RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
http://www.ifri-uac.bj Courriel : contact@ifri.uac.bj

MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Génie Logiciel

Présenté par :
Shadrac Meshac **DIMON**

Développement d'une application mobile de colocation pour étudiant

Sous la supervision :
Ing Fidèle **HOUNSOUNON**

Membres du jury :

COMLAN Maurice	Docteur	IFRI/UAC	Président
GNONLONFOUN Miranda	Ingénieur	IFRI/UAC	Examinatrice
HOUNSOUNON Fidèle	Ingénieur	IFRI/UAC	Rapporteur

Année Académique : **2023-2024**

Sommaire

Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Table des figures vi

Liste des tableaux vii

Liste des acronymes viii

Glossaire ix

Introduction 1

1 Revue de la littérature 3

2 Méthode de conception et choix techniques 9

3 Résultats et discussion 21

Conclusion 33

Webographie 34

Bibliographie 34

Table des matières 36

Dédicace

Je dédie ce travail à :

- l'Eternel, mon Dieu, pour son accompagnement et sa protection tout au long de ce parcours académique qui s'achève ;
- mon père Gilbert DIMON ;
- ma mère Véronique KAMI ;
- mes frères, soeurs et tous ceux ayant contribué d'une manière ou d'une autre à la réalisation de ce travail ;

Remerciements

Bien consciente que ces quelques mots sont loin de refléter la grandeur du rôle précieux que vous avez joué dans la réalisation de ce travail, mes remerciements les plus sincères vont à l'endroit de :

- Professeur C. Eugène EZIN, Directeur de l'Institut de Formation et de Recherche en Informatique (IFRI) ;
- Tout le corps professoral de l'Institut de Formation et de Recherche en Informatique pour l'accompagnement et la formation de pointe durant ces trois dernières années ;;
- Monsieur Fidèle HOUNSOUNON , mon maître de mémoire, qui malgré ses nombreuses occupations, a accepté d'encadrer ce travail ;
- Toute ma famille, mes parents, mes frères et sœurs, qui m'ont toujours soutenu et encouragé ;
- Mes amis, Gustavo-Marie, Gracia, Sams, Shadrak, Chancelle, Camille, Athéna.

Résumé

Aujourd'hui face aux prix assez élevés des logements dans les milieux universitaires, la colocation est la solution de logement adoptée par un grand nombre d'étudiants. Ainsi dans la recherche de colocation et/ou de colocataire, ces étudiants sont confrontés à la difficulté de trouver le bon partenaire, un logement proche du campus à un prix abordable et beaucoup d'autres difficultés. Nous avons proposé dans ce projet, la mise en place d'une application mobile de colocation étudiante. L'implémentation et le développement de l'application android ont été rendus possible grâce au langage de modélisation UML, au framework Flutter et à une base de données temps réels Firebase.

Mots clés : étudiant, uac, up, application mobile,flutter, logement, colocation, colocataire

Abstract

Today, faced with the relatively high housing prices in university areas, co-living is the housing solution adopted by a large number of students. Thus, in the search for shared housing and/or roommates, these students face the difficulty of finding the right partner, a place close to the campus at an affordable price, and many other challenges. In this project, we propose the development of a student co-living mobile application. The implementation and development of the Android application were made possible thanks to the UML modeling language, the Flutter framework, and a real-time Firebase database.

Key words: student, UAC (University of Abomey-Calavi), UP (University of Parakou), mobile application, Flutter, housing, shared housing, roommate

Table des figures

1.1 Logo de Whoomies . . . 5
1.2 Interface de Whoomies . . . 5
1.3 Logo de Coloc'qui . . . 6
1.4 Illustration de Coloc'qui . . . 6
1.5 Illustration de Wahwé . . . 7

2.1 Logo de COLOC ETU . . . 9
2.2 Diagramme de cas d'utilisation de COLOC ETU . . . 11
2.3 Diagramme de cas de classe de COLOC ETU . . . 13
2.4 Diagramme de cas de séquence d'authentification . . . 14
2.5 Diagramme de séquence de publications de logement . . . 15
2.6 Diagramme de séquence de recherche de colocataire . . . 16
2.7 Logo Figma . . . 17
2.8 Logo Dart . . . 17
2.9 Logo Flutter . . . 17
2.10 Logo Firebase . . . 18
2.11 Logo PHP . . . 18
2.12 Logo Laravel . . . 18

3.1 Page de connexion . . . 22
3.2 Interfaces d'inscription sur COLOC ETU . . . 23
3.3 Page d'accueil . . . 24
3.4 Interfaces de publication de logement sur COLOC ETU . . . 25
3.5 Interface de recherche de colocataires . . . 26
3.6 Interfaces de messagerie sur COLOC ETU . . . 27
3.7 Mon profil . . . 28
3.8 Profil public . . . 28
3.9 Interface de gestion des utilisateurs . . . 29
3.10 Détail et Validation de profil utilisateur . . . 30
3.11 Gestion des annonces de colocation . . . 31

Liste des tableaux

2.1 Acteurs et leurs rôles dans l'application 10
2.2 Classes et leurs descriptions 12
2.3 Services Payants à l'Unité 19

Liste des acronymes

BaaS :

Backend-as-a-Service *:* BaaS

JEE :

Java Enterprise Edition *:* JEE

OS :

Operating System *:* OS

UML :

Unified Modeling Language *:* UML

Glossaire

BaaS : est un modèle de service cloud dans lequel les développeurs externalisent tous les aspects de l'arrière-plan d'une application web ou mobile afin de n'avoir à écrire et à maintenir que le frontend. Les fournisseurs de BaaS fournissent des logiciels pré-écrits pour les activités qui ont lieu sur les serveurs, telles que l'authentification des utilisateurs, la gestion des bases de données, la mise à jour à distance et les notifications « push » (pour les applications mobiles), ainsi que le stockage et l'hébergement sur le cloud. viii

JEE : est une technologie développée par Sun puis Oracle pour le développement d'applications distribuées en Java. Depuis la reprise du projet par la Fondation Eclipse, il a été renommé Jakarta EE car la marque Java reste la propriété d'Oracle. Il fait partie d'Eclipse viii

OS : est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs viii

UML : est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet. viii

Introduction Générale

Contexte

La colocation est une solution de logement populaire chez les étudiants, surtout dans les villes de Cotonou ou d'Abomey-Calavi où les loyers sont élevés. Cependant, trouver un logement en colocation peut être compliqué et les plateformes actuelles ne répondent pas toujours aux besoins spécifiques des étudiants. Ils recherchent souvent un logement proche de leur université, à prix abordable, et avec des colocataires compatibles.
C'est dans ce contexte que l'idée de développer une application dédiée à la colocation étudiante prend tout son sens. Cette application permettra aux étudiants de trouver plus facilement des colocataires.

Problématique

Actuellement, de nombreux étudiants recherchant une colocation ou un colocataire publient leurs annonces dans des groupes ou pages Facebook. Cependant, cette méthode présente plusieurs risques, tels que le manque de fiabilité des informations, l'absence de contrôle des annonces, ou encore les difficultés de vérification de l'identité des personnes intéressées. Bien que certains outils en ligne existent pour faciliter la recherche de colocations, ils sont souvent mal adaptés aux besoins spécifiques des étudiants, que ce soit en termes de budget, de localisation, ou de transparence des informations.
Face à cette situation, notre projet vise à concevoir une application mobile dédiée à la colocation étudiante, apportant ainsi une solution à ce problème. Cette application permettra aux étudiants de publier et consulter des annonces de colocation en ayant accès aux informations détaillées sur le logement ainsi que sur les colocataires potentiels, favorisant ainsi une recherche plus sécurisée et adaptée aux besoins et ressources des étudiants.

Objectifs

L'objectif principal de ce travail est de concevoir une application mobile qui a pour but de permettre aux étudiants, à travers des interfaces conviviales, de rechercher des colocataires ou de publier leur

maison en colocation. De façon spécifique, notre solution permettra :

- aux utilisateurs de rechercher des logements disponibles pour la colocation proches du campus en fonction de critères spécifiques tels que le loyer et la localisation ;
- la recherche de colocataires compatibles en fournissant des fonctionnalités de profilage ;
- la communication entre les étudiants intéressés par la colocation à travers des fonctionnalités de messagerie ;
- la vérification et la validation des annonces de logements disponibles pour la colocation et des profils d'utilisateurs pour assurer la fiabilité des informations.

Organisation du mémoire

Le présent document est structuré en trois chapitres. Le premier chapitre présente les solutions existantes et l'intérêt de notre solution par rapport à celles-ci. Le deuxième chapitre expose notre solution à travers sa modélisation, son fonctionnement, et les choix techniques liés à sa réalisation. Les résultats attendus, la présentation des interfaces, suivis des insuffisances qui sont liées à notre solution, sont présentés dans le troisième et dernier chapitre.

Chapitre 1

Revue de la littérature

Introduction

Dans le présent chapitre, nous faisons une étude de l'existant afin de mieux appréhender le problème auquel nous voulons apporter de solution. Ce qui nous permettra de ressortir les insuffisances des solutions existantes et de formuler des hypothèses de recherche

1.1 Clarification conceptuelle

1.1.1 Une application mobile

Une **application mobile** ou appli est un logiciel applicatif conçu pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un smartphone, un baladeur numérique, une tablette tactile. Elles sont pour la plupart distribuées depuis des plateformes de téléchargement (parfois elles-mêmes contrôlées par les fabricants de smartphones) telles que l'**App Store** (plateforme d'Apple), le **Google Play** (plateforme de Google / Android), ou encore le **Microsoft Store** (plateforme de Microsoft pour Windows 10 Mobile). Chacun des mobiles dispose d'un langage de programmation qui lui est propre, permettant le développement de ses applications. Les applications pour les terminaux **Apple** sont développées dans un langage principalement dédié à ces applications mobiles, le **Swift**. Celles pour **Windows Mobile** sont développées en **C sharp**, langage aussi utilisé pour les programmes exécutables de **Microsoft Windows** (.exe). Le système **Android** utilise, quant à lui, un langage universel, le **Java**, langage pouvant être utilisé pour les ordinateurs, le développement Web [9]

1.1.2 La colocation

La **colocation** est définie par l'article 8-1 de la loi du 6 juillet 1989 comme « la location d'un même logement par plusieurs locataires, constituant leur résidence principale, et formalisée par la conclusion d'un contrat unique ou de plusieurs contrats entre les locataires et le bailleur ». Pour être en colocation, il faut que les occupants du logement soient extérieurs à la famille du propriétaire. Ils peuvent cependant avoir des liens sociaux entre eux : frère, sœur, grands-parents, amis, etc. Tout est possible ! [2]

1.1.3 La recherche de colocataire

La **recherche de colocataire** fait référence au processus par lequel une personne déjà en colocation ou disposant d'un logement cherche une ou plusieurs autres personnes pour partager ce logement.

