

Etudiant : DUBOIS - Louis
Maître d'apprentissage: VILLAR - Julien



ANNÉE SCOLAIRE
2018/2019

UNIVERSITÉ DE LA
ROCHELLE

-

LICENCE
PROFESSIONNELLE
MÉTIERS DE
L'INFORMATIQUE

-

DÉVELOPPEMENT
MOBILE

Mémoire :

Alternance développement web et mobile

Agence REFERENCE – Tours

Du 3 Septembre au 30 août

Licence Professionnelle

REMERCIEMENTS

Je tiens à remercier dans un premier temps, toute l'équipe pédagogique du département informatique de l'université de la Rochelle et les intervenants professionnels responsables de la formation développement mobile, pour avoir assuré la partie théorique et l'introduction au professionnalisme durant l'année.

Je remercie également Monsieur JAMAL MALKI pour son aide, ses consignes ainsi que sa disponibilité durant cette alternance, son suivi des opérations a été primordial pour la réussite de cette alternance.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour l'expérience enrichissante et pleine d'intérêt qu'elles m'ont fait vivre durant cette année au sein de l'entreprise REFERENCE :

Madame Martine COHADE, Directrice de l'agence de Marketing REFERENCE, pour son accueil et la confiance qu'elle m'a accordés dès mon arrivée dans l'entreprise.

Monsieur Julien VILLAR, responsable du service Web de l'agence, mon maître d'apprentissage, pour m'avoir intégré rapidement au sein de l'entreprise et m'avoir accordé toute sa confiance ; pour le temps qu'il m'a consacré tout au long de cette période, sachant répondre à toutes mes interrogations ;

L'ensemble du personnel de REFERENCE pour leur accueil sympathique et leur coopération professionnelle tout au long de ces semaines.

Plus particulièrement Messieurs Julien GUILLON et Clément DUMUR pour leurs aides et soutiens pour la finalisation de certains projets ou encore pour leurs conseils et techniques qui m'ont permis de m'adapter et de progresser rapidement afin de terminer mes missions.

SOMMAIRE

Table des matières

INTRODUCTION	3
PRESENTATION DE L'ENTREPRISE	5
CARTE D'IDENTITE DE L'ENTREPRISE	
DOMAINE D'ACTIVITÉ	
GESTION DE PROJETS	
EXEMPLES DE REALISATIONS	
ENVIRONNEMENT DE TRAVAIL	
LES ATTENTES DE L'ENTREPRISE	12
MISSIONS DE DEVELOPPEMENT WEB	
CREATION DE L'APPLICATION CITYA IMMOBILIER	
TECHNOLOGIE UTILISÉE	14
DEFINITION DES FRAMEWORKS	
PRESENTATION DES FRAMEWORKS SILEX ET SYMFONY	
UTILISATION ET MISE EN PRATIQUE	
STRUCTURE AVEC LE MOTEUR TWIG	
PRESENTATION DE VISUAL STUDIO ET XAMARIN	
CRÉATION D'APPLICATIONS MULTIPLATEFORMES	
LES RÉALISATIONS CÔTÉ WEB	18
LES RÉALISATIONS DES SITES WEB BROSSET IMMOBILIER ET CBRE	
LES RÉALISATIONS SUR L'INTRANET	
LA RÉALISATIONS DE L'APPLICATION	21
LES OBJECTIFS DE L'APPLICATION	
CONCEPTION DE L'APPLICATION	
CREATION DE L'APPLICATION	
PUBLICATION SUR LES STORES	
RÉSULTATS ET DIFFICULTÉS RENCONTRÉS	34
CONCLUSION	32
BILAN PROFESSIONNEL	
BILAN PERSONNEL	
BIBLIOGRAPHIE	37
LEXIQUE	38
ANNEXE	40
ANNEXE 1 - REFERENCE DU MÉMOIRE	
ANNEXE 2 – REFERENCE DU RAPPORT	

INTRODUCTION

Dans le cadre de ma licence professionnelle développement mobile, j'ai effectué une alternance durant toute l'année scolaire en tant que développeur web et mobile, cette alternance avait pour but de me former au sein d'une équipe de développeur afin d'acquérir de l'expérience et du professionnalisme pour ma carrière professionnelle. J'ai eu l'opportunité d'intégrer l'agence de communication REFERENCE, basée à Tours, une filiale à 100% de CITYA IMMOBILIER.

Cette agence a pour vocation de gérer tout le domaine marketing et de communication de l'entreprise CITYA IMMOBILIER, dont la mission est d'assurer la création, la réalisation et le suivi de toutes les actions de communication favorables à la politique d'image et de marketing.

Durant mon alternance, j'ai été accueilli et suivi par mon maître d'apprentissage, Julien VILLAR, responsable du pool Web de l'agence.

Ayant déjà réaliser mon stage au sein de cette agence, Mon alternance a débuté par une présentation des nouveaux employés de l'entreprise, présentation des projets en cours et des projets que j'allais réaliser au cours de cette période.

J'ai accompagné les développeurs web dans leurs tâches quotidiennes et j'ai réalisé plusieurs projets en toute autonomie. Mon travail variait de la tâche quotidienne des développeurs web dans l'agence (réponse aux clients, modification de certaines pages, fixations de bug) au projet sur plusieurs semaines.

Ma principale mission a été de développer l'application de l'entreprise CITYA IMMOBILIER, que je présenterai pendant ce mémoire.

Vous pourrez suivre, au travers de ce mémoire, mon évolution tout au long de l'année ainsi que les différentes missions que j'ai pu effectuer. Vous y trouverez également une description de l'entreprise, mon ressenti sur cette alternance ainsi que ce qu'il m'a apporté personnellement et professionnellement.

Tous les mots suivis d'un * sont définis dans le lexique page 36.

PROBLÉMATIQUE

La réalisation de mon projet principale amène à une problématique. Cette problématique est la suivante : Où se situe les applications mobiles cross-platform dans le domaine de l'immobilier ?

En effet, la réalisation d'une application mobile dans le secteur de l'immobilier est complexe, pour la conception de l'application il faut prendre en compte le contexte dans laquelle cette dernière est déployer et comment peut-elle servir l'entreprise.

PRESENTATION DE L'ENTREPRISE

CARTE D'IDENTITE DE L'ENTREPRISE

CITYA est une société d'administration de biens fondée à Tours par Philippe Briand,

Diplômé de l'IUT et de l'IHEE, commandant de réserve de l'armée de l'air, Philippe BRIAND fonde l'entreprise dans les années 1990. Il a la volonté de construire une entreprise solide, avec des collaborateurs enthousiastes et fidèles. Aujourd'hui, l'entreprise compte 155 agences, réparties sur 98 villes en France métropolitaine et en outre-mer, au service de plus d'un demi-million de clients.

Proximité, innovation et qualité du service sont les valeurs de Citya et de ses 2 800 collaborateurs.

Durant mon alternance j'ai intégré l'agence REFERENCE, j'ai pu travailler avec son équipe pluridisciplinaire. Créée en 2000, une filiale à 100%, la SARL Référence, est en charge de la communication de l'entreprise et des agences. Elle est composée de plusieurs pôles distincts, le pôle Graphique, le pôle Événementiel/Communication, le pôle Agencement, et le pôle Web.

Chaque pôle pouvant diriger un projet de A à Z, les pôles sont souvent amenés à collaborer ensemble sur différents projets, notamment les collaborations entre les graphiques et les développeurs Web.

Le pôle web est assuré par des développeurs web, d'un web master, de deux Community Manager. L'ensemble de l'équipe est dirigé par un responsable web.

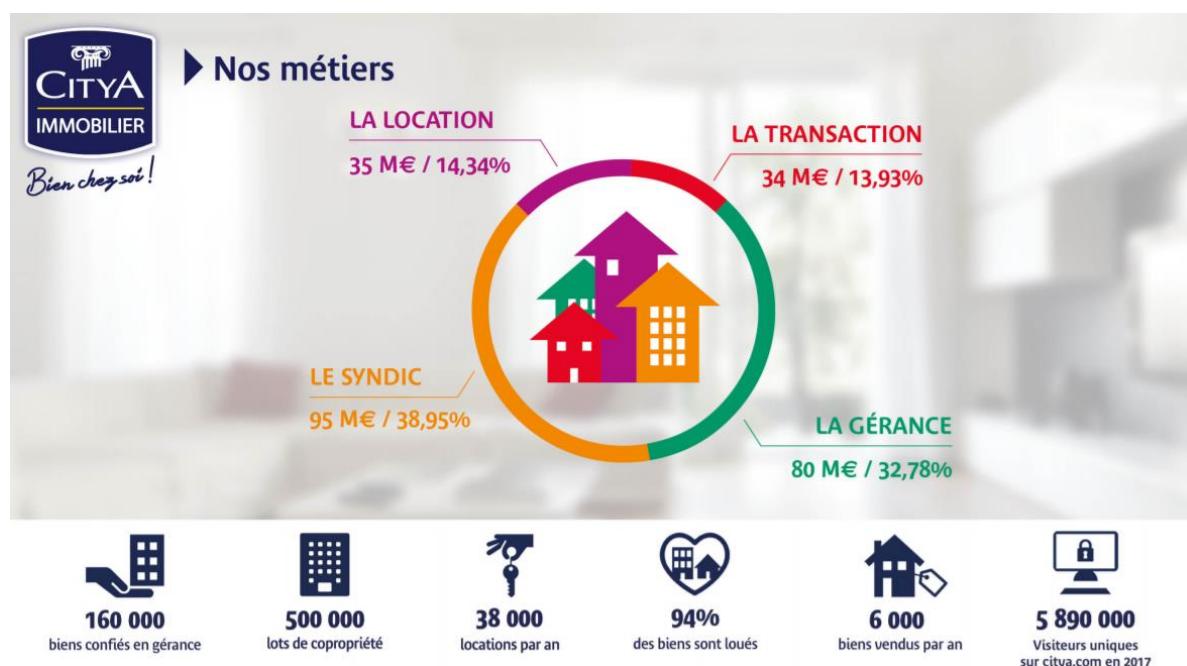
L'agence REFERENCE est à disposition de l'entreprise et des agences Citya IMMOBILIER, elle possède donc son propre site de vente, au service des directeurs d'agences, pour répondre à leurs besoins d'agencement, de décoration d'intérieur des agences ou même de goodies.

Pour réaliser mes différentes missions sur les intégrations de l'intranet, j'ai été en contact avec certains représentants de filiales de l'Arche, j'ai été à l'écoute de leurs modifications et suggestions sur les pages créées.

DOMAINE D'ACTIVITÉ

Une présentation de l'entreprise Citya IMMOBILIER et de la société Arche s'impose afin de comprendre les solutions et les techniques mise en place par l'agence REFERENCE pour assurer la création, la réalisation et le suivi de toute action de communication demandée.

Citya IMMOBILIER est composé de 4 métiers d'administrations de biens ;
La location, la Transaction, la Gérance et le Syndic.



Citya est une structure indépendante, gérée par des professionnels, chaque cabinet étant une entreprise de plein exercice. Les cabinets exercent l'ensemble des métiers de service de l'immobilier.

Arche est une société regroupant un ensemble d'entreprises et de filiales. Avec plus de 10 000 collaborateurs, 700 millions € de chiffre d'affaires, plus de 55 000 transactions par an (les réseaux Citya Immobilier, Laforêt et Guy Hoquet confondus),

Arche se fait fort de proposer à ses clients une palette de prestations complémentaires aux cœurs de métiers de l'immobilier (Location, Gestion, Vente, Syndic) grâce à ses différentes filiales : des solutions d'assurances avec Saint-Pierre Assurances, de financements avec API financement, de diagnostics avec la Snexi...

Elle confirme ainsi ses vellités d'apporter un service complet à ses clients qui bénéficient d'un panel de solutions et de conseils qualitatifs, pour la vente ou l'acquisition de leur logement ou encore pour leur gestion patrimoniale.

Référence est une agence de communication dédiée à l'Arche et ses filiales, dont la mission est d'assurer la création, la réalisation et le suivi de toutes les actions de communication favorables à la politique d'image et de marketing. Elle réalise :

Le Print, signalétique, PLV, Web, goodies, agencement et décoration intérieure, campagnes de communication nationale, l'événementiel, la communication publicitaire, le

site Citya.com, réseaux sociaux, application ainsi qu'une communication Intranet MYCITYA pour Citya Immobilier.

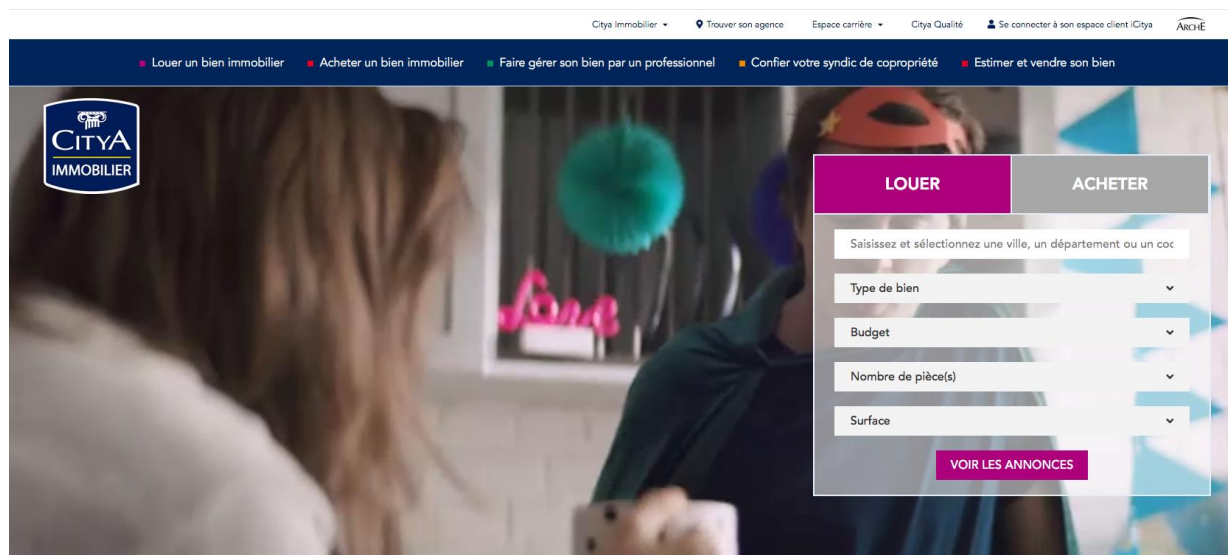
Cet intranet, la boîte à outils des collaborateurs, met à disposition des fichiers de tous les métiers pour plus de 3 000 collaborateurs, l'actualité des agences au quotidien, l'actualité de l'immobilier avec "Aucoeurdelimmo" et un accès à de nombreux services, tels que : Citya Qualité, Citya Assurances, Dépannage informatique, Catalogue des formations, Challenges Location / Transaction, Boutique en ligne Agence Référence. Ces services sont primordiaux pour l'entreprise Citya et la relation entre les différentes agences.

