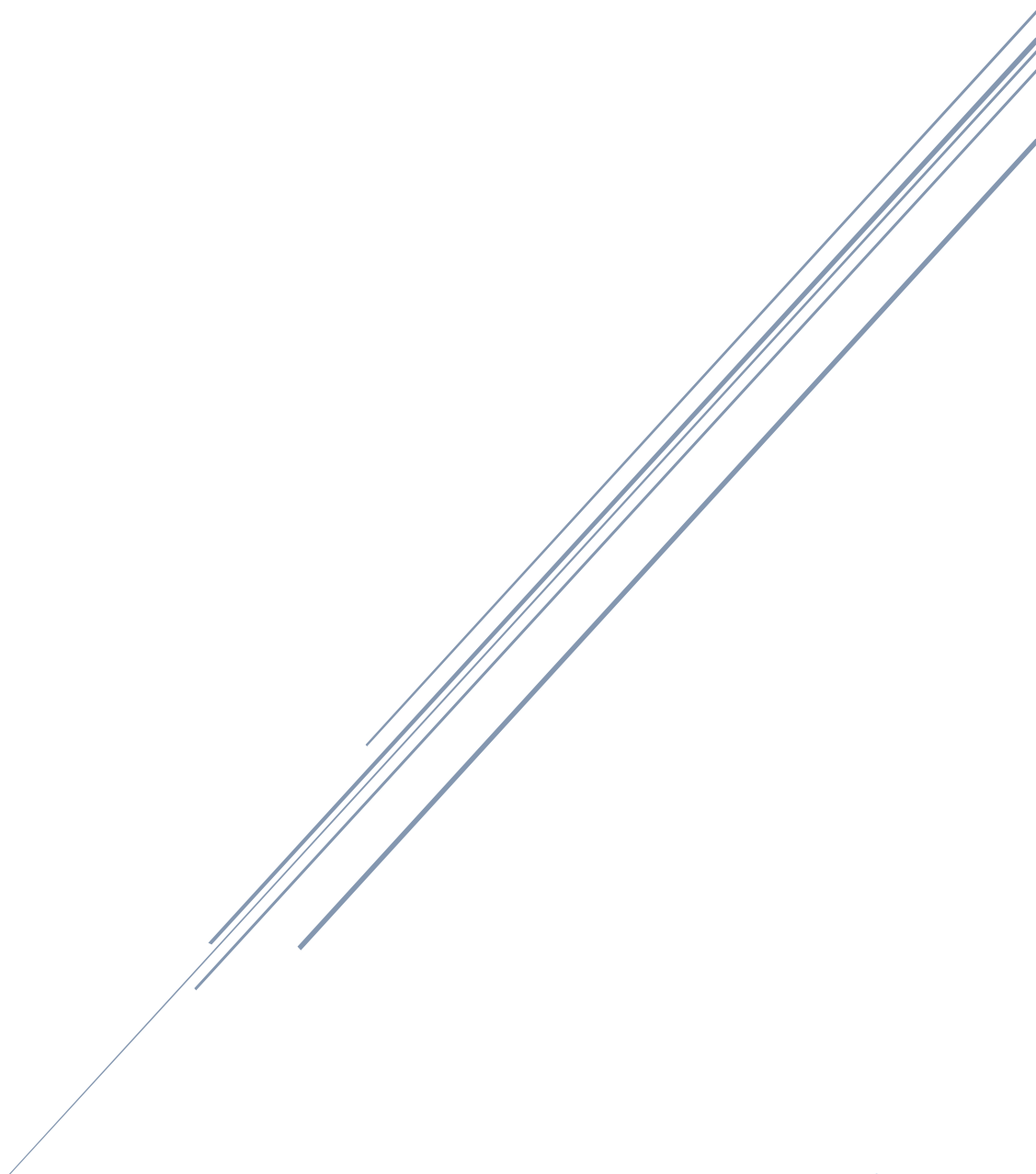


[TITRE DU DOCUMENT]

[Sous-titre du document]