1.1.4 La recherche de colocation

La **recherche de colocation** désigne le processus par lequel une personne cherche un logement partagé avec d'autres locataires.

1.2 Etude des solutions existantes

1.2.1 Whoomies

Whoomies est la première application dédiée à la recherche et à la réservation de de logements partagés (chambres à louer en colocation, résidences étudiantes, appart'hôtels, hôtels et espaces de coliving) et de matching entre colocataires ! [3] La recherche est axée sur la FRANCE et LONDRES. Whoomies vous met en relation avec des profils compatibles et facilite l'accès au logement pour les étudiant(e)s et jeunes actifs.



FIGURE 1.1 : Logo de Whoomies

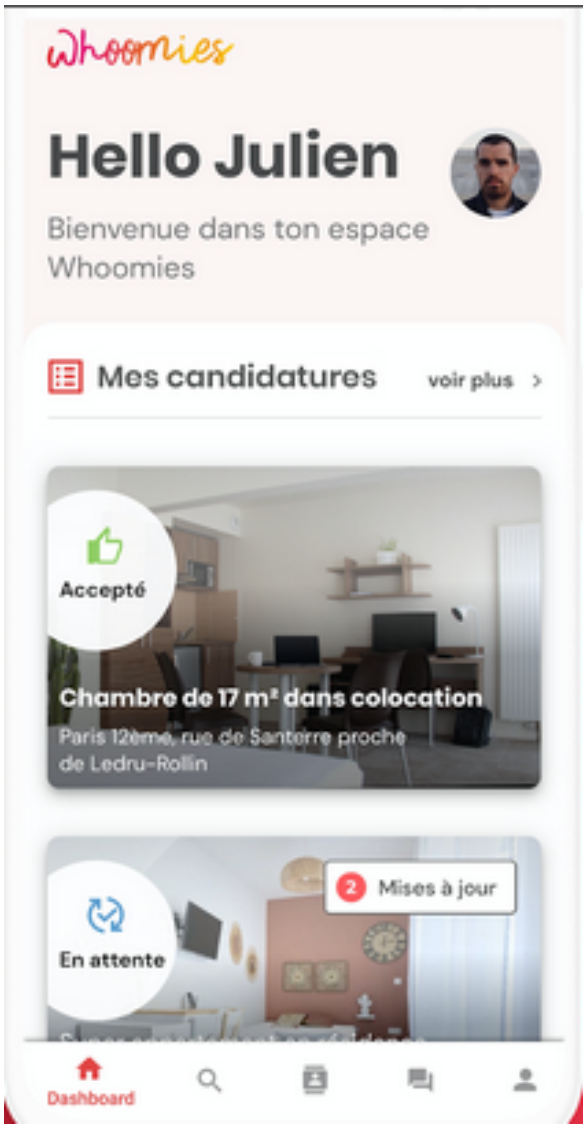


FIGURE 1.2 : Interface de Whoomies

Forces de Whoomies

Whoomies :

- permet aux utilisateurs d'accéder à une large sélection d'annonces de colocation.
- utilise des algorithmes de matching basés sur les préférences et le style de vie des utilisateurs
- dispose d'une interface intuitive, avec des fonctionnalités simples telles que la messagerie interne, les filtres de recherche détaillés, et une navigation fluide.

1.2.2 Coloc'qui

Coloc'qui est une plateforme spécialement conçue pour aider les étudiants et jeunes professionnels à trouver une colocation en Tunisie. Lancée par Chaima Torkhani, cette solution numérique a pour objectif de répondre à plusieurs problèmes majeurs dans le domaine du logement étudiant, notamment l'absence de plateforme centralisée et l'accessibilité réduite aux logements, qu'ils soient publics ou privés.[8] ColocQui agit comme le Tinder de la colocation, facilitant la recherche de colocataires de manière rapide, efficace et sans stress.



FIGURE 1.3 : Logo de Coloc'qui



FIGURE 1.4 : Illustration de Coloc'qui

Forces de Coloc'qui

Coloc'qui :

- est une plateforme adaptée aux étudiants.
- assure la sécurité des utilisateurs en vérifiant les informations des colocataires potentiels, augmentant la confiance entre les parties.
- facilite la répartition des charges financières entre colocataires.
- offre des modèles de contrat de colocation pour protéger juridiquement les utilisateurs

1.2.3 Wahwé

Wahwé, signifiant "Viens à la maison" en langue Fon, est une application mobile prototype développée par **Stephen HOUNSOUGBO**, ancien étudiant d'IFRI dans le cadre de son projet de soutenance. Le projet vise à faciliter la réservation de chambres d'hôtes au Bénin. L'objectif principal de cette plateforme est de permettre aux propriétaires de logements d'augmenter leur visibilité et d'atteindre un public plus large, tout en offrant aux voyageurs une solution simple et efficace pour trouver et réserver un logement adapté à leurs besoins et budgets.

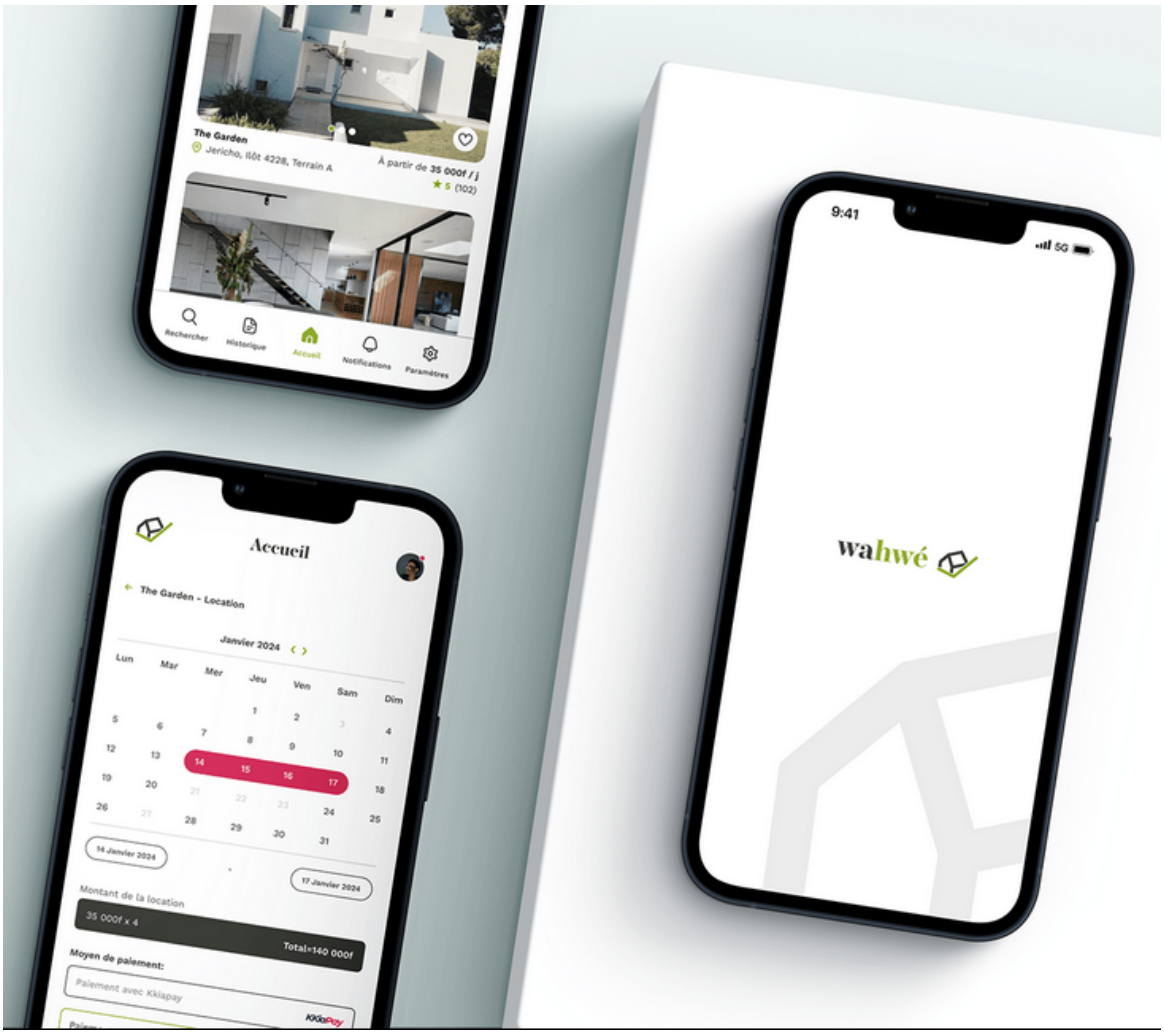


FIGURE 1.5 : Illustration de Wahwé

Forces de Wahwé

Wahwé :

- est le prototype d'une application mobile ciblant des touristes et voyageurs cherchant des logements temporaires.
- permet aux propriétaires de logements d'atteindre un public plus large pour maximiser les locations.
- dispose d'une interface intuitive pour rechercher, comparer et réserver des logements selon le budget et les besoins.
- offre des modèles de contrat de colocation pour protéger juridiquement les utilisateurs

1.3 Limites des solutions

Nous pouvons constater que l'accès géographique est restreint pour les applications Whoomies et COLOC'QUI. Ce qui ne nous permet pas d'utiliser ces applications au Bénin. Quand à Wahwé, elle n'est pas orientée vers la colocation et ne propose pas de fonctionnalités spécifiques pour le partage de logements entre colocataires. De plus, l'application cible principalement les voyageurs et les

touristes, ce qui la rend moins adaptée aux étudiants cherchant des logements abordables pour de longues durées. Par ailleurs, Wahwé ne favorise pas les interactions sociales et la création de relations entre colocataires, un aspect clé pour les applications de colocation. Elle ne propose pas non plus de filtres spécifiques pour trouver des colocataires selon des critères comme le style de vie ou les préférences personnelles. Enfin, il est important de noter que Wahwé n'existe pour l'instant qu'à l'état de prototype. Elle n'est pas encore développée en tant qu'application fonctionnelle, ce qui limite sa capacité à être utilisée ou testée dans un contexte réel.