GESTION DE PROJETS

REFERENCE est en mesure de gérer de bout-en-bout la réalisation de projets, que ce soit en matière de communication ou de développement web, depuis l'expression du besoin du client à la réalisation, en passant par l'élaboration du cahier des charges. Cette gestion assure la qualité et le suivi de ses développements dans les différents domaines. Comme exposé ci-suit, l'agence de communication intégrée Citya IMMOBILIER accompagne les collaborateurs dans la réalisation de projets de communication. Les projets informatiques sont versionnés avec Mercurial SCM. Mercurial est un outil gratuit et distribué, de gestion du contrôle de source. Il gère efficacement les projets de toutes tailles et offre une interface simple et intuitive. On peut comparer Mercurial à Git. Chaque projet de site web possède un environnement de préprod afin de tester la solution avant de la mettre en production. Pour l'intranet, une phase de déploiement permet de travailler en local et d'ajouter des modifications sans risque d'altérer les différentes recettes.

EXEMPLE DE RÉALISATIONS

La filiale REFERENCE réalise de nombreux projets de communication pour toutes les agences Citya, de la campagne print, au développement de leur différents sites Web. La création et la maintenance du site Citya.com est un bon exemple de réalisation. Ce site sert de communication externe pour Citya IMMOBILIER avec plus de 5 980 000 visiteurs uniques et 18 500 000 pages consultées en 2017, il a été refait cette année par le Développeur Julien GUILLON et l'intégrateur Julien ALLAIN. Ce site est une illustration des sites modernes. Il est dit « responsive », comporte une géolocalisation de bien, un « flat design » et une typographie adaptée.



CITYA IMMOBILIER, MARQUE LEADER AU PLAN NATIONAL !
IMPLIQUÉE DANS LA VIE ÉCONOMIQUE LOCALE

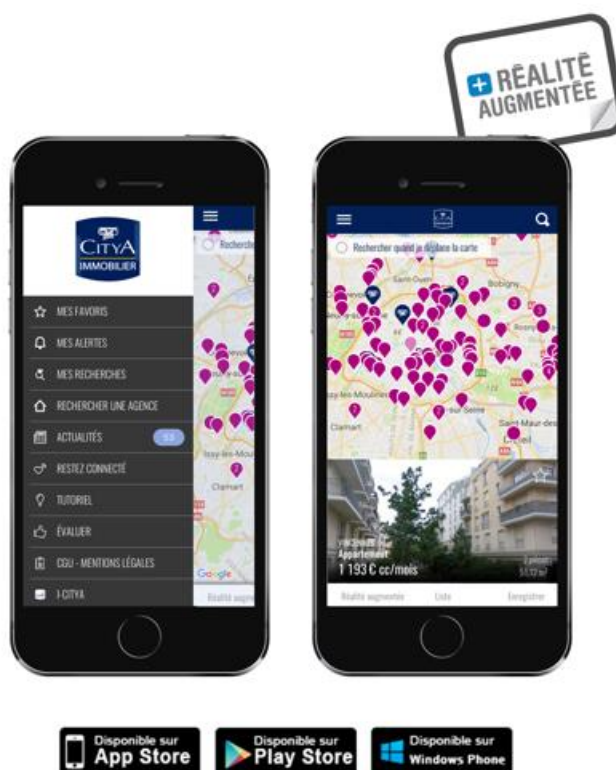
175

2000

1 000 000

Une seconde illustration de la production de l'agence est la création de l'intranet MyCitya, cet intranet regroupe tous les outils des collaborateurs de l'Arche (Avec plus de 10 000 collaborateurs), cette intranet sert aussi de réseau social, il affiche les actualités et les anniversaires des collaborateurs "proches " (les collaborateurs de son agence ou d'une agence voisine, les directeurs...).

L'application Citya, permet de consulter les biens à la vente ou à la location à proximité. Cette application étant vieillissante et contenait des erreurs. Par conséquent, mon projet principal était de reprendre l'application à zéro et d'en réaliser une nouvelle version.



Ces exemples caractérisent l'agence REFERENCE dans la réalisation de ces projets. Innovante et professionnelle, l'équipe du pool web permet la concrétisation des idées des collaborateurs de l'Arche.

ENVIRONNEMENT DE TRAVAIL

J'ai effectué mon alternance dans les locaux de REFERENCE, dans l'open-space dédié aux développeurs. Il se présente comme un open-space composé d'une petite dizaine de bureaux. L'open-space est un avantage certain, car il facilite l'intégration et la communication entre les membres de l'équipe, la demande d'aide lors de problèmes rencontrés et surtout permet de poser plus aisément ses questions.

Pendant la durée de mon alternance, j'ai travaillé avec un environnement MAC, ayant des accès aux serveurs distants de l'agence et au Mercurial du pool Web Reference. J'avais les accès aux différents serveurs de préprod, pour visualiser les projets avant de les mettre en production. La préprod est accessible à certains collaborateurs (à qui les droits ont été attribués) afin qu'ils puissent vérifier si les projets correspondent au cahier des charges. Les résultats de missions sont accessibles en ligne grâce à une URL* spécifique accessible seulement depuis le réseau de l'entreprise et avec les logins autorisés. L'avantage de cette méthode est de pouvoir avoir un aperçu du résultat rapidement tout en n'interférant pas directement avec les outils utilisés par les collaborateurs.

L'intranet utilisait une identification par base de données. Les formulaires, les accès à certaines pages et un grand nombre de fonctions de l'intranet reposaient sur les données des utilisateurs et des différentes agences. Pour le bon développement de mes missions en lien avec cet Intranet. Le déploiement en local me permettait d'utiliser les copies des bases de données de l'entreprise, je pouvais donc les modifier et effectuer des tests avec.

Durant mon alternance, j'ai utilisé différents logiciels et outils me permettant de développer mes projets et de les visualiser en situation de production.

Je vais maintenant vous présenter les différents outils utilisés durant mon alternance :
Les OUTILS :

	<p>Microsoft Visual Studio est une suite de logiciels de développement pour Windows et mac OS conçue par Microsoft.</p> <p>Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages.</p> <p>Ce logiciel m'a permis de développer l'application CITYA IMMOBILIER. L'IDE intègre les SDK pour le développement cross-plateforme. Il permet l'accès aux émulateurs IOS et Android ce qui permet de tester le bon fonctionnement de l'application sur les différentes plateformes.</p>
	<p>N'est autre que le célèbre logiciel de gestion de bases de données. MySQL open source*, propose de gérer vos bases de données.</p> <p>Je l'utilisais pour visualiser les tables* de l'entreprise et les gérer, l'accès aux bases de données étant primordial pour les modifications sur l'intranet.</p>
	<p>PhpStorm est un éditeur pour PHP3, HTML, CSS et JavaScript4, édité par JetBrains. Il possède une coloration syntaxique, affichage des erreurs à la volée, auto-complétion intelligente du code et le réusinage de code.</p> <p>Il intègre l'envoi des fichiers via FTP, un logiciel de gestion de versions, compatible Git, Mercurial et Subversion.</p> <p>Il permet aussi de visualiser l'architecture de bases de données de différentes sources (MySQL, SQLite, ...).</p> <p>Il est donc parfaitement adapté pour l'édition de site utilisant le Framework Symfony. Je l'ai utilisé pour la création complète des différents sites. Je l'ai aussi utilisé pour intégrer les modifications sur l'Intranet, grâce à son logiciel de gestion de version, le versioning de l'Intranet a été simplifié.</p>
	<p>Mercurial est un logiciel de gestion de versions décentralisé disponible sur la plupart des systèmes Unix et Windows.</p> <p>Mercurial est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Il permet notamment de retrouver les différentes versions d'un lot de fichiers connexes.</p> <p>Il est plus précisément un logiciel de gestion de versions décentralisé. Il est utilisé pour le versioning de chaque projet à Reference. Chaque développeur possède un identifiant unique.</p>
	<p>MySQL Workbench est un outil visuel unifié destiné aux architectes de bases de données, aux développeurs et aux administrateurs de bases de données. MySQL Workbench fournit la modélisation de données, le développement SQL et des outils d'administration complets pour la configuration du serveur, l'administration des utilisateurs, la sauvegarde, etc. MySQL Workbench est disponible sous Windows, Linux et Mac OS X.</p> <p>Il m'a permis de visualiser les copies des bases de données de l'intranet ou de Citya en local (pour effectuer des tests ou des vérifications).</p>

LES ATTENTES DE L'ENTREPRISE

PRESENTATION DES MISSIONS WEB

Mes premières missions durant mon alternance ont été la création de sites web. Reference venait de recevoir deux commandes de création de sites, le premier site était la site marchand de Brosset Immobilier. Brosset Immobilier est un réseau d'agence Immobilières situées sur Tours et son agglomération. Cette institution tourangelle maîtrise tous les métiers de l'immobilier (achat, vente, location, gérance, syndic, expertise). Ce site présente Brosset immobilier, ces agences, les métiers de l'immobilier, et un formulaire de recherche permettant de trouver toutes les annonces immobilières en Touraine créer par Brosset Immobilier depuis leur logiciel.

Pour la réalisation de ce premier projet, j'étais accompagné d'un développeur sénior, Vincent VALLE, qui a réalisé le déversement des annonces, depuis le logiciel utilisé par Brosset Immobilier lors de la visite de bien avec les clients, vers un serveur local. Chaque annonce est représentée par un fichier XML contenant la description du bien.

Le site est un projet Symfony stocké sur un serveur local. J'ai commencé le projet en créant les pages principales, les pages statiques de présentation. Puis, pendant que je commençais un nouveau projet de création de site web, Vincent VALLE ajouta le formulaire et les listes d'annonces.

Mon autre projet de création de site web, était la refonte du site marchand de l'entreprise CBRE, à Tours, membre indépendant du réseau CBRE, N°1 mondial de l'immobilier d'entreprise.

Les restes de mes missions web étaient l'ajout de nouvelles fonctionnalités sur l'intranet et l'amélioration de certaines fonctionnalités existantes.

Comme expliqué précédemment, l'intranet regroupe un ensemble d'outils utilisés par tous les collaborateurs de l'entreprise.

La maintenance de cet ensemble d'outils était primordiale, une préproduction ainsi qu'un déploiement en local de l'intranet était obligatoire pour son développement.

Mon premier projet sur l'intranet était l'amélioration du système d'actualités.

La page principale de l'intranet est une liste d'actualité, ces actualités peuvent être des articles ajoutés par les Community Managers, des partages d'articles extérieurs ou encore l'anniversaire d'un collaborateur.

Les anniversaires sont affichés selon le rôle de l'utilisateur. S'il est salarié d'une agence, il ne peut voir que les anniversaires dans son agence, de la holding ou des représentants de l'entreprise. S'il est directeur, les règles ne sont pas les mêmes, s'il appartient à la holding aussi, etc.

Le deuxième projet sur lequel j'ai travaillé est le livret RH de l'entreprise, ce livret est affiché lors de la première connexion d'un nouveau collaborateur. Le livret répertorie toutes les filiales et entreprise en lien avec Citya IMMOBILIER, leurs services, ainsi que certaines statistiques. La suite de ce livret sont les conditions d'utilisation de l'intranet ainsi que différentes mentions légales. Une fois accepté le collaborateur accède à une courte présentation de l'utilisation de l'intranet.

Enfin, le dernier projet en lien avec l'intranet sur lequel j'ai pu travailler est le Dashboard des statistiques des sites web de l'Arche.

Ce Dashboard est seulement accessible par certaines personnes définies : M. Philippe BRIAND (Président Fondateur), M. Frederic CHAMINADE (Directeur Général) et Mme Martine Cohade (Directrice de l'agence Reference).

Il regroupe les statistiques de fréquentation quotidiennes des sites sous formes de graphes.

Il regroupe aussi l'évolution du nombre d'abonnés sur les réseaux sociaux des sites concernés.

Ce Dashboard permet de vérifier si certaines dispositions tels que des campagnes ou des publications augmente la fréquentation des sites.

Il permet aussi de comparer l'évolution de fréquentations quotidiennes par rapport à l'année dernière.

CREATION DE L'APPLICATION CITYA IMMOBILIER

Le principal projet durant cette alternance fut la refonte totale de l'ancienne application mobile de l'entreprise.

L'ancienne application permettait à Citya immobilier de couvrir une plus grande étendue de clients. Elle proposait les mêmes services que le site web : permettre à l'utilisateur de rechercher un bien selon un lieu et des critères définis. Elle permettait donc à l'entreprise d'améliorer la satisfaction du client, d'améliorer l'efficacité et les performances du développement économique de l'entreprise.

La nouvelle version de l'application aura pour but de proposer à l'utilisateur tous les biens gérés par Citya Immobilier. Ces biens seront accessibles par recherche de l'utilisateur, sous forme de liste ou de carte.

En plus de proposer les biens, l'application permettra d'afficher les agences de l'entreprise à proximité ou en totalité, ainsi que les actualités. Elle se voudra remplacer l'ancienne version et reprendre position sur le marché mobile de vente de biens.

L'application a donc pour finalité d'étendre le périmètre de clients potentiels et de pouvoir atteindre plus de personne lors de campagnes ou d'offres grâce aux notifications qu'elle générera. Ces notifications font parties des plus grands avantages des applications, le mobile étant, à notre époque, une extension du cerveau des utilisateurs, les notifications permettent d'impacter un maximum.

Dans la suite du rapport, je vais d'abord décrire comment j'ai réalisé les missions en présentant la technologie utilisée, la manière dont elle est utilisée et ses avantages.

Ensuite je présenterai quelques exemples et la mise en pratique.

Je vais ensuite développer plus en détails le déroulement des missions et les résultats obtenus.

TECHNOLOGIE UTILISÉE

DEFINITION DES FRAMEWORKS

Un Framework n'est pas indispensable pour réaliser une application web, un site statique ou un site de taille réduite, mais il apporte ce qu'il faut pour être plus efficace, et créer une application plus professionnelle.

Un Framework offre donc la possibilité de développer une application professionnelle et reconnue. Il permet de réaliser une application structurée, maintenable et évolutive.

Enfin, il apporte certaines parties de base qui permettent de ne pas avoir à tout refaire de A à Z.

Abstraction de la base de données : Ce Framework utilise PDO, une solution qui permet d'ajouter une couche d'abstraction* entre les bases de données et le code créé. Il est ainsi possible de réaliser des opérations courantes comme la récupération de données ou la sauvegarde d'un objet sans se soucier du code SQL à écrire.

