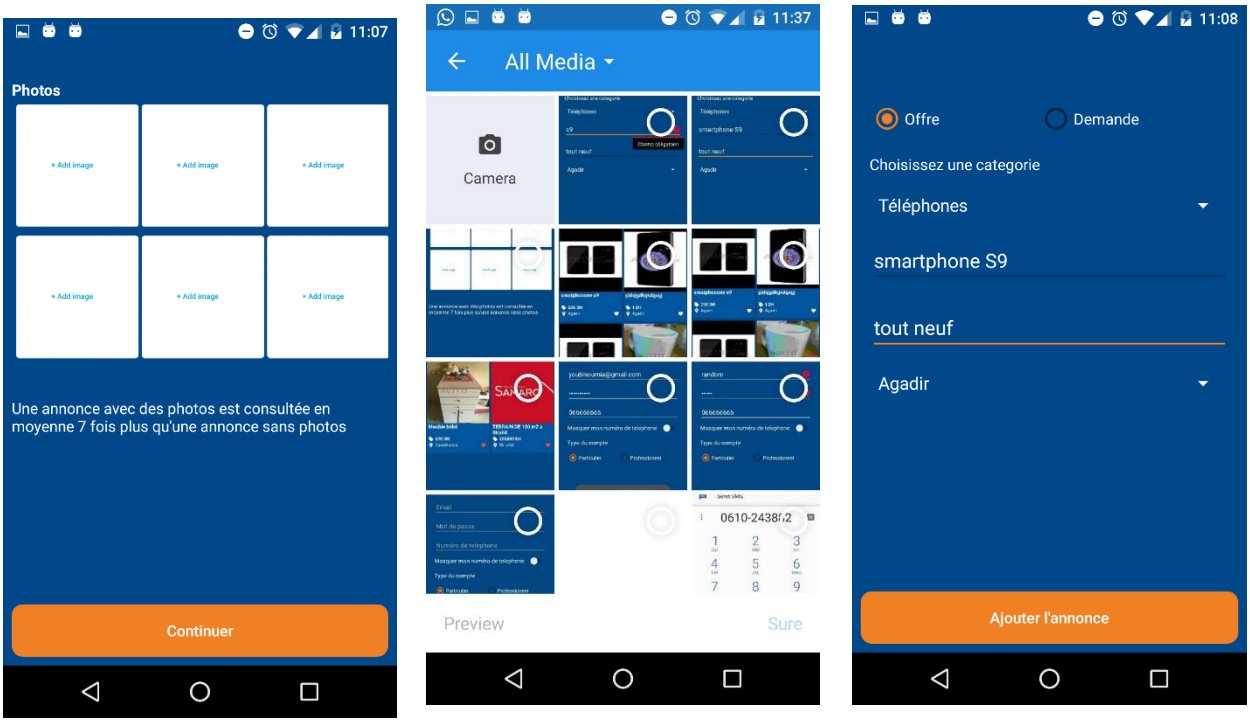


Figure 12 : Etapes d'insertion d'une annonce

- Affichage de Mes annonces :

L'utilisateur connecté, peut visualiser ses annonces insérées dans le troisième onglet de la première interface

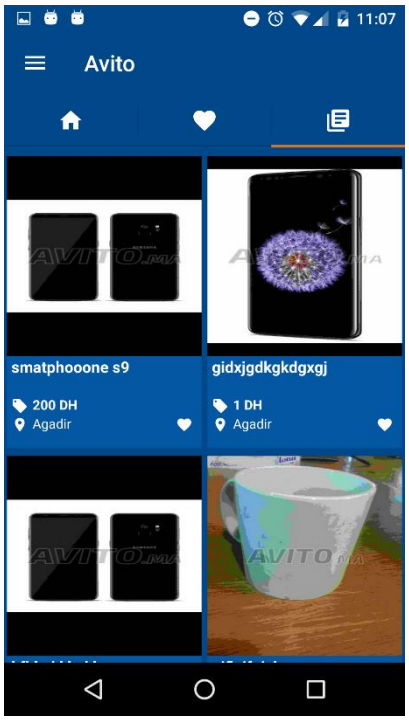


Figure 13 : Liste de mes annonces

Conclusion

Dans ce chapitre, nous avons présenté la méthodologie suivie pour réaliser le projet, ainsi que quelques interfaces de l'application.

Conclusion

Le présent rapport est le résultat d'un stage que j'ai effectué au sein d'Avito, filiale du Groupe Media Schibsted, dans le cadre de ma formation de 2eme année cycle d'ingénieur, filière génie informatique à l'école nationale des sciences appliquées d'Oujda.

Lors de ce stage de 2 mois, j'ai pu mettre en pratique mes connaissances théoriques acquises durant ma formation, de plus, je suis arrivée à réaliser les objectives que j'ai mis au début de cette période bien que j'ai vécu des difficultés, les méthodologies que j'ai utilisées pour les dépasser sont des signes de satisfaction ainsi que à l'aide de l'expertise des employés d'Avito.

Dans un premier temps, nous avons fait une analyse préalable de l'existant comme on a dû aussi se documenter sur l'API d'Avito et nous familiariser avec les nouvelles technologies utilisées. Ensuite, nous avons défini et étudié les aspects que nous devons travailler sur au niveau de l'application. La deuxième phase concernait la réalisation de l'application avec des fonctionnalités de base comme l'affichage des annonces, la consultation d'une annonce, la création d'un compte et l'authentification. Dans la troisième phase, nous avons rajouter la fonctionnalité d'insertion d'une annonce.

Ce stage a été pour moi une expérience très instructive à tous les niveaux, Au niveau technique, ça m'a donné de plus l'occasion d'acquérir des nouvelles connaissances à propos du développement Android, et de maitriser la librairie Retrofit qui seront certes utiles dans nos futures vies professionnelles. Sur le plan relationnel, j'ai réalisé à quel point une bonne communication et un bon sens de responsabilité sont indispensables dans notre vie professionnelle, ainsi que l'autonomie et la patience au niveau personnel.

Bibliographie

La librairie ImageSlider :

https://github.com/daimajia/AndroidImageSlider

La librarie Matisse :

https://github.com/zhihu/Matisse

Outil pour dessiner les diagrammes :

https://www.draw.io/

Plateformes de documentation :

https://developer.android.com/

https://futurestud.io/