1.4 Hypothèses de recherche

La limite que nous avons relevées précédemment nous ont permis d'envisager une solution qui respecte les spécifications suivantes :

- la solution à mettre en place doit être intuitif et ergonomique
- la solution doit être simple d'utilisation
- elle doit être accessible aux étudiants béninois
- elle doit vérifier les annonces et les profils utilisateurs
- elle permettra de diffuser des annonces de logements disponible pour la colocation et de rechercher des colocataires
- Elle doit permettre aux étudiants de pouvoir communiquer à travers un chat intégré

Conclusion

Ce chapitre nous a permis de parcourir quelques solutions pouvant aider les étudiants dans la recherche de colocation ou de colocataire mais elles ne sont pas utilisables au Bénin. Le but de notre travail est de proposer aux étudiants une solution locale. Dans le chapitre suivant nous présenterons notre méthode de conception et les choix techniques opérés.

Chapitre 2

Méthode de conception et choix techniques

Introduction

En nous basant sur les hypothèses de recherche évoquées dans le chapitre précédent, nous avons développé une application : **COLOC ETU**, qui permettra aux étudiants de trouver un coloc ou un logement. Dans ce chapitre nous exposons la méthode de travail utilisée pour conduire le projet, nous modélisons le système et présentons son fonctionnement général.



FIGURE 2.1 : Logo de COLOC ETU

2.1 Modélisation

Pour modéliser les fonctionnalités de notre application mobile, nous avons porté notre choix sur le langage UML. UML est un langage de modélisation visuelle standard destiné à être utilisé pour spécifier, visualiser et documenter des modèles de systèmes logiciels, y compris leur structure et leur conception de manière à répondre à toutes ses exigences.

2.1.1 Identification des acteurs

Un acteur est une entité extérieure au système modélisé, et qui interagit directement avec lui. Pour notre application mobile nous avons recensés deux (02) acteurs qui sont :

- Les étudiants ;
- L'administrateur.

2.1.2 Rôle des acteurs

Le tableau ci-dessous présente les rôles liés à chaque acteur qui interagit avec le système.

N°	Acteur	Rôles
1	Étudiants	• S'inscrire en remplissant un formulaire d'inscription pour créer un compte • S'authentifier pour accéder aux différents menus que lui propose l'application • Publier un logement disponible pour la colocation • Rechercher un colocataire • Communiquer avec un potentiel colocataire • Signaler un autre étudiant
2	Administrateur	• Vérifier l'annonce de logement disponible et les profils des étudiants • Valider une annonce • Supprimer un compte utilisateur

TABLE 2.1 : Acteurs et leurs rôles dans l'application

2.1.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation constitue la première étape de l'analyse en UML qui résume la manière dont les utilisateurs interagissent avec un système. Il permet de recueillir, d'analyser, d'organiser les besoins et de recenser les grandes fonctionnalités d'un système. Nous aurons une vue globale du comportement fonctionnel du système grâce au diagramme de cas d'utilisation. Le diagramme de cas d'utilisation relatif à notre domaine d'étude se présente comme suit :

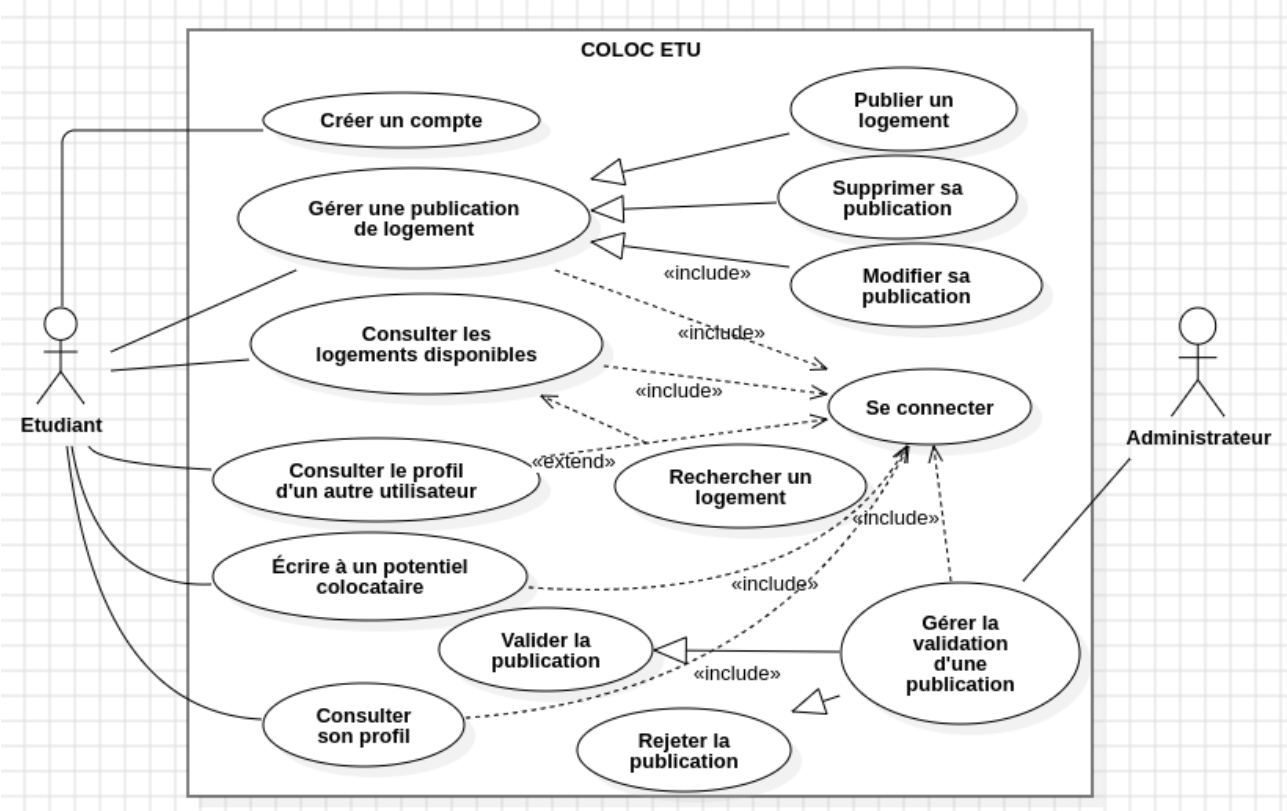


FIGURE 2.2 : Diagramme de cas d'utilisation de COLOC ETU

2.1.4 Diagramme de classe

Le diagramme de classe est un diagramme UML qui exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classe est de modéliser les entités du système d'information. Le diagramme des classes comporte les onze classes suivantes :

Classe	Description
Maison	Représente un logement disponible pour la colocation. A une image, une catégorie, un type, etc. Peut être modifiée, supprimée, et ses détails peuvent être affichés.
Utilisateur	Représente un utilisateur avec des informations comme l'ID, le nom, l'email, etc. Il peut créer un compte, se connecter, se déconnecter et mettre à jour ses identifiants.
Conversation	Représente une conversation entre deux étudiants. Permet de récupérer et afficher les messages.
Message	Contient les informations liées à un message individuel (expéditeur, destinataire, type, etc.). Permet d'envoyer et de lire un message.
MsgAudio	Sous-classe de Message représentant un message audio avec un contenu spécifique audio.
MsgLocation	Sous-classe de Message représentant un message de localisation avec des coordonnées GPS.
MsgImage	Sous-classe de Message représentant un message de type image.
MsgText	Sous-classe de Message représentant un message de type texte.
Etudiant	Contient des informations spécifiques à un utilisateur étudiant telles que numéro d'inscription, université, et date de naissance.
Administrateur	Représente un utilisateur admin avec des privilèges supplémentaires comme valider des publications et gérer des utilisateurs.
Annonce	Représente une publication ou annonce associée à une maison. Elle contient des méthodes pour publier, modifier, et supprimer l'annonce, ainsi qu'une fonctionnalité de recherche liée à la maison.

TABLE 2.2 : Classes et leurs descriptions

La figure ci dessous présente le diagramme de classe de notre application.

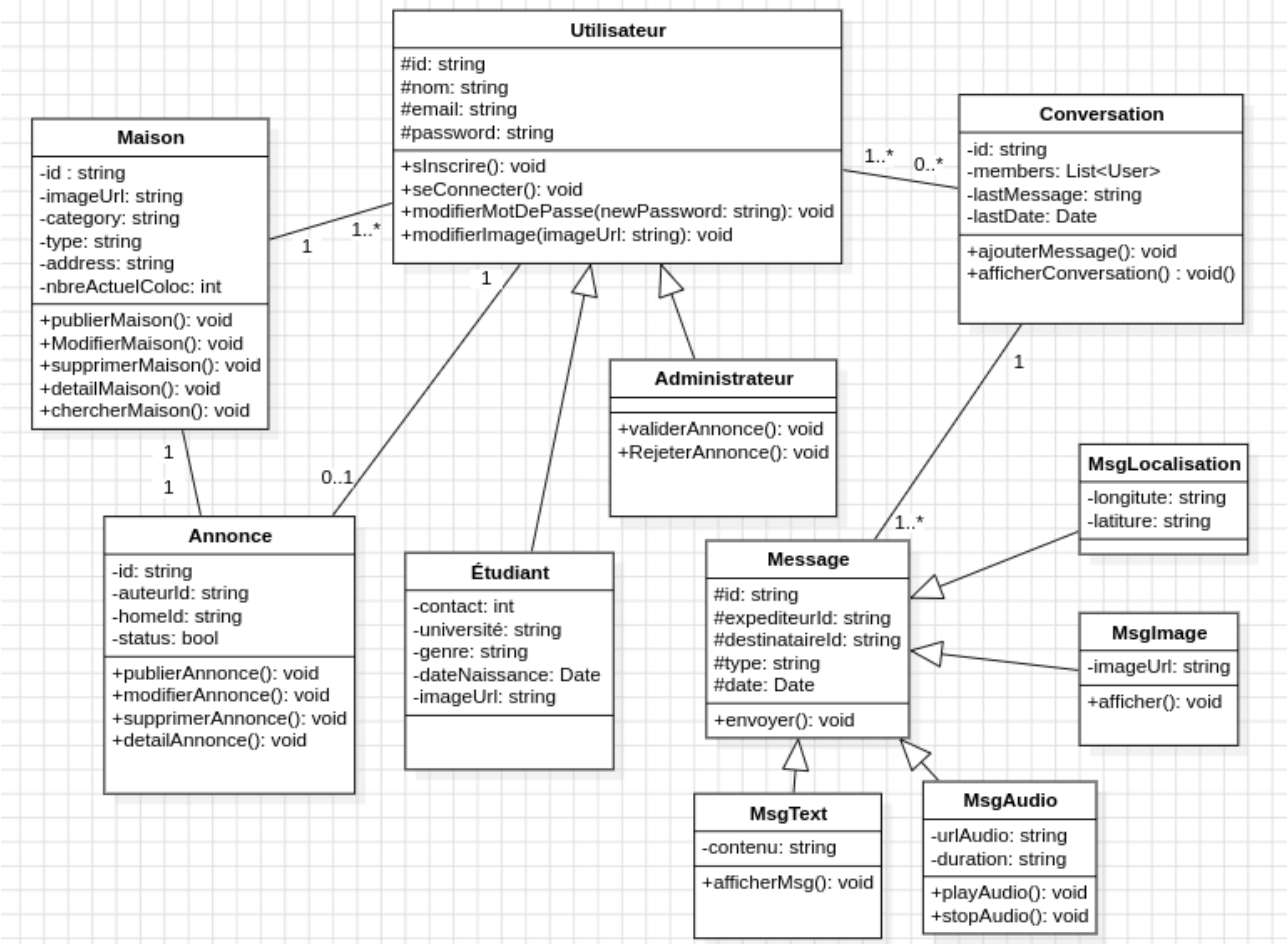


FIGURE 2.3 : Diagramme de cas de classe de COLOC ETU

2.1.5 Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique.