- **Gestion des formulaires :** Le framework offre la possibilité de générer en grande partie tous les widgets HTML, il se charge de la validation du formulaire et de la sécurité CSRF (Cross-Site Request Forgery).
- **Gestion d'utilisateurs :** La plupart des frameworks ont tout ce qu'il faut pour pouvoir gérer l'authentification d'utilisateurs. Ils gèrent la connexion, la déconnexion, la création, la gestion des sessions et des droits.
- **Gestion des erreurs et bien plus :** Certains frameworks comme Symfony offrent de très bons outils de débogage* ainsi qu'un "profilier" qui permet de vous renseigner sur tout ce qu'il se passe dans l'application (variables globales, requêtes de base de données, logs, temps de chargement, etc.).
- **Moteur de template :** De nombreux frameworks intègrent un moteur de templates. Celui-ci permet de simplifier grandement l'écriture de votre code HTML tout en étant propre et efficace. Le moteur utilisé dans mon cas est le moteur Twig comme présenté dans la suite du rapport.

PRESENTATION DES FRAMEWORK SILEX ET SYMFONY

Silex est un micro-framework* PHP développé par la société française SensioLabs, créatrice du framework Symfony. Silex est en quelque sorte le petit frère de Symfony et les deux frameworks reposent sur les mêmes composantes.

Contrairement à Symfony qui fournit (et impose) une architecture complète (dite « full stack »), Silex est un framework minimaliste qui laisse beaucoup de liberté au développeur. C'est pourquoi on peut le qualifier de micro-framework. Il fournit un

ensemble réduit de services au-dessus desquels on peut développer une application Web. Son minimalisme le rend idéal pour s'initier en douceur au fonctionnement d'un framework PHP.

L'avantage de ces frameworks, c'est de pouvoir travailler dans le respect des bonnes pratiques et de réutiliser des méthodologies de conception standardisées. On peut particulièrement citer le design pattern MVC, acronyme de l'expression "Modèle - Vue - Contrôleur". Un « design pattern » est traduit par "patron de conception", c'est donc un modèle qui a fait ses preuves et qui s'avère indispensable à connaître dans le domaine de la programmation.

Modèle : Le modèle est la partie du code qui se charge d'interagir avec la base de données. On y retrouve donc les appels SQL que nous aurons définis. Comme par exemple une fonction qui est équivalent à une requête SQL comme la liste des utilisateurs d'une agence

Vue : La vue est la partie du code qui se charge uniquement d'afficher les données qui lui ont été fournies. En aucun cas, la vue ne doit effectuer de traitement sur les données, sauf dans le but d'en embellir l'affichage.

Contrôleur : Le contrôleur est le centre de notre design pattern. Il reçoit la requête HTTP, l'interprète et coordonne le tout. Il se charge de demander au modèle les données puis effectue plus ou moins de traitements dessus afin d'envoyer à la vue les données à afficher et de retourner une réponse à l'émetteur de la requête.

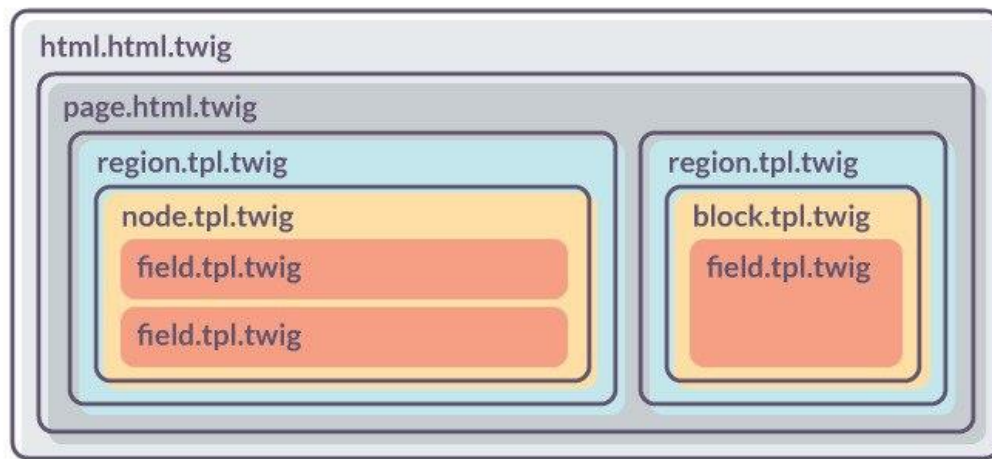
Routeur : Le rôle du routeur est, à partir d'une URL, de déterminer quel contrôleur appeler et avec quels arguments. Cela permet de configurer son application pour avoir des url plus parlantes pour les utilisateurs.

Ce Framework permet d'innover, Symfony est tout ce qu'on attend d'un Framework : rapidité, flexibilité, des composantes réutilisables, etc. De plus avec la grande communauté de Symfony j'ai pu me documenter sur chaque impasse et problèmes rencontrés grâce à des conversations par mail partagés. Des channels IRC existent pour exposer ses problèmes, de nombreux travaux sont publiés régulièrement et, bien évidemment, le Site du Zéro est là pour accompagner, et des sites d'e-learning comme open Classroom ou même le site officiel de Symfony possède une documentation complète et adaptée à chaque utilisateur.

Symfony tend à respecter un maximum de conventions de codage. Le Framework s'efforce de respecter des standards comme PSR-0 et HTTP. Enfin, Symfony est distribué sous licence Open Source MIT, qui n'impose pas de contraintes et permet le développement de l'Open Source ainsi que des applications propriétaires.

STRUCTURE AVEC LE MOTEUR TWIG

Le moteur de Template twig est utilisé pour la vue de la page, il est l'équivalent de l'HTML. Pour chaque filiale, un menu était intégré dans l'intranet. Ce menu héritait du bloc de présentation de l'intranet (Cf. bloc_presentation_base_template), avec le moteur Twig, la page web du site peut être comparé à un puzzle.



On ajoute les pièces secondaires, qui représentent finalement la partie propre de la filiale à des pièces principales comme le menu horizontal et vertical ainsi que l'arrière-plan commun pour toutes les pages de l'intranet.

PRESENTATION VISUAL STUDIO ET XAMARIN :

Visual studio est l'environnement de développement pour créer des applications Xamarin.

Il intègre des kits de ressources selon le type de développement, comme le développement de jeux sous Unity, le développement d'application sur Windows, des applications .Net et bien sûr des application Xamarin.

Xamarin Forms est un kit de ressources d'interface utilisateur multiplateforme qui permet aux développeurs de créer efficacement des dispositions d'interfaces utilisateur natives pouvant être partagées entre des applications iOS, Android et UWP (plateforme Windows universelle).

La création d'applications mobiles peut être aussi simple qu'ouvrir l'IDE, écrire et tester une application puis soumettre le résultat à un App Store, le tout en une seule après-midi. Cela peut aussi consister en un processus extrêmement complexe impliquant une conception préalable rigoureuse, des tests d'utilisabilité, des tests d'assurance qualité sur des milliers d'appareils, un cycle de vie complet avec des versions bêta, puis un déploiement effectué de plusieurs façons différentes.

Xamarin permet de développer dans C#, avec une bibliothèque de classes et un runtime qui fonctionnent sur toutes les nombreuses plateformes, notamment iOS, Android et Windows, tout en compilant des applications natives (non interprétées) qui sont suffisamment performantes même pour les jeux exigeants.

Les fonctionnalités de Xamarin viennent répondre aux besoins des développeurs qui souhaitent utiliser un langage et une plateforme modernes pour développer des applications mobiles multiplateformes.

Sortie de l'application

Lorsque les applications Xamarin sont compilées, un paquet d'application est créé (un fichier .app dans iOS ou un fichier .apk dans Android). Ces fichiers ne se distinguent pas des paquets d'application générés à l'aide des IDE par défaut de la plateforme, et peuvent être déployés exactement de la même façon.

Source : <https://docs.microsoft.com/fr-fr/xamarin/cross-platform/get-started/introduction-to-mobile-development>

Création d'applications multiplateformes

Le cycle de vie du développement mobile n'est globalement pas différent de celui des applications web ou pour poste de travail. Comme pour ces applications, le processus comprend généralement 5 parties principales :

1. Lancement : toutes les applications commencent par une idée. Cette idée est généralement affinée jusqu'à constituer une base solide pour une application.
2. Conception : la phase de conception consiste à définir l'expérience utilisateur de l'application, comme la présentation générale, son fonctionnement, etc., ainsi que la conversion de cette expérience utilisateur en une interface utilisateur appropriée, généralement avec l'aide d'un infographiste.
3. Développement : généralement la phase la plus consommatrice de ressources, c'est la phase de création réelle de l'application.
4. Stabilisation : quand le développement est suffisamment avancé, l'assurance qualité commence généralement à tester l'application et les bogues sont corrigés. Le plus souvent, une application passe par une phase bêta limitée, durant laquelle un public plus large a la possibilité de l'utiliser, de fournir des commentaires et d'obtenir des modifications.
5. Déploiement

Voici un résumé des points clés pour la création d'applications multiplateforme de Xamarin :

Utilisez C# -écrire vos applications C#. Code existant écrit dans C# peut être déplacée vers iOS et Android à l'aide de Xamarin très facilement et évidemment utilisé dans les applications Windows.

Utiliser des modèles de conception MVC ou MVVM -développer d'Interface d'utilisateur de votre application à l'aide du modèle de modèle // contrôleur d'affichage. L'architecture de votre application à l'aide d'un modèle/affichage/contrôleur ou du modèle/affichage/ViewModel où il existe une séparation claire entre le « modèle » et le reste. Déterminer quelles parties de votre application sera à l'aide des éléments d'interface utilisateur native de chaque plateforme

(iOS, Android, Windows, Mac) et utilisez-le comme indication pour diviser votre application en deux composants : « Core » et « Interface utilisateur ».

Créer des interfaces utilisateur natives -chaque application spécifique à un système d'exploitation fournit une couche d'interface utilisateur différentes (implémenté dans C# outils de conception avec l'aide de l'interface utilisateur native) :

Sur iOS, utilisez les APIs UIKit pour créer des applications natives attrayantes, éventuellement qui utilisent le Concepteur de Xamarin iOS pour créer votre interface utilisateur visuellement.

Sur Android, utilisez Android.Views pour créer des applications de qualité natif, en tirant parti de concepteur d'interface utilisateur de Xamarin.

Sur Windows vous utiliserez XAML pour la couche de présentation, créé dans Visual Studio ou Blend Concepteur d'interface utilisateur.

Sur Mac, vous allez utiliser des tables de montage séquentiel pour la couche de présentation, créée dans Xcode.

Les projets Xamarin.Forms sont pris en charge sur toutes les plateformes et vous permettent de que créer des interfaces utilisateur qui peuvent être partagées entre les plateformes à l'aide de XAML de Xamarin.Forms.

LES RÉALISATIONS DES SITES WEB BROSSET IMMOBILIER ET CBRE :

Les deux projets sont des projets Symfony, chacun découpé en deux bundles distincts, un bundle intitulé CoreBundle, contenant les vues, les types des formulaires ainsi que les contrôleurs. Le deuxième bundle, Entitybundle, contient les entités ainsi que leur Dépôt. Chaque projet été versionné sur un mercurial.

En collaboration avec l'agence de communication IDÊKA, cette agence m'a envoyé les maquettes des deux sites web, en suivant ces maquettes, j'ai réalisé le graphisme des deux sites. Les pages était de simple pages html avec des fonctions JQuery et Js.

Pour les formulaires de recherche, l'utilisateur saisie des détails de recherche, puis lors de la soumission du formulaire, l'utilisateur était redirigé vers la page de résultats.

Les paramètres de la recherche étaient passé dans les paramètres de la page.

Une requête SQL était effectuée par le site vers la base de données avec les données de la recherche. Avec le résultat de la requête, le programme retournée une liste d'annonces, affichées dans la page.

LES RÉALISATIONS SUR L'INTRANET

Durant ma période d'alternance, j'ai réalisé plusieurs missions sur l'intranet de l'entreprise, comme expliqué au début de mon mémoire, l'intranet est l'outil de tous les collaborateurs et collaboratrices.

Certaines missions, comme expliqué ci-dessus, ont été ajoutées pendant mes projets principaux. Il fallait réaliser les missions dans le temps imparti. Pour cela, il m'a fallu être professionnel et rapide dans la réalisation de ces tâches. On peut citer la modification de la page de présentation de Citya-immobilier, la finalisation de la partie Citya-patrimoine avec l'ajout d'un formulaire et de plusieurs pages classiques ainsi que la réalisation de la page d'annonces pour les ressources humaines.

La première mission était la création du livret RH, ce livret récapitulant et présentant les filiales ou partenaires de l'Arche. Toutes les filiales étaient disposées dans un conteneur, lors d'un clic sur une filiale, un modal avec des informations sur la filiale était affiché (Voir Annexe 3).

Le livret se termine par l'affichage des différentes mentions légales, spécifique au numérique et à l'intranet. Le nouveau collaborateur devait accepter les mentions légales pour continuer à utiliser l'intranet.

Lorsqu'il accepte, des champs liés à son compte étaient changés, par exemple le champ `Accepted_Charte_Informatique` est remplie avec la date à laquelle l'utilisateur a accepté la charte.

La deuxième mission était la création d'un tableau de bord contenant toutes les statistiques des sites web de l'Arche.

Un onglet est situé en haut de page, pour que l'utilisateur puissent sélectionner le sites web voulus.

Ce tableau de bord affiche sous formes de graphes les statistiques web, ces stats sont remplies à l'aide d'un formulaire que j'ai intégré dans le backoffice de l'intranet.

Chaque statistique est représenté par un objet avec en propriété, la date, le nombre de sessions sur le site, les abonnés Facebook, les abonnés Twitter...

Lors d'un clic sur un site dans la barre d'onglets, la page est réactualisée, on charge de nouvelle donnée appartenant au site sélectionné :

```
$statsTwoYear = $this->getStatistiquesFilterBySites($site);
$statsRs = StatistiqueSiteWeb::lastDateRS($site);
$arrayDiffDay = $this->getDayRsDiff($statsRs);
```

Ces fonctions utilisent des méthodes jointes aux Models de l'intranet. Les méthodes effectuent des requêtes Query sur la base de données et retourne une liste d'objets.