[École]

[Titre du cours]

## DEDICACES

*Nous dédions ce modeste travail, comme preuve de respect, de gratitude, et de reconnaissance à : Nos chers parents : Qui n'ont jamais cessé de nous encourager et nous conseiller, Ils nous ont beaucoup aidé tout en long de notre chemin, grâce à leur amour, leur compréhension, leurs sacrifices, leur tendresse, leurs prières et leur patience sans jamais nous quitter des yeux ni baisser les bras et leur soutien moral et matériel, on ne saurait jamais traduire ce qu'on ressent vraiment envers eux.*

*Nos chers frères et sœurs : Pour leurs encouragements permanents, et leur soutien moral, pour leur indulgence en notre faveur qu'ils touchent ici l'affection la plus intime qu'on ressent à leur égard.*

*Nos amis et collègues : Pour leur compagnie et bons moments passés ensemble. Et tous ceux qui nous sont chers...*

## REMERCIEMENTS

*Ce rapport est le fruit des efforts fournis et des sacrifices consentis par plusieurs personnes que je ne pourrai oublier de remercier.*

*Mes remerciements s'adressent d'abord à Dieu, créateur de toutes choses, pour tous ses innombrables bienfaits.*

*Je tiens particulièrement à remercier, le Professeur Ahmed Sejad d'avoir accepté de nous encadrer dans la conception et l'élaboration de ce travail, ainsi que pour ses orientations continues et pour le grand soutien qui nous apporté.*

## TABLE DES MATIERES

DEDICACES .....	0
REMERCIEMENTS .....	2
TABLES DE Figures .....	5
INTRODUCTION .....	6
CHAPITRE 1 : PRÉSENTATION GÉNÉRALE DU PROJET .....	7
A. PROBLEMATIQUE .....	7
B. objectifs.....	7
C. Conclusion du chapitre .....	8
chapitre 2 : processus de developpement.....	9
A. PROCESSUS DE DEVELOPPEMENT LOGICIEL .....	9
B. POURQUOI UNE METHODE ?.....	9
C. PRESENTATION DE QUELQUES METHODES DE DEVELOPPEMENT LOGICIEL .....	10
L'APPROCHE TRADITIONNELLE .....	10
L'APPROCHE AGILE.....	11
LA METHODE SCRUM.....	11
CHOIX DE LA METHODE DE DEVELOPPEMENT : SCRUM .....	12
D. Un processus de modelisation avec uml.....	13
1. UML (UNIFIED MODELING LANGUAGE) .....	13
2. BASES D'UML .....	13
3. DIFFERENT TYPES DE DIAGRAMME D'UML .....	13
3.1 Diagramme de cas d'utilisation : .....	14
3.2 Diagramme de séquence : .....	14
3.3 Diagramme de classe : .....	15
E. Conclusion du chapitre .....	15
CHAPITRE 3 : LES FONCTIONNALITES .....	16
CHAPITRE 4 : CONCEPTION DE LA SOLUTION .....	17
A. DIAGRAMMES DE SEQUENCES DES CAS D'UTILISATIONS.....	17
1. DIAGRAMME DE SEQUENCE .....	17
CHAPITRE 5 : RÉALISATION ET DÉVELOPPEMENT .....	20
A. OUTILS ET TECHNOLOGIES.....	20
1. Flutter.....	20
2. Dart .....	20
3. uml .....	21

4. android studio.....	21
visual studio code.....	21
Firebase.....	22
B. INTERFACES DE NOTRE APPLICATION .....	23
C. Admin.....	28
CONCLUSION DU CHAPITRE .....	30
CONCLUSION ET PERSPECTIVES .....	31
REFERENCES.....	32

## TABLES DE FIGURES

Figure 1 : Scrum .....	12
Figure 2 : Diffèrent Type De Diagramme D' Uml.....	14
Figure 3 : Diagramme De Séquence .....	18
Figure 4 : Diagramme De Cas D'utilisation .....	19
Figure 5 : Flutter.....	20
Figure 6 : Dart.....	20
Figure 7 : Uml.....	21
Figure 8 : Android studio .....	21
Figure 9 : Visual Studio Code .....	21
Figure 10 : Firebase.....	22
Figure 11 : Interface Client .....	23
Figure 12 : Interface Admin.....	28