2.1.5.1 Diagramme de séquence pour l'authentification

La figure présente le diagramme de séquence pour l'authentification sur l'application.

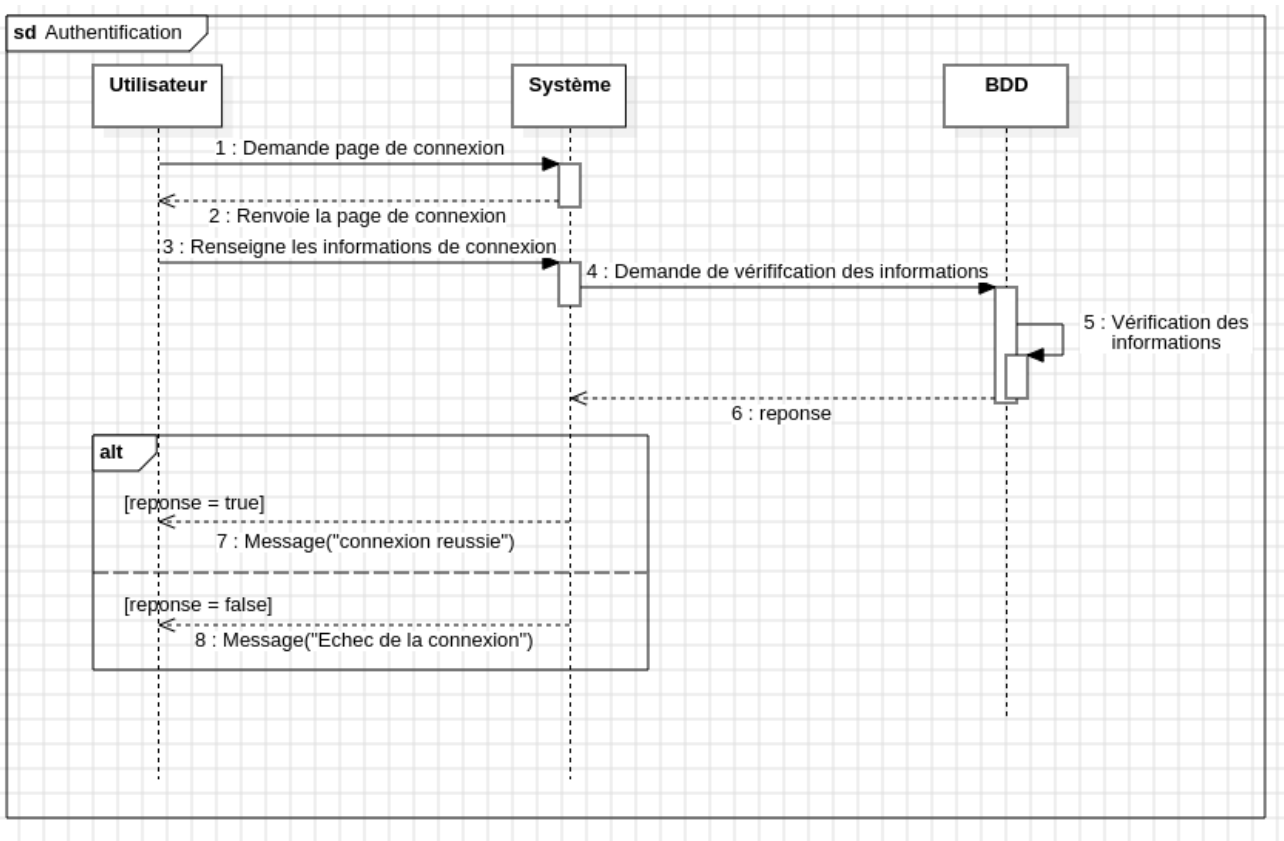


FIGURE 2.4 : Diagramme de cas de séquence d'authentification

2.1.5.2 Diagramme de séquence de publications de logement disponible pour la colocation

A travers ce diagramme de séquence nous allons montrer comment un utilisateur fait sa publication.

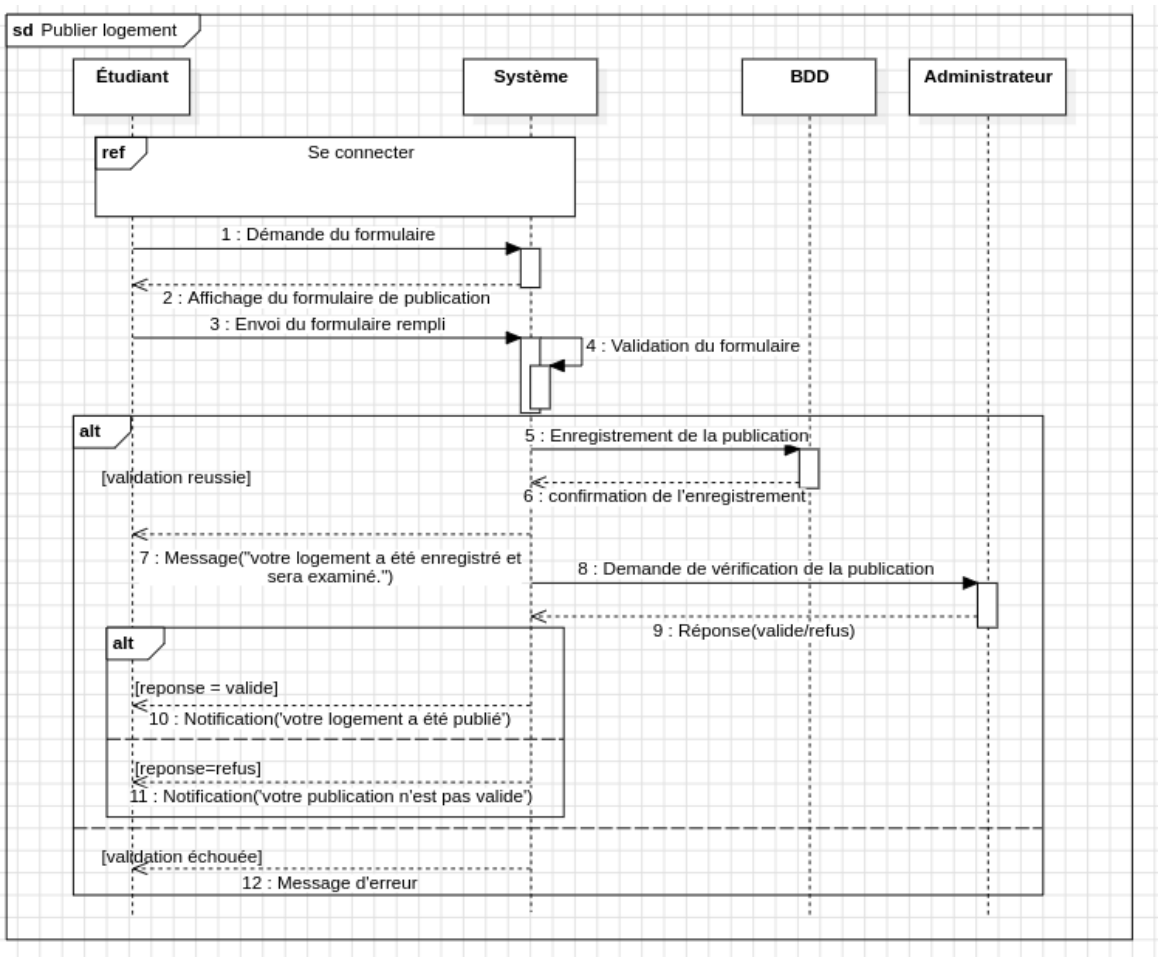


FIGURE 2.5 : Diagramme de séquence de publications de logement

2.1.5.3 Diagramme de séquence de recherche de colocataire

A travers ce diagramme de séquence nous allons montrer comment un utilisateur fait la recherche de colocataire