Ces données sont passé en paramètre dans la page :

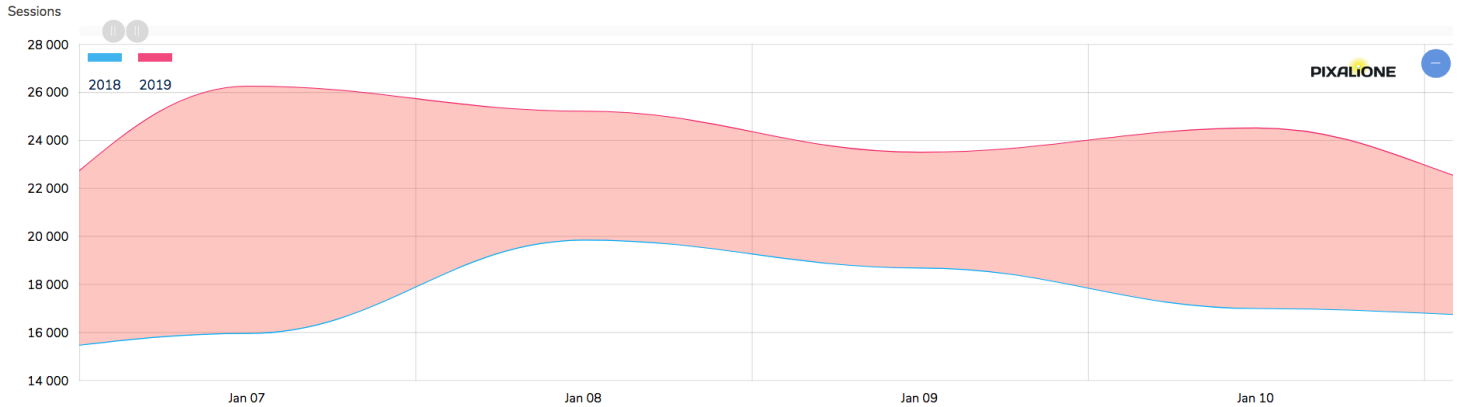
```
return $app['twig']->render('/StatsSites/site.html.twig', array('current_route' => $request->get('key: '_currentRoute'), 'currentYear' => $currentYear, 'diffDayInsert' => $arrayDiffDay, 'listeRsRangOneYear' => $listeRsRangOneYear,
```

Un script js utilise ses données pour les insérer en forme de tableau dans le diagramme.

Pour manipuler les graphiques, j'utilise la librairie Amcharts qui, avec un couple de tableau de données permet d'afficher plusieurs graphiques.

Avec cette librairie, on affiche donc deux tableaux de données sous formes de graphes comme ci-dessous :

Statistiques quotidiennes de fréquentation du site



Pour des raisons de confidentialités, je n'ai affiché sur le graphe seulement le nombre de sessions sur quelques jours.

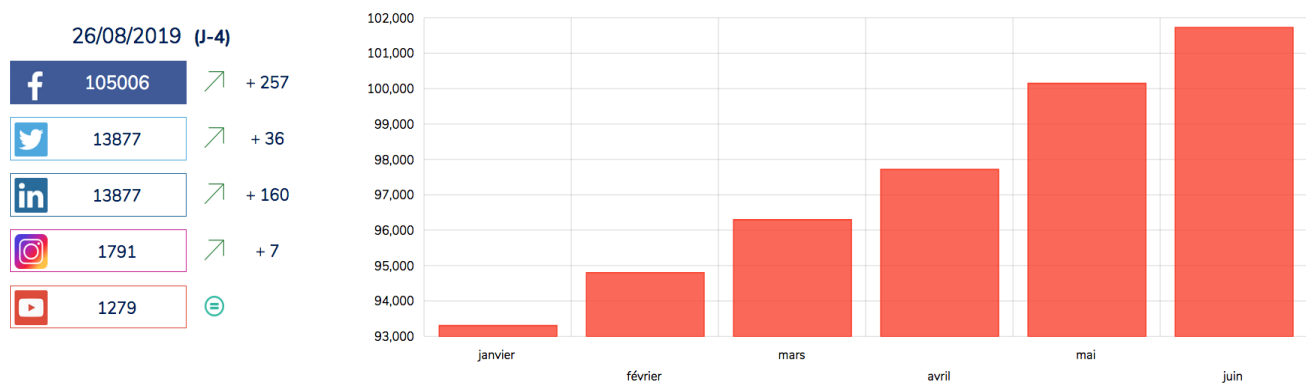
La suite de la page est consacrée aux données des réseaux sociaux.

Un menu est positionné à la gauche de la page, il contient les différentes plateformes sociales propre à l'entreprise choisies (ici Citya Immobilier).

L'utilisateur peut sélectionner son réseau, pour chaque sélection, le graphe placé à la droite du menu est actualisé et affiche les données sous formes de colonnes.

Ses données correspondent au nombre d'abonnés ou de followers du compte de l'entreprise.

Réseaux sociaux 2019



26/08/2019 (J-4)

f	105006	↗ + 257
t	13877	↗ + 36
in	13877	↗ + 160
ig	1791	↗ + 7
yt	1279	↔

LA RÉALISATIONS DE L'APPLICATION :

LES OBJECTIFS DE L'APPLICATION

Citya Immobilier, première entreprise indépendante d'administration de biens français utilise de nombreux outils numériques pour agrandir son cercle de vente et sa notoriété afin de prospérer dans le domaine de l'administration de biens.

Parmi ces outils, il existe une application mobile gratuite sous IOS et Android servant d'interface entre les clients et Citya Immobilier. Elle est composée d'une douzaine de pages complétant différentes fonctionnalités.

La dernière mise à jour de l'application du côté Android et IOS date de début septembre 2018. Cette solution devenant vieillissante et comportant quelques défauts d'affichage ; de plus elle n'est pas à jour des dernières techniques de conception et de réalisation d'application mobile, l'expérience utilisateur ainsi que l'interface utilisateur sont devenues primordiales pour une application mobile. Une refonte de l'application est nécessaire pour optimiser son utilisation. De surcroît, un autre point important amène à la refonte de l'application : L'ancienne version ayant été réalisée par une agence extérieure.

En effet, elle a été réalisée par une agence de développement externe répondant à un cahier des charges fourni. Nous n'avons donc plus la possibilité de la modifier, de l'utiliser ou de développer des fonctionnalités supplémentaires comme nous le voulons.

L'application créée en repartant de rien est une nouvelle version de l'application Citya Immobilier et utilise une charte graphique différente ainsi qu'une ergonomie plus adaptée, une utilisation plus poussée de l'UX (l'expérience utilisateur) et de l'UI (l'interface utilisateur) design dans l'application. L'ancienne version était une application d'annonces de biens qui listait sous plusieurs format (carte ou liste) l'ensemble de biens immobiliers gérés par Citya Immobilier selon des critères définis par l'utilisateur, qui affiche des agences Citya Immobilier sur une carte ou encore à l'aide de la réalité augmentée affiche les biens autour du client. Les profils des utilisateurs de l'application mobile sont similaires à ceux des clients de Citya Immobilier.

Nous visons un public non-professionnel qui souhaite accéder à des offres de biens immobiliers. La moyenne d'âge varie de l'étudiant/jeune travailleur au retraité, tous ayant des critères de sélection différents et adaptés à leurs besoins.

Pour les contraintes techniques :

- La maintenance devait pouvoir se faire facilement par le pôle Web de l'agence Reference.
- Des systèmes de traqueurs devait être mis en place afin d'avoir un retour sur l'utilisabilité de l'application.
- La plateforme utilisée pour développer l'application est Xamarin, cette plateforme permet de créer des applications Native IOS et Android avec une seule version de l'application.

Cette solution était la plus adaptée à notre solution, les fonctionnalités demandées pouvait être réalisées telles que la gestion des droits utilisateurs, la Carte, la gestion des annonces et permettre une connexion hors ligne.

- Une des contraintes techniques majeures, était le nombre de ressources humaines, un développeur.

CONCEPTION DE L'APPLICATION

Le commencement du projet fut l'élaboration d'un cahier des charges définissant le périmètre du projet, la charte graphique, le wireframe et le maquettage, Contraintes techniques, Les livrables ainsi que le planning.

Le type de l'application n'a pas dérivé de son ancienne version, elle reste une application d'annonces de biens immobilier qui permet à l'utilisateur d'effectuer des recherches précises.

L'application avait comme contrainte, être compatible avec Apple et Android sur les supports mobiles et tablette.

Elle est opérationnelle et adaptés pour des mises à jour à venir sur l'application.

Pour la charte graphique, L'identité de l'application sera celle de l'entreprise Citya Immobilier, le code couleur sera le même, reprenant les couleurs principales de Citya immobilier. L'application reprend les mêmes codes graphiques que le site web Citya.com, afin de garder une cohérence.

Même chose pour l'ensemble des logos et images dans l'application, ces derniers sont liés à l'identité de Citya Immobilier et validés par l'agence de communication Reference.

Des effets visuels dynamiques sont à mettre en place pour rendre l'application plus conviviale et finie. Le style de l'interface sera « user friendly » c'est-à-dire simple et facile d'utilisation avec comme principales caractéristiques :

- a. L'accessibilité à tous les utilisateurs,
- b. La compatibilité avec tous les supports mobiles,
- c. L'intuitivité de la navigation,
- d. Le traitement des erreurs,
- e. Les contrastes des couleurs,
- f. L'utilisation de formes pour délimiter les contenus.

Le flat design devra être le style graphique prépondérant de l'application.

CREATION DE L'APPLICATION

Base de l'application :

Le fichier App.xaml est le point de départ des déclarations de chaque application xamarin. Visual Studio le crée automatiquement quand on commence une nouvelle application.

L'une des fonctionnalités les plus utilisées du fichier App.xaml est de définir des ressources globales pouvant être utilisées et accessibles depuis l'ensemble d'une application, par exemple des styles globaux.

Xamarin crée aussi le fichier Code-behind appelé App.xaml.cs. Ils fonctionnent un peu comme une fenêtre, où les deux fichiers sont des classes partielles, coopérant pour permettre de travailler à la fois sur le balisage (XAML) et le code-behind.

App.xaml.cs étend la classe Application, qui est une classe centrale dans une application Windows WPF. Le .NET Framework ira à cette classe pour démarrer les instructions et ensuite démarrer la fenêtre voulue ou la page désirée à partir de là. C'est aussi l'endroit pour souscrire à des événements d'application importants, tels que le démarrage de l'application, l'instanciation de certains plugins, les exceptions non gérées...

Les Paquets :

Les « packages NuGet » contiennent le code compilé (sous forme de DLL) ainsi que tout le contenu nécessaire pour les projets qui les utilisent.

NuGet est le gestionnaire de packages le plus répandu pour le développement .NET, et il est intégré à Visual Studio pour Mac et à Visual Studio pour Windows. On peut rechercher des packages et les ajouter à nos projets Xamarin.

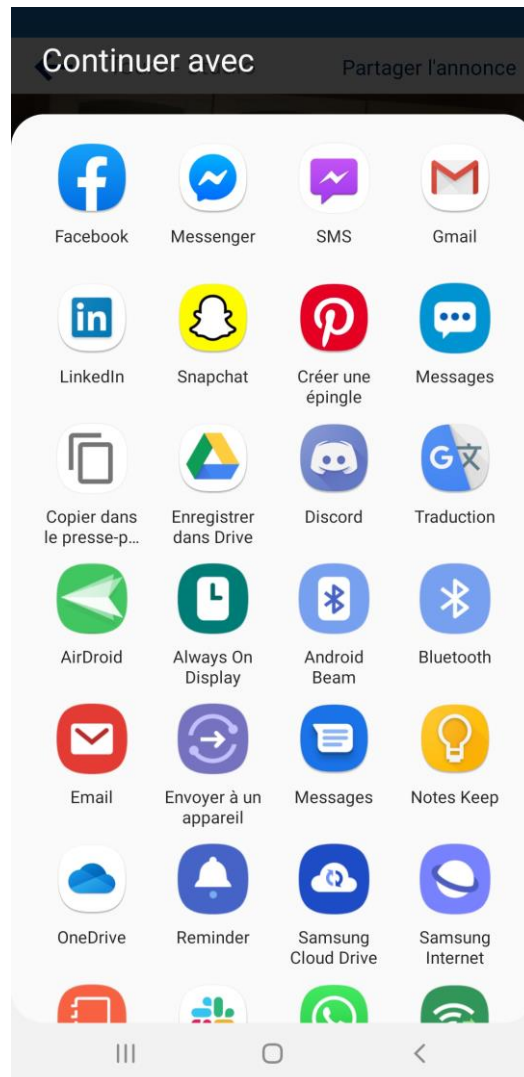
Plusieurs paquets sont utilisés dans le projet Citya Immobilier :

- Xamarin.Forms.Map
- Xamarin.Firebase.Core
- Xamarin.Firebase.Messaging
- Xamarin.FFImageLoading
- Plugin.Share
- Plugin.Toast
- Xamarin.Plugin.AnimationNavigationPage
- Xam.Plugins.Settings
- Xam.Plugin.Geolocator

Ces paquets sont initialisés dans les fichiers de lancement de l'application (ici MainActivity.cs pour android et AppDelegate.cs pour iOS).

Certains paquets sont initialisés dans le fichier App.Xaml.cs.

Ces paquets permettent d'avoir accès à certaines fonctionnalités sans les coder nous-mêmes, par exemple, le paquet Plugin.Share permet de rapidement implémenter la fonctionnalité de partage :



Le fait de pouvoir partager une annonce est une bonne fonctionnalité, cela permet d'apporter une bonne plus-value à l'application.

Custom Renderer :

Les interfaces utilisateur Xamarin.Forms sont affichées à l'aide des contrôles natifs de la plateforme cible, ce qui permet aux applications Xamarin.Forms de conserver l'apparence appropriée pour chaque plateforme. Les convertisseurs personnalisés permettent aux développeurs de remplacer ce processus pour personnaliser l'apparence et le comportement des contrôles Xamarin.Forms sur chaque plateforme. Le changement de l'apparence d'un contrôle Xamarin.Forms, sans renderer personnalisé, est un processus en deux étapes qui implique la création d'un contrôle personnalisé par l'intermédiaire d'une sous-classe, puis la consommation du contrôle personnalisé à la place du contrôle d'origine.

Les custom renderers dans l'application :

- CustomTabbedRenderer
- MyEntryRenderer

CustomTabbedRenderer permet la customisation de l'affichage de la page TabbedPage. Le TabbedPage de Xamarin.Forms se compose d'une liste d'onglets et d'une zone de détails plus grande, chaque onglet chargeant du contenu dans la zone de détails. Les icons sont

de base affichée en glyph (une seule couleur) Pour intégrer des icônes multicolores il fallait utiliser un customRenderer

Le deuxième customRenderer : MyEntryRenderer Ce custom renderer change les propriétés du type Entry de xamarin.forms Xamarin.Forms Entry est utilisé pour l'entrée de texte à ligne unique. Le Entry, comme l'Editor afficher, prend en charge plusieurs types de clavier. En outre, le Entry peut être utilisé comme un champ de mot de passe. Ce custom renderer modifie la couleur de fond, ajoute des effets d'ombre, rend les bords de l'entrée carrés ou les bords courbables. Contrairement au CustomTabbedRenderer qui remplaçait par default, celui-ci est utilisable. On peut appeler le custom renderer de cette façon

```
<local:MyEntry CornerRadius="1" BackgroundColor="White" IsTextPredictionEnabled="false">
  <Entry.Keyboard>
    <Keyboard x:FactoryMethod="Create">
      <x:Arguments>
        <KeyboardFlags>
          Suggestions, CapitalizeWord
        </KeyboardFlags>
      </x:Arguments>
    </Keyboard>
  </Entry.Keyboard>
</local:MyEntry>
```

On remarque l'accès au propriété CornerRadius="1" BackgroundColor="White"

Les permissions :

Les permissions sont nécessaires au bon fonctionnement de l'application. Sans ces permissions, certaines fonctionnalités comme la géolocalisation ou l'affichage de notifications à distance ne peuvent pas être utilisées.