## INTRODUCTION

*Bienvenue sur notre application de réservation d'appartements ! Nous comprenons que trouver l'appartement parfait peut être un processus difficile et chronophage. C'est pourquoi nous avons créé une application qui simplifie le processus de recherche d'appartement et vous aide à trouver l'endroit idéal pour vous sentir chez vous.*

*Notre application fournit une plate-forme conviviale qui vous permet de rechercher facilement des appartements en fonction de votre emplacement préféré, de votre fourchette de prix et des commodités souhaitées.*

*C'est là qu'intervient notre application. est une application mobile qui simplifie le processus de réservation d'appartements en fournissant une large sélection d'appartements dans différentes villes du monde, des informations de disponibilité et de prix en temps réel, des avis et des notes d'utilisateurs*

## CHAPITRE 1 : PRÉSENTATION GÉNÉRALE DU PROJET

*Ce chapitre a pour objectif de situer le projet dans son contexte général, à savoir la problématique qui a inspiré la création de notre application, la description du projet et les objectifs à atteindre.*

### A. PROBLEMATIQUE

*Le problème qui a conduit à la création de l'application SEKENI est l'inefficacité et la transparence du processus de location des appartements traditionnels aux locataires.*

*Les locataires rencontrent souvent des difficultés lorsqu'ils cherchent à louer un appartement pour leur résidence. Ils ont dû passer par des agences de location ou consulter plusieurs sites Web, ce qui peut être chronophage, fastidieux et peu fiable.*

*De plus, les informations disponibles sur les appartements étaient souvent limitées, avec une brève description et quelques photos. Les voyageurs ne pouvaient pas se faire une idée précise de l'apparence et des commodités de l'appartement. Cela a souvent conduit à des déceptions et des problèmes une fois sur place.*

*Ainsi, le motif de création de l'application SEKENI était le désir de résoudre ces problèmes et de simplifier le processus de location pour les locataires. L'application vise à fournir une plateforme centralisée où les locataires peuvent facilement trouver des appartements de qualité avec des informations détaillées.*

*En bref, le problème qui a conduit à la création de l'application SEKENI est la nécessité de simplifier, d'améliorer la transparence et de renforcer la confiance dans le processus de réservation des appartements pour les locataires.*

### B. OBJECTIFS

*Le but de cette formation est de créer une application sur le téléphone mobile, et l'idée de cette application est due au manque de services qui relient le travailleur et le client en plus d'avoir une application pour le transport. Nous avons créé cette application car il n'y a pas une telle application dans notre pays qui sert la communauté. Cette application se caractérise par plusieurs avantages, notamment la protection des informations et la sécurité des utilisateurs, L'application relie l'utilisateur de l'application au client de l'appartement à louer dans l'application en cliquant sur des appartements ou en recherchant des appartements afin que l'utilisateur puisse communiquer avec le propriétaire de l'appartement via des messages dans*



*l'application WhatsApp ou appuyer sur l'appel pour accéder directement au contact avec le numéro.*

*Une application simple avec des photos, des adresses et de la correspondance aurait pu être créée, mais nous aurions aimé que cette application ait une fonctionnalité qui lui permettrait de devenir virale.*

*Nous avons travaillé dur pour le rencontrer*

*Ce concept n'est pas du tout nouveau donc des titres comme fiverr ont été l'inspiration pour ce projet.*

## C. CONCLUSION DU CHAPITRE

*Après avoir défini le contexte du projet ainsi que la problématique et les objectifs, nous aborderons dans le prochain chapitre les différentes méthodes existantes pour le développement des logiciels et, à la fin, nous allons choisir l'une d'entre elles sur laquelle nous travaillerons durant le projet.*