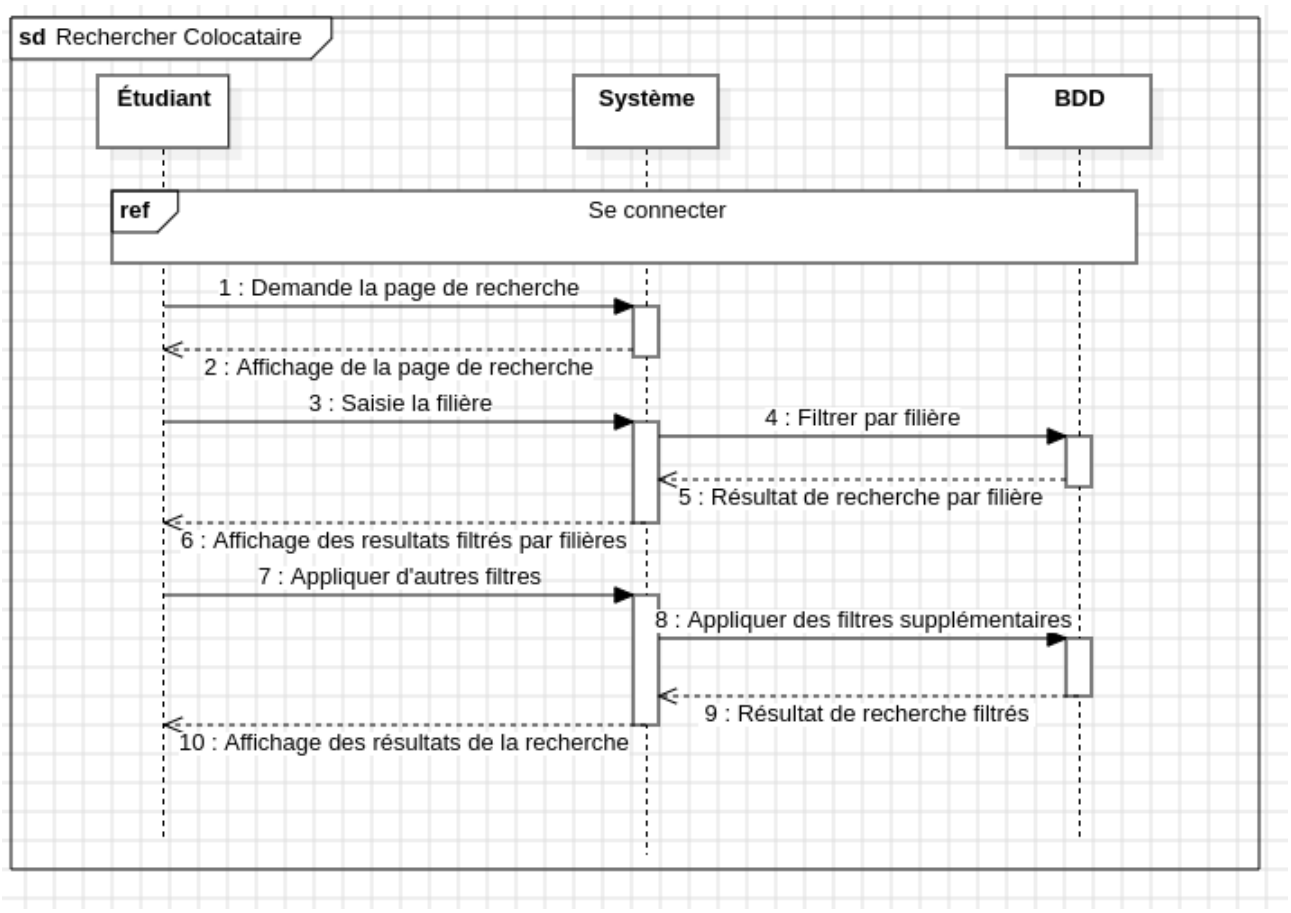


FIGURE 2.6 : Diagramme de séquence de recherche de colocataire

2.2 Choix techniques

2.2.1 Front-end

Le Front-end représente tout ce que l'utilisateur voit, tout ce qui lui permet d'interagir avec l'application comme : les boutons, les images, les contenus textuels, les couleurs, etc. Les technologies qui entrent en jeu dans le Front-end s'exécutent du côté client, c'est-à-dire au niveau du navigateur.

2.2.1.1 Figma

Figma est un outil de création et de manipulation graphique des éléments vectoriels pour le prototypage et la création des interfaces des logiciels, des applications Web et mobiles ainsi que des sites Web. Il s'agit d'un **outil collaboratif** par excellence qui permet de communiquer et réaliser des modifications en temps réel via un navigateur Web. [1]

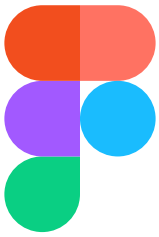


FIGURE 2.7 : Logo Figma

2.2.1.2 Dart

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web. Dart est un langage orienté objet à ramasse-miettes- un sous-système de gestion automatique de la mémoire) avec une syntaxe de type C++. Dart peut se compiler en code natif ou en JavaScript. Il prend en charge les interfaces (un ensemble de signatures de méthodes publiques d'un objet.) , les mixins (une classe destinée à être composée par héritage multiple avec une autre classe pour lui apporter des fonctionnalités) , les extensions de classes, les classes abstraites, les génériques réifiés et l'inférence de type 3.[4] La version stable actuelle (06/10/2024) est 3.5.3 [7]



FIGURE 2.8 : Logo Dart

2.2.1.3 Flutter

Flutter est un kit de développement logiciel (SDK) d'interface utilisateur open-source créé par Google. Il est utilisé pour développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code. La première version de Flutter était connue sous le nom de code "Sky" et fonctionnait sur le système d'exploitation Android. Elle a été dévoilée lors du sommet des développeurs Dart de 2015, avec l'intention déclarée de pouvoir effectuer un rendu cohérent à 120 images par seconde . Lors du discours d'ouverture des Google Developer Days à Shanghai, Google a annoncé la sortie de Flutter Release Preview 2 qui est la dernière grande version avant Flutter 1.0. Le 4 décembre 2018, Flutter 1.0 a été publié lors de l'événement Flutter Live, ce qui représente la première version "stable" du Framework. [5] Flutter :

- permet de créer des applications pour Android, iOS, web et desktop à partir d'un seul code
- offre des performances fluides et une expérience utilisateur proche du natif
- bénéficie d'une grande communauté de développeurs ce qui aide lorsqu'on rencontre des problèmes



FIGURE 2.9 : Logo Flutter

2.2.2 Back-end

« Backend » est un terme anglais signifiant littéralement « **partie arrière** », ou encore « sous-structure ».le backend correspond à la partie en arrière-plan, invisible pour les utilisateurs. Le stockage et le traitement des données s'effectuent dans le backend.[6]

2.2.2.1 Firebase

Firebase est une plateforme mobile de Google facilitant la création de Backend scalable et performant, dédiée principalement aux applications mobiles. C'est une plateforme de développement d'applications Backend-as-a-Service (BaaS) qui fournit des services backend hébergés tels qu'une base de données en temps réel, le stockage dans le cloud, l'authentification, les rapports d'incident, l'apprentissage automatique, la configuration à distance et l'hébergement de fichiers statiques.



FIGURE 2.10 : Logo Firebase

2.2.2.2 PHP

PHP (acronyme de PHP Hypertext Preprocessor) est un langage de script open source, côté serveur, largement utilisé pour le développement web.



FIGURE 2.11 : Logo PHP

2.2.2.3 Laravel

Laravel est un framework PHP qui fournit une sécurité de haut niveau avec des bibliothèques orientées objet et fonctionne sur l'architecture MVC.Ils offrent aux développeurs web un certain nombre d'avantages, notamment un développement plus rapide, un besoin réduit d'écrire du code et une sécurité renforcée.

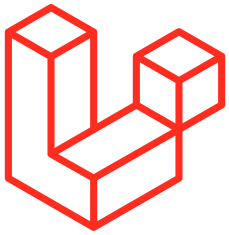


FIGURE 2.12 : Logo Laravel

2.3 Modèle Économique

Le modèle économique de Coloc Étu a été conçu pour répondre aux besoins spécifiques des étudiants en recherche de logement tout en assurant la viabilité du projet. En combinant une approche Freemium, des services payants à l'unité, et un programme de parrainage, ce modèle vise à offrir une solution accessible et attrayante, tout en générant des revenus durables. Cette section présente les principales composantes du modèle, leurs avantages, et leur contribution à la pérennité du service. COLOC ETU est une application mobile conçue pour faciliter la recherche et la mise en relation de colocataires étudiants au Bénin. Son modèle économique s'articule autour d'un service freemium avec une seule fonctionnalité payante.

2.3.1 Présentation du Modèle

COLOC ETU est une application mobile conçue pour faciliter la recherche et la mise en relation de colocataires étudiants au Bénin. Son modèle économique est basé sur un accès entièrement gratuit à toutes les fonctionnalités, rendant la plateforme accessible à tous sans restrictions. Un seul service payant est disponible, conçu pour offrir un soutien spécifique aux utilisateurs qui souhaitent publier une annonce :

Services Payants à l'Unité	**Prix**
Publication d'Annonce de Logement	150 FCFA (jusqu'à trouver un colocataire)

TABLE 2.3 : Services Payants à l'Unité

2.3.2 Structure de Coûts

2.3.2.1 Coûts Fixes

- **Développement et maintenance** : Utilisation de technologies comme Flutter pour l'application et Laravel pour le backend.
- **Infrastructure technique** : Hébergement sur Firebase, y compris la gestion des données utilisateurs et des images.
- **Support client** : Assistance continue pour les utilisateurs.
- **Marketing et communication** : Actions promotionnelles pour attirer et retenir les utilisateurs, surtout dans les principales villes universitaires.

2.3.2.2 Coûts Variables

- **Vérification des profils** : coûts associés aux processus de vérification de l'identité des utilisateurs et des annonces.

2.3.3 Stratégie de Croissance

2.3.3.1 Court Terme

- **Lancement ciblé** dans les principales villes universitaires du Bénin pour une visibilité immédiate.
- **Conversion Freemium à Premium** : Mise en avant des avantages Premium pour encourager les utilisateurs gratuits à souscrire.
- **Renforcement de la base d'utilisateurs actifs** : Promotion locale pour atteindre un seuil critique de membres actifs.

2.3.3.2 Moyen-Long Terme

- **Extension régionale** : Expansion vers d'autres pays d'Afrique de l'Ouest avec une forte population étudiante (Sénégal, Côte d'Ivoire, Togo).
- **Diversification des services** : Intégration de services complémentaires pour enrichir l'expérience utilisateur.
- **Partenariats stratégiques** : Collaboration avec entreprises locales et organismes d'étudiants pour des offres groupées.

2.3.4 Indicateurs Clés de Performance (KPI)

2.3.4.1 Métriques d'Acquisition

- **Nombre d'inscriptions** : Indicateur principal de l'attractivité de la plateforme.
- **Taux de conversion Freemium à Premium** : Indicateur de la valeur perçue des fonctionnalités Premium.
- **Coût d'acquisition client (CAC)** : Mesure de l'efficacité des campagnes marketing.

2.3.4.2 Métriques d'Engagement

- **Nombre d'annonces publiées** : Suivi de l'activité et de l'engagement sur la plateforme.
- **Taux de réponse aux messages** : Mesure de la réactivité et de l'engagement des utilisateurs.
- **Nombre de colocations réussies** : Évaluation de la satisfaction des utilisateurs.
- **Durée moyenne d'utilisation** : Indicateur de l'adoption et de la fidélité.

Conclusion

Dans ce chapitre, nous avons présenté les diagrammes de classes et de cas d'utilisation qui ont permis d'avoir l'architecture globale de notre projet. Nous avons aussi présenté les outils que nous avons utilisés et le modèle économique de COLOC ETU. Les résultats de notre travail seront présentés dans le chapitre suivant par les captures d'écran de l'application mobile.