Sur iOS, la gestion des droits s'administre dans le fichier info.plist, il contient aussi l'accès aux autorisation à la caméra, au microphone, aux appareils Bluetooth de l'appareil.

Pour Android, la gestion des autorisations se déclare dans le fichier AndroidManifest.xml, pour demander ses autorisations à l'utilisateur, une fonction dans le fichier MainActivity.cs récupère toutes les autorisations nécessaires au bon fonctionnement de l'application et les soumet à l'utilisateur.

La classe Utils.cs permet la requête des permissions nécessaires. Cette classe utilise le plugin : Plugin.PushNotification.

Au début de l'application, on appelle la méthode CheckPermissionStatusAsync contenu dans cette classe, avec l'id de la permission souhaitée. Pour chaque plateforme, la méthode affiche la demande de droit à l'utilisateur.

Selon l'état de la réponse, on utilise les méthodes de géolocalisation pour récupérer la ville dans laquelle se trouve l'utilisateur.

La navigation :

Xamarin.Forms propose plusieurs expériences différentes de navigation dans les pages, selon le type de page utilisée.

La classe `NavigationPage` propose une expérience de navigation hiérarchique où l'utilisateur est en mesure de parcourir les pages, vers l'avant et vers l'arrière, comme il le souhaite (push et pop).

Pour passer d'une page à une autre, une application envoie une nouvelle page dans la pile de navigation, où elle devient la page active.

Pour revenir à la page précédente, l'application dépile la page actuelle, et la nouvelle page tout en haut devient la page active.

Le `TabbedPage` de `Xamarin.Forms` se compose d'une liste d'onglets et d'une zone de détails plus grande, chaque onglet chargeant du contenu dans la zone de détails. Cet article montre comment utiliser un `TabbedPage` pour parcourir une collection de pages. C'est ce format que j'ai choisi pour l'application `Citya IMMOBILIER`.

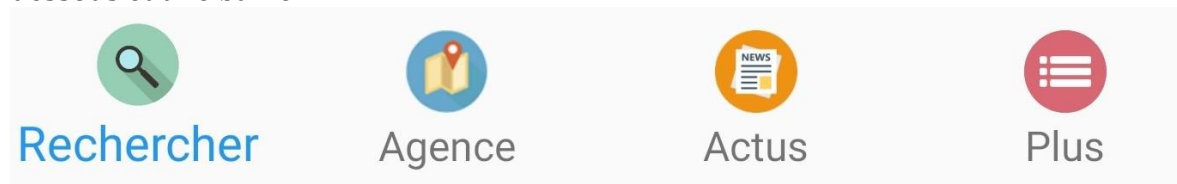
La disposition d'un `TabbedPage`, et de ses onglets, dépend de la plateforme :

- Sur iOS, la liste des onglets apparaît au bas de l'écran et la zone de détails apparaît au-dessus. Chaque onglet a également une image d'icône qui doit être au format PNG 30x30
- Sur Android, la liste des onglets apparaît en haut de l'écran par défaut, et la zone de détails est au-dessous. Toutefois, la liste d'onglets peut être déplacée vers le bas de l'écran à l'aide de paramètres propres à la plateforme.

Par défaut, les icônes des onglets sont en mode "Glyph" c'est-à-dire qu'ils sont affichés en une couleur.

Voulant afficher les icônes avec leurs couleurs d'origine, j'ai dû utiliser un custom renderer.

Ce custom renderer est intitulé `CustomTabbedPageRenderer`, pour chaque plateforme, le renderer positionner des icônes avec leurs images d'origine, avec le titre de la page en dessous et une barre :



Pour revenir à la Navigation hiérarchique, lors de l'apparition d'une page, aucune animation de navigation n'est employée. J'ai utilisé un plugin

(`Plugin.AnimationNavigationPage`) qui permet d'ajouter une animation lors de l'apparition d'une page. Pour se faire, on implémente la page avec l'interface `IAnimationPage` (appartenant au plugin) puis on déclare une balise dans la page pour spécifier quelle transition nous aimerions utiliser.

Les différentes transitions sont : Depuis la droite, la gauche, le bas, le haut, en rotation. On peut changer la durée de l'animation : Courte, Moyenne, rapide, par défaut, aucune.

Les listes :

Un `ListView` est rempli avec les données à l'aide du `ItemsSource` propriété (comme vu ci-dessus il peut être rempli avec des objets récupérés depuis l'API, qui peut accepter toute collection implémentant `IEnumerable`). La façon la plus simple pour remplir un `ListView` est l'utilisation d'un tableau de chaînes comme objet. `ListView` présente les listes déroulantes, chaque cellule peut être utilisée pour afficher du texte et des images, indiquant un état `true/false` et recevoir des entrées d'utilisateur. Lorsque les cellules

intégrées ne fournissent pas la disposition requise, cellules personnalisées implémentent la disposition requise. Par exemple, vous souhaiteriez présenter une cellule avec deux étiquettes qui ont le même poids. Un TextCell serait insuffisant, car le TextCell a un nom qui est plus petit. La plupart des personnalisations de cellule ajoutent des données en lecture seule supplémentaires (par exemple, des étiquettes supplémentaires, images ou d'autres informations d'affichage).

Toutes les cellules personnalisées doivent dériver de ViewCell, la même classe de base que tous de la cellule intégrée types utilisent. Exemple :

```
<ListView x:Name="ItemsListView" Margin="0,-5,0,0"
    Grid.Row="1"
    IsPullToRefreshEnabled="true"
    ItemSelected="OnItemSelected"
    RowHeight="400"
    ItemsSource="{Binding Annonces}"
>
    <ListView.ItemTemplate>
        <DataTemplate>
            <ViewCell>
                <custom:AnnonceCellView BindingContext = "{Binding}" />
            </ViewCell>
        </DataTemplate>
    </ListView.ItemTemplate>
</ListView>
```

La cellule personnalisée est imbriquée dans une DataTemplate, ce qui se trouve dans ListView.ItemTemplate. Ceci est le même processus qu'à l'aide de n'importe quelle autre cellule. ViewCell est le type de la cellule personnalisée. L'enfant de la DataTemplate élément doit être un dériver du type ViewCell.

<custom:AnnonceCellView BindingContext = "{Binding}" />

AnnonceCellView est la vue contenue dans le fichier AnnonceCellView.xaml.cs, on lie à cet vue chaque objet de la liste ; Dans cette vue on récupère l'objet de cette façon:

```
protected override void OnBindingContextChanged()
{
    annonce = this.BindingContext as Annonce;
    if (annonce != null)
    {
```

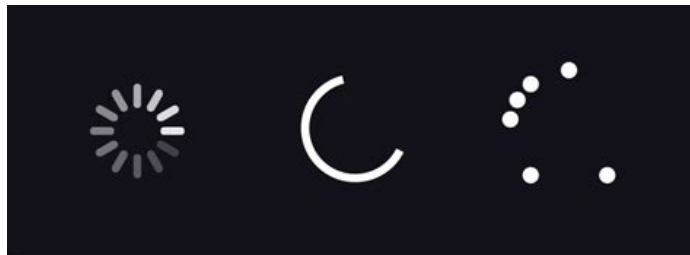
Une fois l'annonce récupérée, on peut attribuer ces propriétés à des éléments graphiques de la vue :

```
typeUsable.Text = annonce.type.FirstLetterMaj();
nbrePiece.Text = " (" + annonce.nbrePiecesFormater + ")";
ville.Text = annonce.ville;
NbrPhoto.Text = annonce.nbrePhotos;
prix.Text = annonce.prix.ToString() + "€";
String urlImage = App.IpAddressServeur + "/media/images/agences/biens/" + annonce.agence_id + "/"
photo.Source = urlImage;
```

Les listes Xamarin contiennent la propriété RefreshCommand, cette propriété est une commande qui permet de rendre la liste dynamique, avec la possibilité de la recharger lorsqu'on la "tire" (pull en anglais) vers le bas.

Ainsi on peut exécuter une action lors de ce mouvement.

Durant cette action, un indicateur d'action de rafraîchissement de la liste est affiché :



Souvent cette action sera une nouvelle requête de la liste afin de l'actualiser :

```
ItemsListView.RefreshCommand = new Command((obj) =>
{
    Data.DownloadDataClass.DownloadData(rechercheContext, async (annonces) =>
    {
```

Une fois l'action fini, on réaffiche la liste actualisée et on passe le booléen IsRefreshing de la liste à false, pour indiquer au programme que le rafraîchissement est fini.

La géolocalisation :

Au lancement de l'application, une fois les droits demandés et accordés, on peut récupérer la position de l'utilisateur en utilisant le paquet Plugin.Geolocator, si la position est accessible, alors on change un Boolean dans les pages Formulaire et Agence, deux pages principales de l'application, ces pages, font une requête à l'API avec la latitude et la longitude de la position du mobile. L'API retourne en réponse un nom de quartier, une ville ou le code Insee du département.

Avec ces résultats, s'en suit des requêtes différentes selon les pages, pour le formulaire, on récupère la ville et on l'ajoute dans le model de la recherche d'annonce, cela permet à l'utilisateur de ne pas être obligé de renseigner une ville et pouvoir directement lancer la recherche. Pour la page Agences, on effectue une requête de liste d'agences avec le code département, cette liste d'agences est directement affichée dans la carte, la carte est centrée sur l'agence la plus proche de l'utilisateur.

J'ai choisi de seulement récupérer par défaut la liste des agences dans le département pour alléger le résultat, pour rendre donc l'application plus rapide lors de l'affichage de cette page (plusieurs tests ont montré que l'affichage de toutes les agences de France est trop long). L'utilisateur peut afficher les agences de la France entière en cliquant sur un bouton au-dessus de la carte.

La carte :

Les cartes de l'application sous Android utilisent une clé API Maps, cette clé est administrable via la plateforme console cloud de Google.

Pour IOS, Apple met à disposition une carte par défaut.

Bien évidemment, pour l'utilisation de la géolocalisation et les Maps, des droits sont demandés.

On initialise ensuite les maps dans les fichiers principales de lancement de l'application MainActivity et AppDelegate: Xamarin.FormsMaps.Init();

Maintenant que la carte est initialisée, on peut maintenant l'utiliser, pour l'utiliser :

```
customMap = new Map
{
    IsShowingUser = true,
    HeightRequest = 100,
    WidthRequest = 960,
    VerticalOptions = LayoutOptions.FillAndExpand,
    HorizontalOptions = LayoutOptions.FillAndExpand
};
```

On déclare dans le fichier xaml.cs l'objet Map avec certaines propriétés, ici l'affichage de l'utilisateur, la largeur et la hauteur.

Différents de type de mappage sont à disposition, le contenu de la carte peut être modifié en définissant la propriété MapType, on peut afficher une carte classique (la valeur par défaut), une carte en images satellite ou une combinaison des deux (dit hybride).

Sur les cartes on peut ajouter des épingles, par exemple des épingles représentant les biens situés sur une ville.

Le code suivant permet de placer une épingle à une position voulu, avec un label et des détails :

```
var positionAnnonce = new Position(annonce.latitudeAnnonce, annonce.longitudeAnnonce);
var pin = new CustomPin
{
    Type = PinType.Place,
    Position = positionAnnonce,
    Label = annonce.type.FirstLetterMaj() + ": " + annonce.prix + "€ - " + annonce.surfaceBien + "m²",
    Address = "(cliquez ici pour plus de détails) "
};

pin.Clicked += async (object sender, EventArgs e) =>
{
    await Navigation.PushModalAsync(new ItemDetailPage(annonce.id));
};
```

Ici on définit la position de l'épingles avec la position de l'annonce, ensuite on paramètre l'épingle avec les valeurs souhaitées, comme le titre de l'épingles ou l'affichage complémentaire de l'adresse.

Pin.Clicked permet de définir une action lors du click d'une épingle, ici ouvrir la page ItemDetailPage en passant en paramètre l'id de l'annonce.

Pour ajouter une épingle sur une carte, on accède à la propriété liste "Pins" de la carte, puis comme pour une simple liste, on ajoute l'épingle grâce à la méthode Add:

```
customMap.Pins.Add(pin);
```

En fournissant une instance MapsSpan à un constructeur de la carte, on définit la vue initiale (point central et un niveau de zoom) de la carte lorsqu'il est chargé.

La méthode statique MapSpan.FromCenterAndRadius() permet de créer une étendue d'une "position" et en spécifiant une "distance". Ainsi on peut centrer la carte sur un point précis ou sur l'utilisateur selon une distance voulue.

Pour centrer la carte sur un ensemble d'épingles, le développement est légèrement plus complexe :

J'ai tout d'abord créé la fonction coverAllMarker, qui sera appelée lors du chargement de la carte. Dans cette fonction, on passe en argument la liste d'épingles, pour chaque épingle, on récupère la latitude et la longitude, on les place ensuite dans deux listes

distinctes, la liste des latitudes et la liste des longitudes, pour chaque liste je récupère la valeur la plus petite et la valeur la plus grande.

Ensuite j'appelle une fonction de la classe GeoCodeCalc (classe de calcul de distance trouvée sur des forums de xamarins) avec en paramètre les 4 valeurs trouvées avant.

Cette fonction me retourne une distance.

Pour centrer la map, il faut une distance et un centre, la latitude du centre est la moyenne de la plus grande latitude avec la plus petite latitude, le tout divisé par deux. Même méthode pour la longitude, ainsi nous pouvons utiliser la méthode statique MapSpan.FromCenterAndRadius() avec comme paramètre notre distance et notre centre.

Maintenant nous allons voir comment l'application communique avec l'API.

Les Tokens :

L'accès à l'API nécessite des tokens, pour chaque requête, un token est demandé.

Les tokens sont demandés à cette adresse :

<https://www.citya.com/authentication token>

Lors du lancement de l'application, ou lors d'un changement d'état de connexion de l'appareil, on demande l'obtention d'un token au serveur.

On passe dans la requête POST le contenu suivant :

```
{
  "username": "userApplicationCitya",
  "password": "voir KeyPass"
}
```

Cet identifiant est stocké dans la BDD Citya.

Le serveur envoie en réponse le token sous forme de Json.