Chapitre 3

Résultats et discussion

Introduction

Ce chapitre présente le résultat de notre travail au travers de captures d'écran ainsi que ses limites. Il s'agit d'une application mobile offrant la possibilité aux étudiants de trouver des colocataires ou de trouver une colocation au moyen de leur téléphone.

3.1 Fonctionnement général de l'application

L'application mobile réalisée, appelée **COLOC ETU**, a pour objectif de permettre aux étudiants de trouver des colocataires ou une colocation.

3.1.1 Interfaces d'authentification et dashboard

L'étudiant doit s'authentifier pour accéder à sa page d'accueil et pouvoir interagir avec l'application. Sur cette page, on trouve les différentes catégories suivantes : Home (pour voir les logements disponibles), Search (pour rechercher des colocataires), Messages (pour discuter avec des colocataires potentiels), et Profil (pour voir ses informations personnelles et accéder aux paramètres de l'application).

- **Page d'authentification** :



FIGURE 3.1 : Page de connexion

L'étudiant est invité à saisir son email et son mot de passe pour se connecter.



FIGURE 3.2 : Interfaces d'inscription sur COLOC ETU

- **Page d'accueil** :

Une fois connecté, l'étudiant accède à sa page d'accueil et peut ainsi utiliser pleinement l'application. Les interfaces sont conçues pour être simples d'utilisation. Il peut consulter et éditer son profil, accéder à la liste des logements disponibles pour la colocation, consulter les profils et discuter avec des colocataires potentiels.

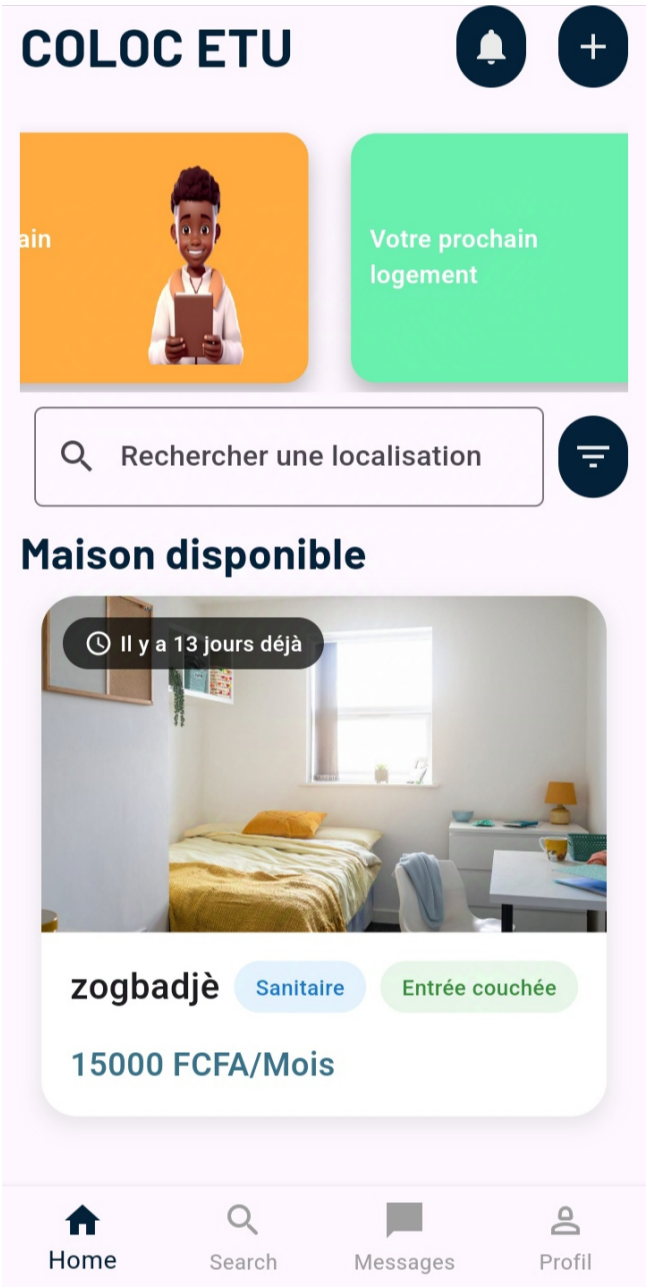


FIGURE 3.3 : Page d'accueil

3.1.2 Interface de publication de logement

La page de publication de logement dans COLOC ETU permet aux étudiants de poster des annonces de logement disponibles pour la colocation. Pour cela, l'étudiant complète un formulaire et ajoute des images, comme le montrent les captures ci-dessous :



FIGURE 3.4 : Interfaces de publication de logement sur COLOC ETU

3.1.3 Interface de recherche de colocataires

Cette interface permet à l'utilisateur de rechercher des colocataires potentiels selon plusieurs critères (emplacement, type de logement, genre souhaité, budget maximum). La recherche affiche les profils des colocataires avec des informations essentielles telles que le nom, l'université et le genre.



FIGURE 3.5 : Interface de recherche de colocataires

3.1.4 Interface de messagerie

L'interface de messagerie permet aux utilisateurs d'envoyer des messages en temps réel avec d'autres colocataires potentiels. Cela favorise le dialogue entre utilisateurs pour mieux se connaître avant de choisir un colocataire.

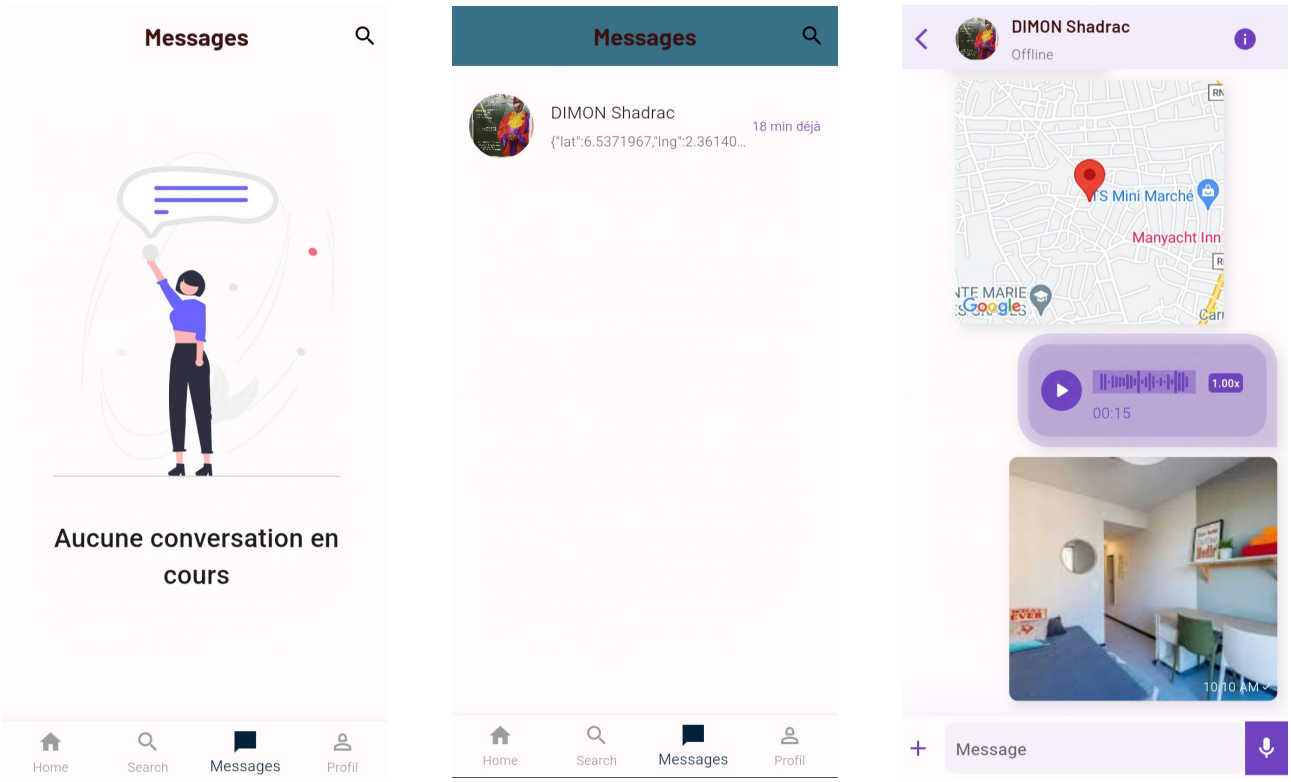


FIGURE 3.6 : Interfaces de messagerie sur COLOC ETU

3.1.5 Interface de consultation du profil utilisateur

Cette interface permet d'accéder aux informations d'un autre utilisateur comme le nom, le numéro de téléphone, et l'université, facilitant ainsi l'évaluation de la compatibilité avec un potentiel colocataire.

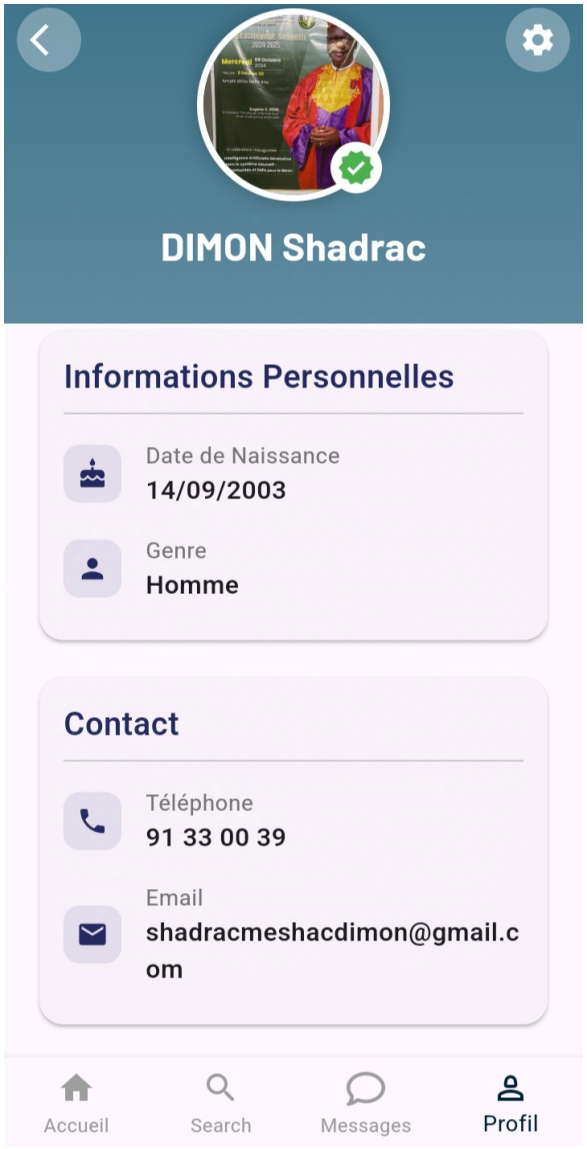


FIGURE 3.7 : Mon profil

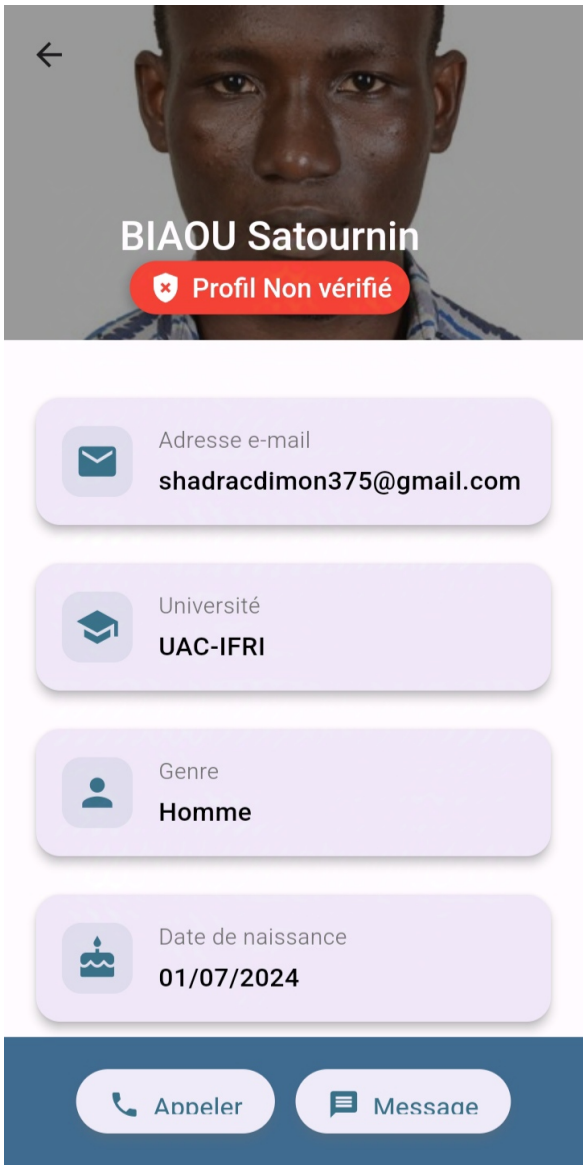


FIGURE 3.8 : Profil public

3.2 Interfaces de l'administrateur de l'application COLOC ETU

L'interface administrateur permet de gérer les utilisateurs et les annonces de colocation. L'administrateur peut valider ou supprimer un profil utilisateur, et aussi vérifier les documents d'identification téléchargés pour chaque utilisateur.

3.2.1 Gestion des utilisateurs

L'administrateur peut accéder à une liste complète des utilisateurs avec la possibilité de filtrer et de rechercher des profils spécifiques. Chaque profil présente des informations détaillées telles que le nom, le statut (validé ou non), et des actions rapides comme la validation ou la suppression de l'utilisateur.

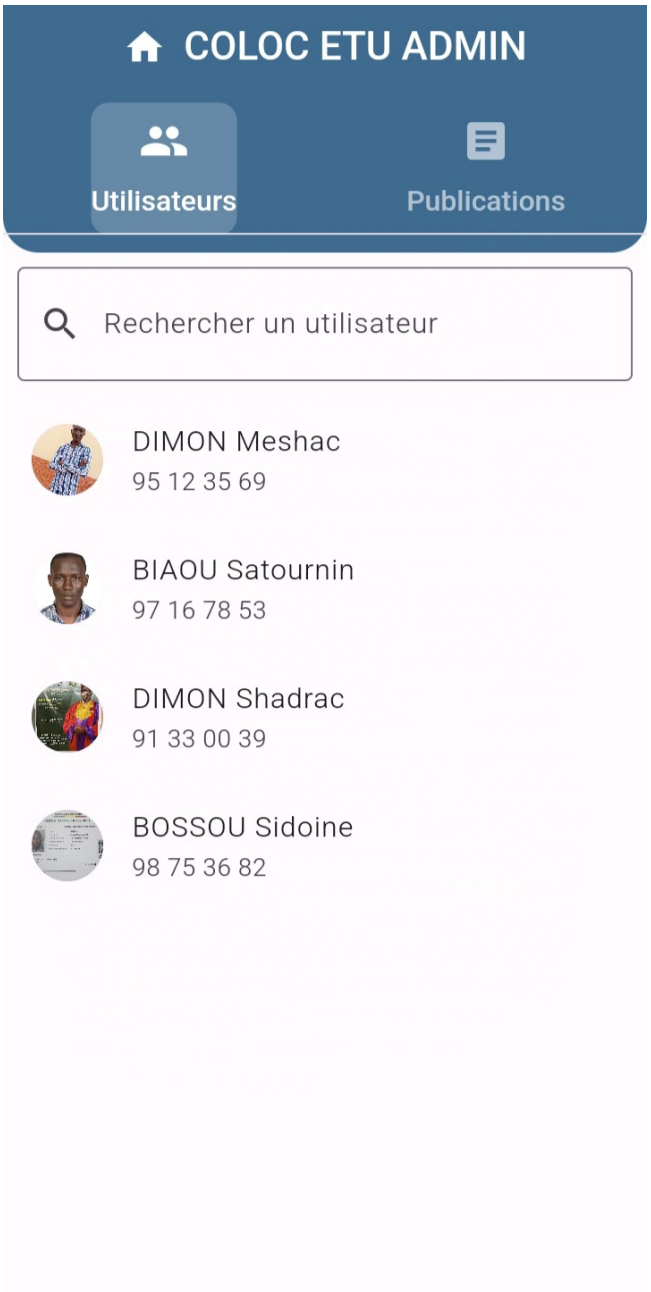


FIGURE 3.9 : Interface de gestion des utilisateurs

3.2.2 Validation des profils

L'administrateur peut vérifier l'authenticité des utilisateurs en consultant leurs informations personnelles ainsi que la carte d'identité qu'ils ont téléchargée. Après validation, l'utilisateur obtient un statut vérifié.

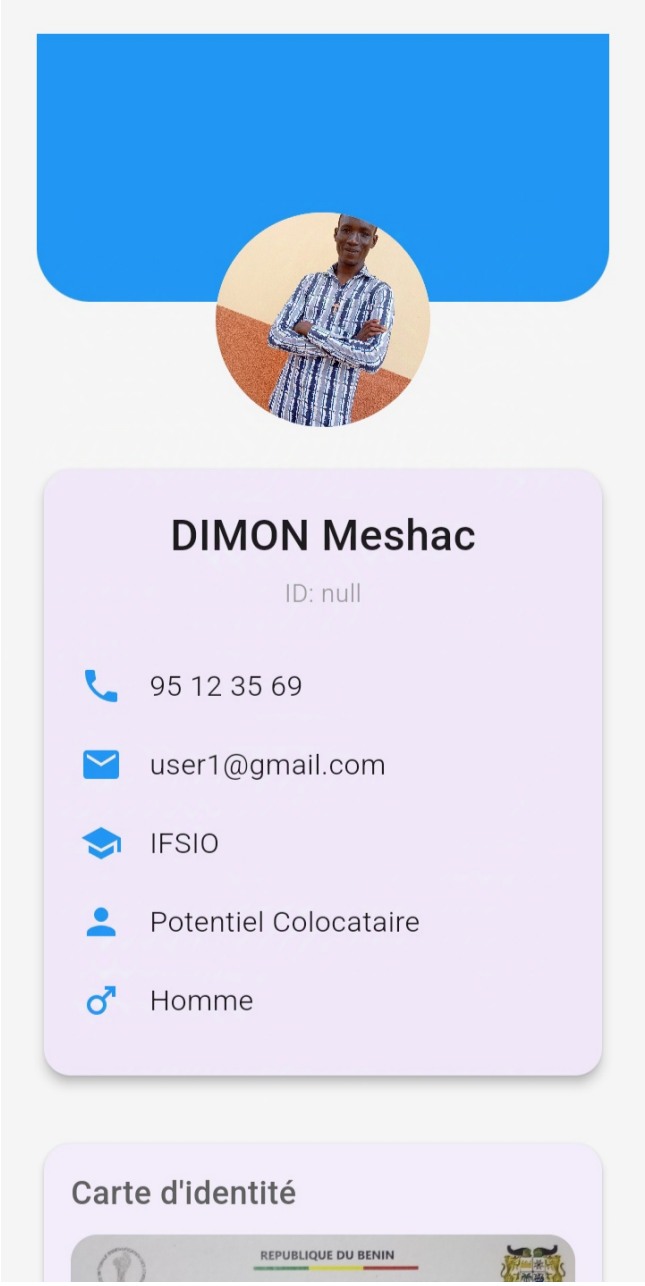


FIGURE 3.10 : Détail et Validation de profil utilisateur

3.2.3 Gestion des annonces de colocation

L'administrateur peut également consulter, valider ou supprimer des annonces de logement publiées par les utilisateurs. Cela permet d'assurer que toutes les annonces respectent les règles et les standards de la plateforme.