Pour chaque requête vers l'API, le token est inséré dans l'entête ;

Exemple au début d'une requête :

```
webclient.Headers["Authorization"] = "Bearer " + Application.Current.Properties["token"].ToString();
webclient.DownloadStringCompleted += (object sender, DownloadStringCompletedEventArgs e) =>
```

On intègre le token dans l'entête de la requête.

L'API :

L'adresse de l'api : <https://www.citya.com/api/> (Retourne du Json)

Pour envoyer des requêtes, on utilise les actions dans le fichier DownloadDataClass.cs

Ce fichier comporte toutes les actions qui interagissent avec l'API

Pour chaque action (L'adresse du serveur est contenu dans la propriété IPAddressServer du fichier App.Xaml.cs) :

```

string URL = App.IpAddressServeur + "URL";
// on instancie un WebClient (package System.Net.Http)
using (var webclient = new WebClient())
{
    webclient.DownloadStringCompleted += (object sender, DownloadStringCompletedEventArgs e) =>
    {
        try
        {
            // e.Result correspond à la reponse du serveur (Json)
            string resultJson = e.Result;
            String codeInsee = resultJson.ToString();
            Device.BeginInvokeOnMainThread(() =>
            {
                // pour finir on invoke le resultat (List, object, string..)
                action.Invoke(codeInsee);
            });
        }
        catch (Exception ex)
        {
            // en cas d'echec
            Device.BeginInvokeOnMainThread(() =>
            {
                action.Invoke(null);
            });
        }
    };
    webclient.DownloadStringAsync(new Uri(URL));
}

```

Exemple d'utilisation d'une de ces actions :

```

Data.DownloadDataClass.DownloadDataAgences(codeDepartement, (agences) =>
{
    if (agences != null)
    {
        ItemsListView.ItemsSource = agencesCitya;
        isLoading.IsVisible = false;
        ItemsListView.IsVisible = true;
    }
});

```

On passe en paramètre un code de département, qui sera utilisé dans la requête. Ici, on nourrit la liste ItemsListView avec la liste d'objet récupéré par l'action. On cache ensuite le gif de chargement et on affiche la liste remplie.

Firestore Analytics :

Firestore Analytics est une solution gratuite de mesure d'applications qui fournit des informations sur l'utilisation des applications et l'engagement de l'utilisateur.

Firestore affiche sous forme de graphes, le nombre de connexions et d'utilisateur actifs par jours.

L'envoi d'événements personnalisé est intégré dans le projet.

Une interface est implémentée pour Android et iOS, EventTrackerDroid et EventTrackerIOS, ces deux interfaces possèdent 3 méthodes qui envoient des événements à Firestore avec différentes données (string, triple string, dictionnaire).

Pour utiliser cette interface pour envoyer des événements à Firestore :

On déclare l'interface au début d'une classe .xaml.cs :

```
readonly IEventTracker eventTracker;
```

Puis on l'initialise lors de la création de la page :

```
eventTracker = DependencyService.Get<IEventTracker>();
```


Ensuite on peut appeler la fonction SendEvent :
 eventTracker.SendEvent(“titre”, “Recherche avec details” , “détails”);
 Firebase recevra les événements et affichera les détails, on peut ainsi connaître quelles recherches effectue les utilisateurs ou quelles pages sont visualisées.

Firestore Cloud Messaging :

Pour l'utilisation de cloud messaging avec une application Xamarin.Forms, son déploiement est plus complexe et nécessite l'utilisation d'un plugin :
 Plugin.PushNotification

Ce plugin permet de faciliter la réception de notifications à distance l'initialisation pour chaque plateforme est légèrement différentes : voir

<https://github.com/CrossGeeks/PushNotificationPlugin/blob/master/docs/GettingStarted.md>

Le message de la notification est récupéré dans le fichier App.xaml.cs grâce à la méthode : CrossPushNotification.Current.OnNotificationOpened, cette méthode passe en paramètre un objet qui correspond à la notification.

L'Annexe 1 correspond à une présentation des pages de l'application.

PUBLICATION SUR LES STORES

Avant la publication sur les stores, j'ai procédé à une phase d'optimisation et de test de l'application.

Pour l'optimisation, la documentation Microsoft présente sous formes de catégories, différent moyen pour optimiser son application, réduire sa taille et améliorer ses performances. Je vais maintenant vous présenter les moyens que j'ai pu mettre en place sur l'application.

1) Activer le compilateur XAML :

XAML peut être éventuellement compilé directement en langage intermédiaire (IL) avec le compilateur XAML (XAMLC). XAMLC offre un certain nombre d'avantages :

- Il effectue une vérification au moment de la compilation du code XAML et informe l'utilisateur des erreurs rencontrées.
- Il supprime une partie du temps de chargement et d'instanciation pour les éléments XAML.
- Il permet de réduire la taille de fichier de l'assembly final en n'incluant plus les fichiers .xaml.

Il est activé au niveau de l'assembly en ajoutant l'attribut XamlCompilation.

Ainsi ce préfixe s'applique à la totalité de l'assembly.

2) Déployer la compression des dispositions :

La compression des dispositions supprime les dispositions spécifiées à partir de l'arborescence visuelle dans le but d'améliorer les performances de rendu de page.

Le résultat du processus de mise en page est une hiérarchie qui inclue des convertisseurs de conteneur supplémentaire et des wrappers pour les convertisseurs de plateforme, gonfler davantage la hiérarchie d’affichage d’imbrication.

Plus simplement, la hiérarchie d’affichage imbriquée contient des vues inutiles qui peuvent être supprimées en compressant la disposition.

Dans XAML, la compression des dispositions peut être activée en définissant le **CompressedLayout.IsHeadless** propriété (true) sur une classe de disposition (comme un `stacklayout` ou un `absolutelayout`).

3) Réduire la taille du dictionnaire de ressources de l’application :

Toutes les ressources qui sont utilisées dans l’application doivent être stockées dans le dictionnaire de ressources de l’application pour éviter la duplication. Cela permet de réduire la quantité de code XAML qui doit être analysé dans toute l’application.

Toutefois, le code XAML qui est spécifique à une page ne doit pas être inclus dans le dictionnaire de ressources de l’application, car les ressources sont alors analysées au démarrage de l’application et non lorsqu’une page le demande. Si une ressource est utilisée par une page qui n’est pas la page de démarrage, elle doit être placée dans le dictionnaire de ressources pour cette page, ce qui contribue à réduire le code XAML qui est analysé quand l’application démarre. Dans mon projet, certains styles présents sur une page était déclaré dans les ressources de l’application, j’ai donc créé des dictionnaires de ressources dans les pages spécifiques.

Plusieurs autres moyens d’optimisation auraient pu être mis en place pour optimiser l’application, mais fautes de temps et d’anticipation je n’ai pas pu les intégrer au projet.

Créer l’application en interpréteur de commandes aurait été un bon moyen de réduire le temps de démarrage de l’application. Le format d’application “`TabbedPage`” (4 pages principales) est un format long au démarrage, l’application attend de charger entièrement les pages principales ainsi que certaines méthodes comme la géolocalisation et la récupération de la ville où se situe l’utilisateur.

En utilisant `Xamarin.Forms Shell`, le temps de démarrage est réduit, Les applications `Shell` peuvent utiliser un modèle de navigation basée sur des URI dans lequel des itinéraires permettent d’accéder à n’importe quelle page dans l’application, sans avoir à suivre une hiérarchie de navigation définie.

Au démarrage du projet, ce type d’application était en beta est pouvait être complexe pour un néophyte de `Xamarin`.

Un deuxième point intéressant pour optimiser l’application est : “Les liaisons compilées” Les liaisons compilées sont résolues plus rapidement que les liaisons classiques. Elles améliorent ainsi les performances des liaisons de données dans les applications `Xamarin.Forms`.

Le processus pour mettre en place les liaisons compilées est le suivant :

Définir un attribut `x:DataType` sur un objet `VisualElement` correspondant au type de l’objet auquel `VisualElement` et ses enfants seront liés.

La compilation d’une expression de liaison génère du code compilé qui obtiendra une valeur d’une propriété de la *source* et la définira sur la propriété de la *cible* spécifiée dans le balisage. Ce processus est long et fastidieux à mettre en place, je n’ai pas eu le temps de l’intégrer dans le projet.

La mise en production de l'application n'a pas encore été effectué, la publication sur les stores nécessite l'accord de M.Chaminade. Au jours d'aujourd'hui, cet accord n'a pas encore été donné.

RÉSULTATS ET DIFFICULTÉS RENCONTRÉES

Durant cette alternance j'ai pu réaliser et terminer toutes mes tâches. Mes missions principales ont toutes abouties ainsi que mes missions supplémentaires. Le travail que j'ai réalisé tout au long de cette période m'a donné satisfaction, j'ai découvert comment fonctionne des outils du quotidien, comme l'affichage d'un grand nombre d'annonce, la gestion de page automatique pour chaque mandat ou encore la création d'un fichier Excel programmé.

Mes travaux vont être utilisés par un grand nombre de collaborateurs de l'entreprise et de clients finaux, il est agréable de savoir que ses résultats vont être exploités tous les jours.

Je me suis senti utile pour l'entreprise, j'ai participé au développement d'un projet important pour l'équipe et chaque mission qui m'a été attribué m'a demandé de me surpasser. Mon tuteur m'a fait confiance dans la réalisation de ces tâches et m'a permis d'être dans la plus grande autonomie possible.

Certaines tâches réalisées demandaient une rigueur exemplaire et chaque erreur pouvait être importante et donner mauvaise réputation à l'agence REFERENCE. Il m'a fallu développer une attitude professionnelle et bienveillante durant la réalisation de mes missions.

Pour les difficultés rencontrées, bien évidemment, il y a eu l'apprentissage et l'utilisation de Xamarin, qui est très complexe, mes bases et connaissances en développement mobile cross-plateforme étant très faibles par rapport à la demande.

La gestion du projet fut assez emmêlée, Mon planning était peu précis sur la réalisation des tâches et des fonctionnalités, ne connaissant pas Xamarin, je me suis quelque fois compliqué la tâche.

J'ai rencontré quelques difficultés lors de l'ajout de la géolocalisation au sein de l'application ainsi qu'avec les permissions. L'intégration des permissions dans Xamarin est assez éloignée de l'utilisation des permissions dans un développement natif.

Pour finir, lors des tests sur des vrais mobiles iOS, l'application crashait sans log, j'ai dû réaliser un grand nombre de test afin de cerner l'erreur (Un plugin n'était pas entièrement compatible sous iOS).

CONCLUSION

Pour la problématique, l'application n'étant pas encore publiée sur les différents stores, nous ne pouvons pas nous avancer sur son efficacité. L'ajout de la possibilité de partager une annonce ainsi que l'intégration des notifications sont des bonnes améliorations de l'application et pourront améliorer son intérêt.

BILAN PROFESSIONNEL

Cette alternance m'a permis de me faire une idée du travail en équipe dans un cadre professionnel ainsi que la réalisation d'un projet sur plusieurs mois, j'ai ainsi pu découvrir le travail et le quotidien d'un développeur Web/Mobile dans une agence dynamique.

Durant cette alternance j'ai pu utiliser certaines compétences acquises durant ma Licence professionnelle, mais j'ai dû renforcer mes connaissances avec beaucoup de travail et d'apprentissage supplémentaire, l'aide de l'équipe de développeurs m'a été utile pour accélérer certains processus.

Mes missions ont été assez variées, elles m'ont permis de développer des compétences dans le domaine du développement web ainsi que dans le domaine du développement mobile. Ces compétences me permettront de continuer de travailler et d'approfondir mes connaissances dans ce domaine.

Cette alternance m'a permis de mieux appréhender les phases d'un projet concret. Il m'a permis de comprendre comment répondre au besoin d'un client.

J'ai donc développé des compétences plus professionnelles en matière de maîtrise de nouveau matériel ou encore de méthodologie. Ma méthode de travail est maintenant plus rigoureuse et adaptée au format du travail en entreprise.

BILAN PERSONNEL

Les différentes missions qui m'ont été confiées m'ont permis de devenir de plus en plus autonome, mais aussi de savoir poser des questions quand je suis dans une impasse.

Mon insertion dans l'équipe s'est véritablement bien déroulée, j'ai appris beaucoup de points importants sur le monde de l'entreprise, le rythme de travail à adopter et la communication avec l'équipe.

Je me suis rendu compte que travailler dans une entreprise pluridisciplinaire était fait pour moi, le mélange des différents domaines et très bénéfique pour le travail, comme la synergie entre le pôle graphique et le pôle web.

Cette alternance aura également été une très bonne passerelle pour la poursuite de ma carrière et me conforte dans mon choix de mon avenir professionnel.

Pour conclure, cette alternance a été une expérience professionnelle significative, car elle m'a permis de choisir la voie dans laquelle je voulais me spécialiser.

Ma formation, bien que dans un domaine légèrement différent, m'a permis de réaliser cette alternance sereinement, la base théorique et technique apportée durant l'année, m'a permis de prendre du recul sur chaque mission réalisée.

BIBLIOGRAPHIE

La Documentation de Xamarin réalisé par Microsoft, pour Xamarin.Forms, elle est divisée en plusieurs parties, Notions de base des applications, interface utilisateur, Déploiement et Test...

Url : <https://docs.microsoft.com/fr-fr/xamarin/xamarin-forms/index>

Un article sur Firebase Analytics :

<https://medium.com/firebase-developers/firebase-analytics-on-xamarin-forms-c60ae8bbb24a>

Les gits des différents Plugins :

<https://github.com/jguertl/SharePlugin>

<https://github.com/xamarin/xamarin-forms-samples/blob/master/GetStarted/Notes/Database/Notes/Data/NoteDatabase.cs>

<https://github.com/luberda-molinet/FFImageLoading/tree/master/samples>

<https://github.com/alexrainman/CarouselView/blob/master/Demo/MainViewModel.cs>

Chacun de ces gits contient une doc et une présentation du plugin, ainsi qu'un exemple d'utilisation.