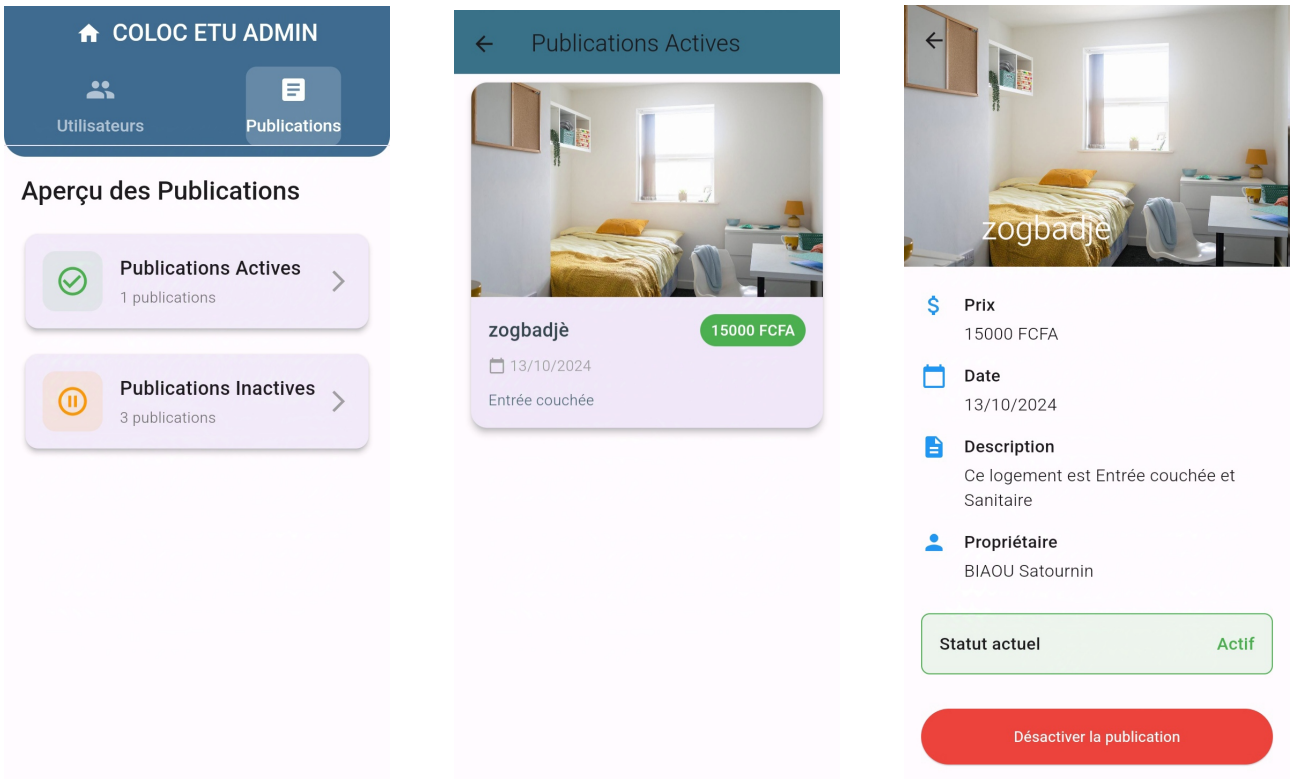


FIGURE 3.11 : Gestion des annonces de colocation

3.3 Discussion

Notre solution COLOC ETU est une application mobile conçue pour faciliter la gestion de la colocation entre étudiants. Elle répond aux défis souvent rencontrés dans la recherche de logements et de colocataires adaptés. Grâce à des fonctionnalités intuitives, COLOC ETU permet aux utilisateurs de consulter et d'éditer facilement leur profil, d'accéder à une liste mise à jour des logements disponibles pour la colocation, de consulter les profils des potentiels colocataires, et d'engager des discussions directes avec eux pour mieux se connaître avant de faire un choix.

De plus, COLOC ETU propose un système de recherche avancée, permettant aux étudiants de filtrer les résultats en fonction de critères spécifiques tels que la localisation, le prix, ou le type de logement. Les interfaces simples et ergonomiques assurent une prise en main rapide et agréable, même pour les utilisateurs les moins expérimentés.

Cependant, bien que notre application! offre déjà une solution efficace à de nombreux défis, il existe encore des améliorations à envisager pour les futures mises à jour. Notamment, COLOC ETU ne propose pas encore de gestion des paiements ou de suivi des transactions entre colocataires, ni de notifications pour informer les utilisateurs des nouvelles annonces qui correspondent à leurs critères. De plus, l'application ne permet pas de suivre les étudiants une fois qu'ils décident de se mettre en colocation. Actuellement, il n'y a aucun moyen de savoir si les étudiants, après s'être rencontrés via COLOC ETU, ont effectivement formé une colocation, ni d'obtenir des informations sur ce qui se passe ensuite, une fois la colocation établie.

Les fonctionnalités à ajouter pourraient inclure la gestion des budgets de colocation, un système de suggestion automotique de colocataires, ainsi qu'un système de notation ou de recommandations pour améliorer la confiance entre les colocataires. Ces futures améliorations rendront notre solution

encore plus complète et adaptée aux besoins des étudiants en quête de colocation.

Conclusion

Dans ce chapitre a été exposé le fruit de notre travail. Les résultats obtenus au terme de la phase de développement de l'application ont été présentés sous forme de capture d'écran et un rapport des mérites et insuffisances a également été présenté.

Conclusion Générale

Dans ce travail nous avons proposé COLOC ETU une solution mobile innovante, répondant aux besoins croissants des étudiants en matière de recherche de colocation. Grâce à des fonctionnalités pratiques telles que la consultation des profils, la discussion avec des colocataires potentiels, et la recherche avancée de logements, cette application facilite non seulement la gestion des colocations mais aussi la création de liens entre les utilisateurs. L'intégration de technologies modernes comme Flutter pour le front-end et Firebase pour le stockage des données garantit une expérience utilisateur fluide et réactive.

Au terme de ce travail, il est évident que COLOC ETU est une réponse efficace à la problématique du logement étudiant, en optimisant la recherche de logements et en favorisant la mise en relation entre colocataires potentiels. Cette application se distingue par son interface conviviale et ses fonctionnalités adaptées aux attentes des utilisateurs.

Pour aller plus loin, plusieurs améliorations peuvent être envisagées afin de rendre COLOC ETU encore plus complète et performante. Parmi les perspectives de développement, nous pouvons citer :

- Ajouter un système sécurisé pour permettre aux colocataires de gérer leurs paiements et de suivre leurs transactions directement via l'application.
- Mettre en place des notifications pour informer les utilisateurs des nouvelles annonces ou mises à jour correspondant à leurs critères de recherche, ou pour rappeler certaines échéances importantes.
- Intégrer un système de notation des colocataires et des logements
- Étendre l'application à d'autres types de logements étudiants, comme les résidences universitaires, pour toucher un plus large public.

Ces pistes d'améliorations permettront à COLOC ETU de s'adapter aux évolutions des besoins des étudiants et de rester une solution innovante et performante pour la gestion des colocations.

Webographie

[1] Figma, outil d'excellence pour vos design systems https://www.usabilis.com/figma-outil-dexcellence-vos-design-systems/.

[2] La colocation : Définition et avantages https://colivys.com/blog/colocation-definition/(consulté le 13 octobre 2024).

[3] Whoomies - trouver un colocataire https://play.google.com/store/apps/details?id=studio.lunabee.whoomies&hl=fr.

[4] Dart (langage) https://fr.wikipedia.org/wiki/Dart_(langage). Consulté le 13 octobre 2024.

[5] Flutter (logiciel) https://fr.wikipedia.org/wiki/Flutter_(logiciel). Consulté le 13 Octobre 2024.

[6] I. Blog. Qu'est-ce qu'un backend ? https://www.ionos.fr/digitalguide/sites-internet/creation-de-sites-internet/quest-ce-quun-backend/. Consulté le 13 Octobre 2024.

[7] D. Documentation. Get dart https://dart.dev/get-dart. Consulté le 13 Octobre 2024.

[8] LinkedIn. Coloc'qui - trouver des colocataires rapidement https://tn.linkedin.com/company/coloc-qui. Consulté le 13 octobre 2024.

[9] Wikipedia. https://fr.wikipedia.org/wiki/Application_mobile(consulté le 13 octobre 2024). 2024.

Bibliographie

[1] V. LOZANO. Tout ce que vous avez toujours voulu savoir sur latex sans jamais oser le demander. Framabook, ISBN : 978-2-35209-149-3, 2008.

[2] L. Piechocki. Uml, le langage de modélisation objet unifié. Developpez.com, Septembre 2009.

Table des matières

Dédicace ii

Remerciements iii

Résumé iv

. iv

Abstract v

. v

Table des figures vi

Liste des tableaux vii

Liste des acronymes viii

Glossaire ix

Introduction 1

1 Revue de la littérature 3

Revue de littérature 3

1.1 Clarification conceptuelle 3

1.1.1 Une application mobile 3

1.1.2 La colocation 3

1.1.3 La recherche de colocataire 4

1.1.4 La recherche de colocation 4

1.2 Etude des solutions existantes 4

1.2.1 Whoomies 4

1.2.2 Coloc'qui 5

1.2.3 Wahwé 6

1.3 Limites des solutions 7

1.4 Hypothèses de recherche 8

Conclusion 8

2 Méthode de conception et choix techniques 9

Introduction 9

2.1 Modélisation 9

2.1.1 Identification des acteurs 10

2.1.2 Rôle des acteurs 10

2.1.3 Diagramme de cas d'utilisation . . . 10
2.1.4 Diagramme de classe . . . 11
2.1.5 Diagramme de séquence . . . 13
2.1.5.1 Diagramme de séquence pour l'authentification . . . 13
2.1.5.2 Diagramme de séquence de publications de logement disponible pour la colocation . . . 14
2.1.5.3 Diagramme de séquence de recherche de colocataire . . . 15
2.2 Choix techniques . . . 16
2.2.1 Front-end . . . 16
2.2.1.1 Figma . . . 16
2.2.1.2 Dart . . . 17
2.2.1.3 Flutter . . . 17
2.2.2 Back-end . . . 18
2.2.2.1 Firebase . . . 18
2.2.2.2 PHP . . . 18
2.2.2.3 Laravel . . . 18
2.3 Modèle Économique . . . 19
2.3.1 Présentation du Modèle . . . 19
2.3.2 Structure de Coûts . . . 19
2.3.2.1 Coûts Fixes . . . 19
2.3.2.2 Coûts Variables . . . 19
2.3.3 Stratégie de Croissance . . . 20
2.3.3.1 Court Terme . . . 20
2.3.3.2 Moyen-Long Terme . . . 20
2.3.4 Indicateurs Clés de Performance (KPI) . . . 20
2.3.4.1 Métriques d'Acquisition . . . 20
2.3.4.2 Métriques d'Engagement . . . 20
Conclusion . . . 20

3 Résultats et discussion **21**
Introduction . . . 21
3.1 Fonctionnement général de l'application . . . 21
3.1.1 Interfaces d'authentification et dashboard . . . 21
3.1.2 Interface de publication de logement . . . 24
3.1.3 Interface de recherche de colocataires . . . 25
3.1.4 Interface de messagerie . . . 26
3.1.5 Interface de consultation du profil utilisateur . . . 27
3.2 Interfaces de l'administrateur de l'application COLOC ETU . . . 28
3.2.1 Gestion des utilisateurs . . . 28
3.2.2 Validation des profils . . . 29
3.2.3 Gestion des annonces de colocation . . . 30
3.3 Discussion . . . 31
Conclusion . . . 32

Conclusion **33**

Webographie 34

Bibliographie 34

Table des matières 36