Le site référent de Symfony (avec la doc) :

<https://symfony.com/>

LEXIQUE

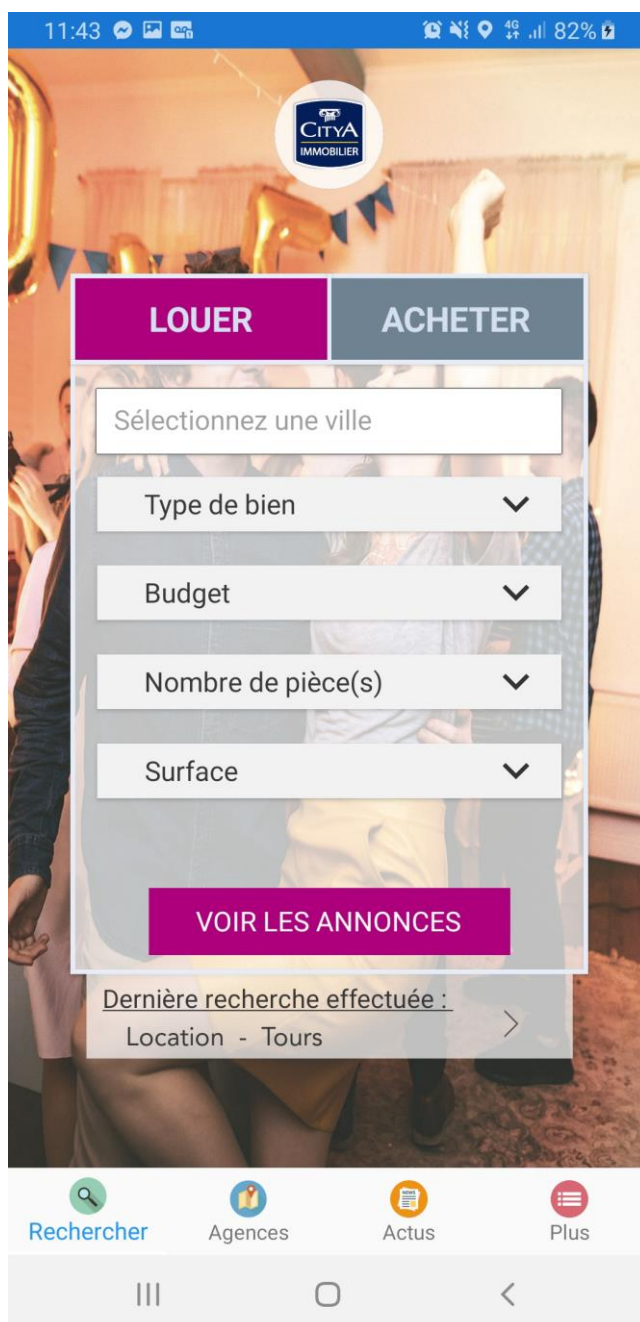
CONTROLLEURS	Les contrôleurs sont des fonctions généralement regroupées dans des classes par "thème" qui génèrent une réponse à partir d'une requête
Open source	Un logiciel Open source est un programme informatique dont le code source est distribué sous une licence permettant à quiconque de lire, modifier ou redistribuer ce logiciel.
Table	Une table est un ensemble de données organisées sous forme d'un tableau où les colonnes correspondent à des catégories d'information (une colonne peut stocker des numéros de téléphone, une autre des noms...) et les lignes à des enregistrements, également appelés entrées.
Intranet	Réseau informatique interne (à une entreprise, un organisme...), utilisant les techniques d'Internet.
Base de données	Une Base de données ensemble d'informations structurées accessibles au moyen d'un logiciel.
Site statique	Une page web statique est une page web dont le contenu ne varie pas en fonction des caractéristiques de la demande, c'est-à-dire qu'à un moment donné tous les internautes qui demandent la page reçoivent le même contenu.
Site dynamique	Une page web dynamique est une page web générée à la demande, par opposition à une page web statique. Le contenu d'une page web dynamique peut donc varier en fonction d'informations (heure, nom de l'utilisateur, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation.
Librairie	En informatique, une bibliothèque logicielle est

	une collection de routines, qui peuvent être déjà compilées et prêtes à être utilisées par des programmes ^{1,2} . Les bibliothèques sont enregistrées dans des fichiers semblables, voire identiques aux fichiers de programmes ³ , sous la forme d'une collection de fichiers de code objet rassemblés ² accompagnée d'un index permettant de retrouver facilement chaque routine
Débogage	Est le fait d'enlever les erreurs d'un programme, en suivant le déroulement de ce programme afin d'en repérer et corriger les dysfonctionnements
Requête SQL	Une requête SQL est un texte qui donne un ordre à exécuter à un moteur de base de données sur une base de données. Elle peut concerner un enregistrement ou plusieurs, elle peut mettre en jeu une table ou plusieurs. La requête est composée de mots clés, les commandes SQL
XAML	XAML est un langage déclaratif développé pour les besoins des systèmes d'exploitation de Microsoft, Windows Vista, Windows 7 et Windows 8 et qui permet la description de données structurées.

ANNEXE

ANNEXE 1 – L'APPLICATION :

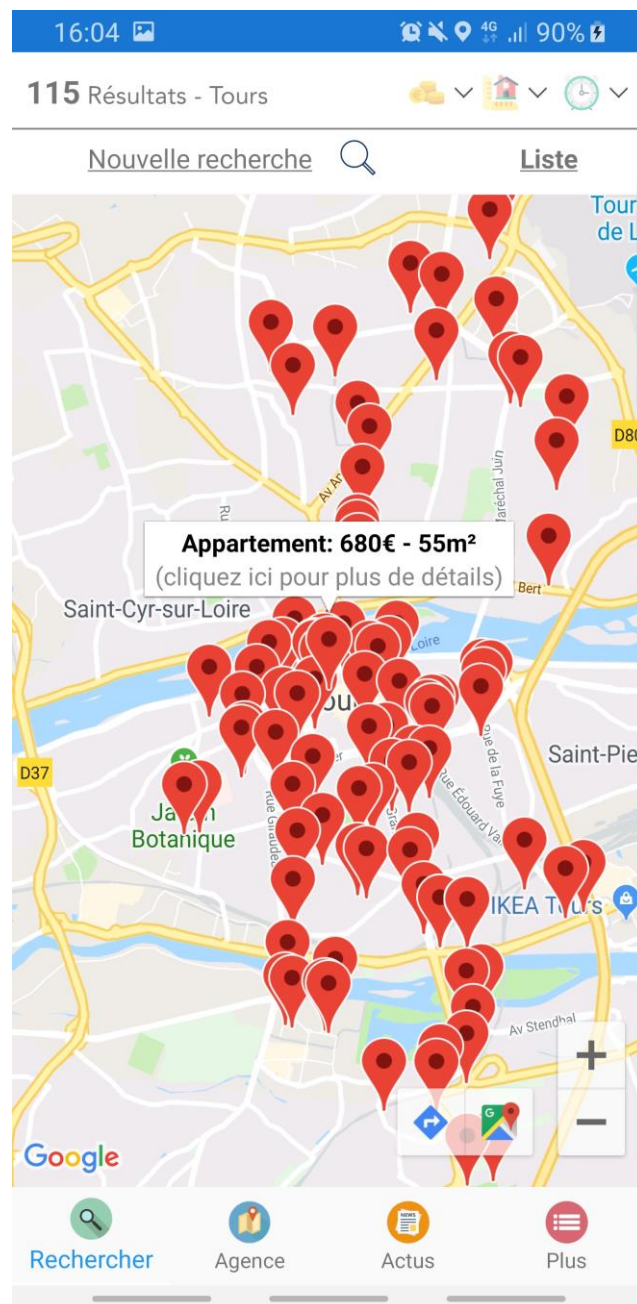
Page d'accueil :



Page résultats de recherche (affichage en mode liste) :



Page résultats de recherche (affichage en mode carte) :



Page détails d'un bien (Partie 1) :



Page détails d'un bien (Partie 2) :



Page détails d'un bien (Partie 3) :

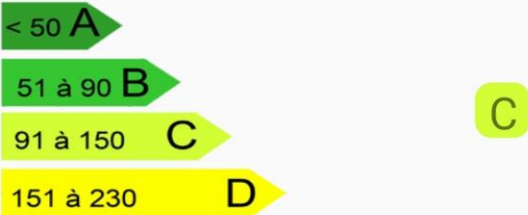
11:52 4G 83%

- + Mode chauffage: Electrique
- + Exposition Nord
- + Standing Grand standing
- + Chauffage: Individuel
- + Mécanisme chauffage: Radiateur
- + Appartement neuf
- + Exclusif

Détails sur le prix :

- + Libre le 28/07/2019 00:00:00
- + 559,5€ de loyer par mois charges com...
- + 445€ de loyer par mois hors charges
- + 445€ de dépôt de garantie
- + 297€ d'honoraires location charge loc...
- + 81€ d'honoraires d'état des lieux charg...
- + 0€ d'honoraires d'état des lieux charge...
- + 297€ d'honoraires total charge locataire
- + 297€ d'honoraires total charge proprié...

Mise en ligne: 08/07/2019 00:00:00
Référence: GES59990009-19



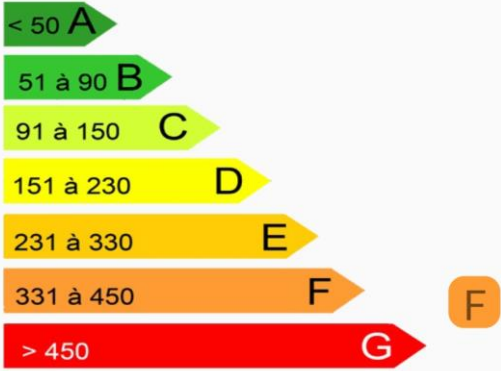
< 50 A
51 à 90 B
91 à 150 C
151 à 230 D

Appel Message

Page détails d'un bien (Partie 4) :

11:58 420€ d'honoraires total charge proprié...

Mise en ligne: 30/07/2019 00:00:00
Référence: GES00160006-32



The image shows a horizontal energy efficiency scale with seven color-coded arrows pointing right, each labeled with a letter and a range of values:

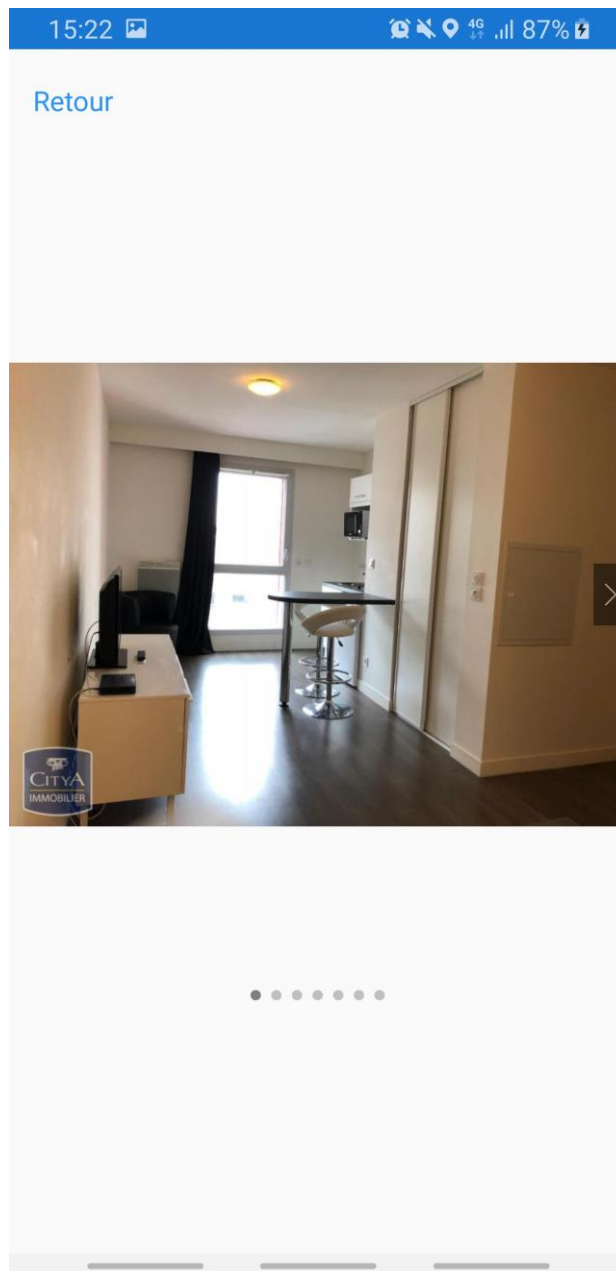
Label	Range
A	< 50
B	51 à 90
C	91 à 150
D	151 à 230
E	231 à 330
F	331 à 450
G	> 450

The scale is positioned above a map. The map shows a location in Tours, France, with a red pin marking the property. Landmarks include the Jardin Botanique, IKEA Tours, and the Saint-Pierre-des-Corps district. The map is provided by Google.

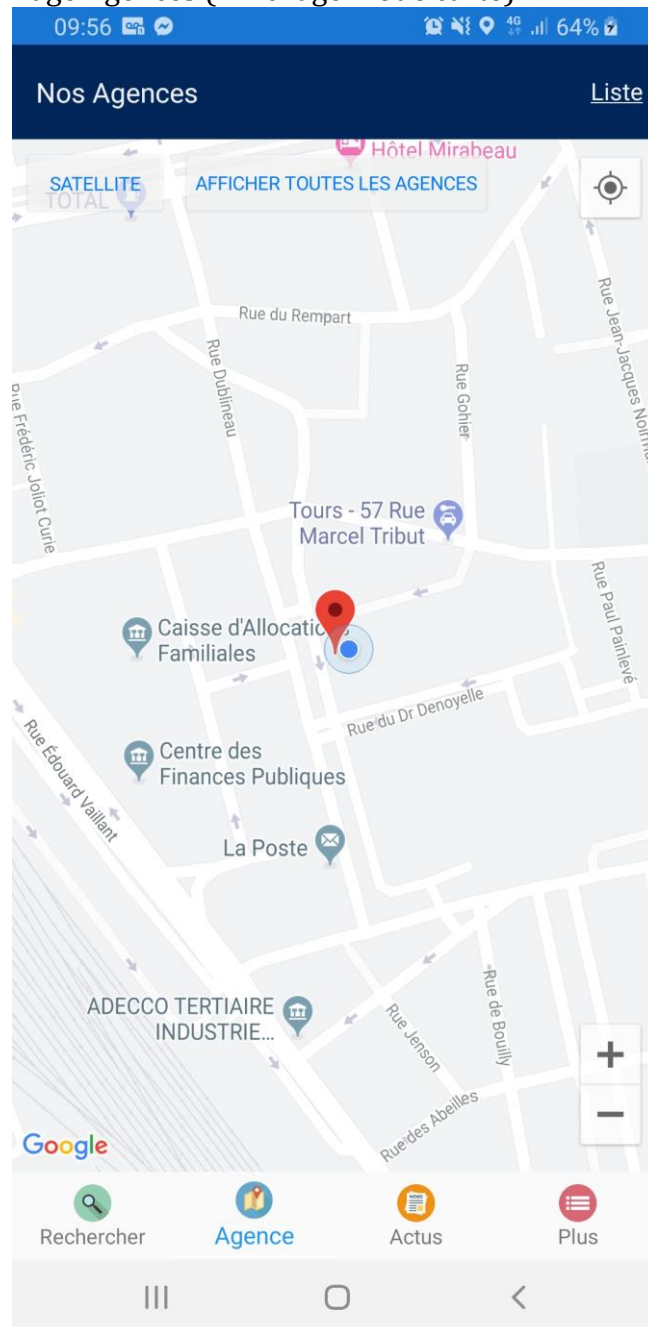
Géolocalisation du bien:

Appel Message

Page détails d'un bien (Partie 5) :



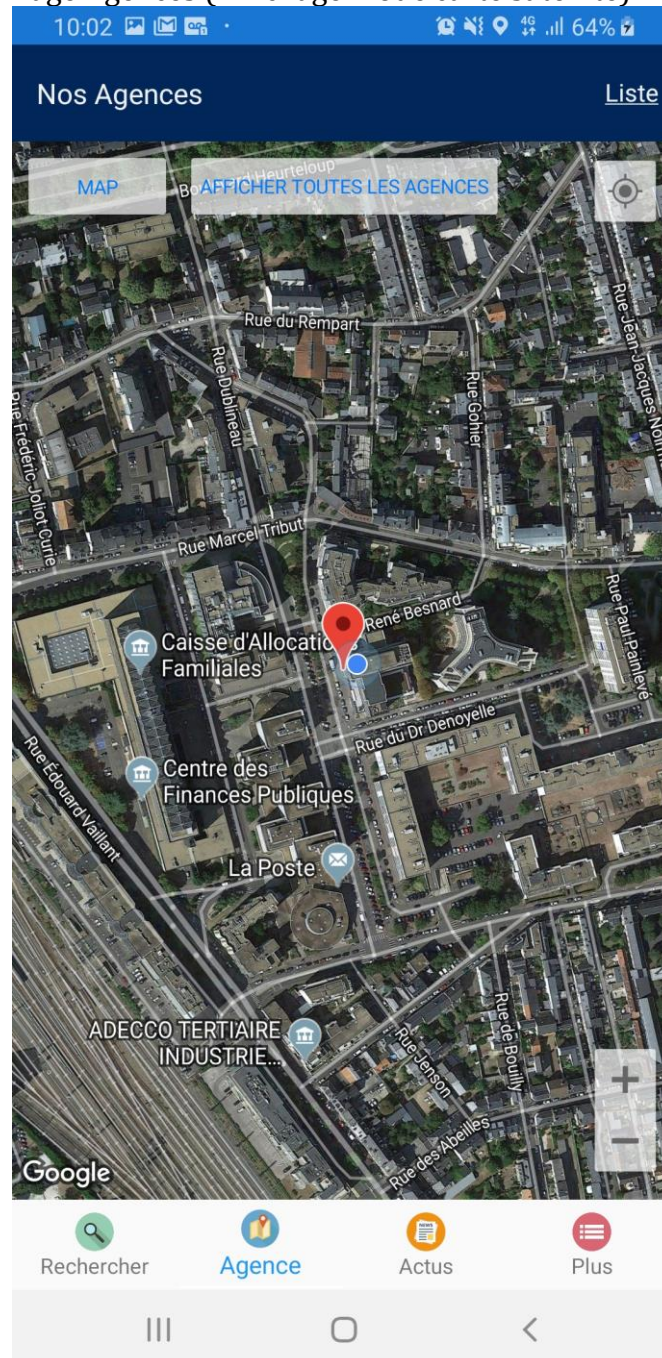
Page Agences (Affichage mode carte) :



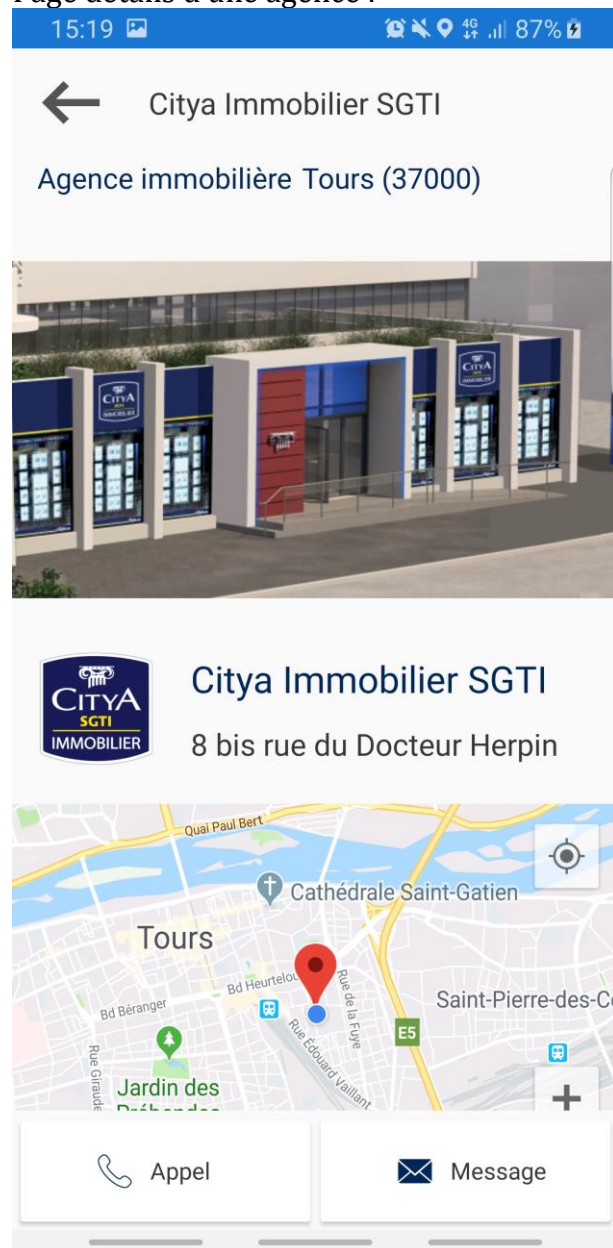
Page Agences (Affichage mode liste) :



Page Agences (Affichage mode carte satellite) :



Page détails d'une agence :



Page des actualités :



Belle fête du 14 juillet à tous !

Publié par Antoine Billault le 07/14/2019 | Actualités presse



Page des actualités, avec le filtre par thèmes :



Page détails d'une actualité :



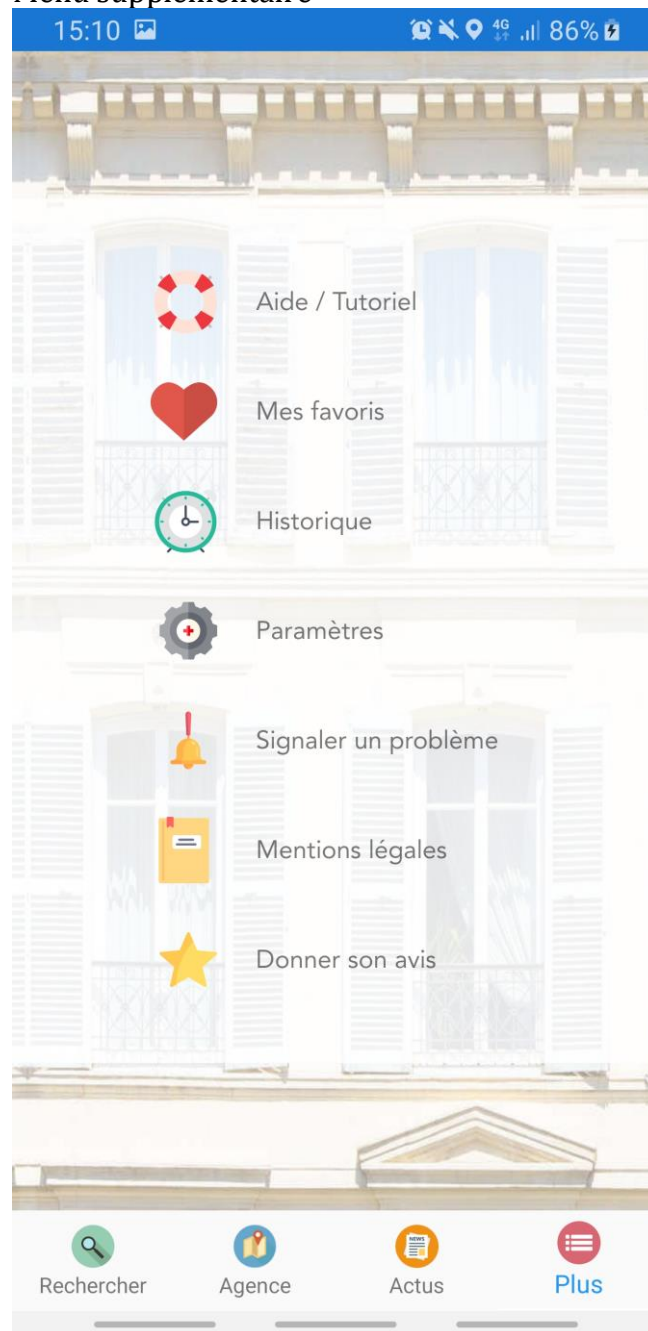
Aujourd'hui, les résultats du BAC tombent ! Les étudiants de la France entière vont décider de leur futur logement sur le site de petites annonces Leboncoin !



Une belle opportunité pour **Citya Immobilier** d'accompagner tous les nouveaux bacheliers dans leurs recherches d'appartements !

Découvrez notre habillage exclusif sur ordinateurs, tablettes et mobiles toute la journée

Menu supplémentaire



ANNEXE 2 – REFERENCE DU RAPPORT

<https://docs.microsoft.com/fr-fr/xamarin/>

<https://medium.com/firebase-developers/firebase-analytics-on-xamarin-forms-c60ae8bbb24a>

<https://docs.microsoft.com/en-us/xamarin/android/data-cloud/google-messaging/firebase-cloud-messaging>

ANNEX 3 - RÉALISATION SUR L'INTRANET

Le livret Rh :

Bienvenue Louis

Découvrez nos filiales

175 agences
Vente, Location, Gérance, Syndic

700 agences franchisées
Vente, Location, Gérance

Cabinet de courtage
en assurances

Garanties des risques
locatifs

Réseau de prestataires indépendants
assurant les EDL entrée et sortie

Structure spécialisée dans la négociation
de mandats de gestion en immobilier neuf

Structure spécialisée dans l'optimisation
patrimoniale et financière

Courtier expert en prêts
immobilier

Réseau de mandataires
immobilières

Organisme de formation
et cabinet de recrutement

Agence de communication
& événementielle intégrée

Service informatique intégré

Réseau immobilier
500 agences et cabinets d'affaires

Réseau immobilier
500 agences et cabinets d'affaires

*Devenez ambassadeur
de notre marque sur le web*

Abonnez-vous sur nos réseaux
et rejoignez nos communautés !

ÉTAPE SUIVANTE

UN PROJET IMMOBILIER est-il facile ?
A DÉCOUVRIR EN 2019

0 vues cumulées
depuis le 14/05/2019

3 320 410 visiteurs uniques
sur citya.com depuis le 01/01/2019
(3 033 829 en 2018)

0 locations
depuis le 01/01/2019
(36 731 en 2018)

0 ventes
depuis le 01/01/2019
(8 112 en 2018)

0 locations
depuis le 01/01/2019
(36 731 en 2018)

0 ventes
depuis le 01/01/2019
(8 112 en 2018)

0 lots en gérance
depuis le 01/01/2019
(153 627 en 2018)

0 lots en copro
depuis le 01/01/2019
(898 801 en 2018)

0 connexions à i-Citya
depuis le 01 janvier 2019
(892 611 connexions en 2018)

99 116 fans

13 325 followers

12 222 abonnés

1 527 fans

1 189 abonnés

72 155 téléchargements

Conditions générales d'utilisation de l'Intranet

Le site INTRANET est mis à la disposition des Utilisateurs à des fins professionnelles. De ce fait, toute connexion à l'INTRANET est subordonnée au respect, par l'Utilisateur, des présentes Conditions d'utilisation. En tant que futur Utilisateur d'INTRANET, son utilisation implique donc votre acceptation des Conditions ci-dessous. Nous vous invitons à les lire attentivement.

Notez que si vous n'êtes pas d'accord avec tout ou partie des présentes conditions d'utilisation, il vous appartient de ne pas utiliser l'Intranet et de cesser toute utilisation des services proposés.

Article 1 - Définitions générales

Nous : il s'agit de la Société CITYA Immobilier et/ou toute Société appartenant au même réseau.

Vous ou Utilisateurs : il s'agit des personnes physiques ayant accepté les conditions d'utilisation des systèmes d'informations et portails web suivants : <http://mycitya> et <https://stella.citya.com>, et pouvant dès lors y accéder.

Intranet : Il s'agit d'un « réseau de télécommunication et de téléinformatique destiné à l'usage exclusif d'un organisme et utilisant les mêmes protocoles et techniques que l'Internet ». L'accès à l'INTRANET doit être sécurisé dans la mesure où il permet à ses Utilisateurs d'avoir accès à un grand nombre d'informations

UN PROJET IMMOBILIER est-il facile ?
A DÉCOUVRIR EN 2019

97 239 vues cumulées
depuis le 14/05/2019

3 320 410 visiteurs uniques
sur citya.com depuis le 01/01/2019
(3 033 829 en 2018)

12 126 locations
depuis le 01/01/2019
(36 731 en 2018)

2 578 ventes
depuis le 01/01/2019
(8 112 en 2018)

Louis DUBOIS

Mémoire alternance - REFERENCE

58

d'emprisonnement et de 375 000 euros d'amende).

L'inobservation des Conditions définies ci-dessus pourra également entraîner le prononcé de sanctions disciplinaires en fonction de la gravité du manquement commis.

Enfin, vous ne devez en aucun cas vous livrer à une activité concurrente à celle de notre Société ou susceptible de nous causer un quelconque préjudice, en utilisant les systèmes d'informations et portails web mis à votre disposition.

De manière générale, chacun, par son action et son attitude contribue à la sécurité des systèmes d'information. Une pratique responsable et l'application par chacun des quelques principes évoqués ci-dessus, ainsi que la sollicitation des informaticiens pour des besoins spécifiques, sont les garants de la sécurité des systèmes d'information mis en place par notre Société.

Cochez la case ci-dessous pour pouvoir accéder aux systèmes d'informations et portails web suivants : <http://mycitya> et <https://stella.citya.com>.

- Je reconnais avoir attentivement lu et pris connaissance de l'intégralité des Conditions d'utilisation ci-dessus.
- Je m'engage à respecter chacune des Conditions d'utilisation mentionnées.
- Je reconnais avoir pris connaissance des sanctions encourues en cas de non-respect des Conditions d'utilisation.

ÉTAPE PRÉCÉDENTE

ÉTAPE SUIVANTE

148 639

lots en gérance

depuis le 01/01/2019

(153 627 en 2018)

518 205

lots en copro

depuis le 01/01/2019

(488 901 en 2018)

256 172

connexions à i-Citya

depuis le 01 janvier 2019

(992 611 connexions en 2018)

99 116 fans

13 325 followers

12 222 abonnés

1 527 fans

1 189 abonnés

72 155 téléchargements

Disponible sur App Store

Disponible sur Play Store