## CHAPITRE 2 : PROCESSUS DE DEVELOPPEMENT

*Dans le présent chapitre, nous allons présenter certaines des plus importantes méthodologies concernant le développement de logiciel et faire et choisir l'une d'elles pour enfin présenter la modélisation par le langage UML, pour mener à bien la réalisation de notre projet*

### A. PROCESSUS DE DEVELOPPEMENT LOGICIEL

*Un processus de développement logiciel est un ensemble d'étapes partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. Après avoir placé le sujet dans un contexte précis qui décline la volonté de sa mise en œuvre, il paraît évident d'opter pour une méthodologie de développement logiciel afin que le système à réaliser puisse remplir les attentes qui lui ont été assignées.*

### B. POURQUOI UNE METHODE ?

*La méthode de travail pour un logiciel est une manière de structurer, planifier et contrôler le processus de développement. Il s'agit d'une démarche, d'un ensemble d'étapes ou procédures à mettre en œuvre dans une logique méthodologique accompagnée d'outils et de techniques. L'utilisation d'une méthode est incontournable dans l'entreprise de tout projet, particulièrement dans la réalisation de projets informatiques. Dans ce cas-ci, les méthodes utilisées sont des méthodes d'analyse et de conception qui ont pour but la formalisation des étapes préliminaires au développement de systèmes logiciels, en d'autres termes : analyse, modélisation et conception.*

*L'urgence de l'utilisation de ces méthodes trouve son explication dans un certain nombre de facteurs :*

*De nombreux échecs de projets informatiques dans le passé sont dus à un manque d'organisation, ou une non satisfaction des besoins.*

*La révolution de l'industrie logicielle engendrée par les échecs informatiques qui introduit de nouveaux facteurs de validation de la qualité logicielle : le génie logiciel*

*Les nombreuses exigences liées au coût, aux délais et à la complexité des projets informatiques. L'utilisation de méthodes de développement de logiciels permet ainsi l'élaboration de systèmes informatiques de manière fiable et viable tout en répondant à l'ensemble des exigences du client et du génie logiciel*

## C. PRESENTATION DE QUELQUES METHODES DE DEVELOPPEMENT LOGICIEL

*Il existe plusieurs méthodes de développement informatique. L'on distingue deux approches : l'approche traditionnelle et l'approche agile. Les deux approches se distinguent essentiellement dans la manière de décomposer le projet.*

*Les méthodes cartésiennes ou fonctionnelles ou encore traditionnelles se sont imposées les premières.*

### L'APPROCHE TRADITIONNELLE

*Cette approche s'inspire directement de l'architecture des ordinateurs. Les méthodes traditionnelles prônent une démarche strictement planifiée avec une séquence d'activités bien définie. La succession des activités et le planning doivent être clairement respectés et aucun changement n'est permis. Il est attendu du client une spécification globale des besoins. Elle doit être détaillée, claire, précise et validée en entrée. Ainsi, tout doit être prévisible, du début du projet à la livraison du produit, d'où l'appellation de méthodes prédictives. Selon le planning adopté, les méthodes cartésiennes proposent plusieurs modèles d'exécution des activités du projet :*

✎ *Le modèle en cascade : Le modèle en cascade est un processus d'implémentation séquentielle, souvent utilisé dans le processus de développement de logiciel. Dans ce modèle, le processus de développement est découpé séquentiellement et de façon linéaire*

*selon les activités intrinsèques du cycle de vie du développement logiciel : l'analyse, la conception, le codage et les tests. Le plan de déroulement des phases (planification prédictive) est élaboré en tout début de processus. Le passage à une phase donnée n'est fait que si le résultat de la phase précédente a été validé et est jugé satisfaisant par le client et les utilisateurs.*

✎ *Le modèle en V : Le cycle en V est à la base de tout développement logiciel, il en représente les activités intrinsèques. Il tient d'avantage compte de la réalité que le modèle en cascade, le processus de développement n'est pas réduit à un enchaînement de tâches séquentielles. Le modèle en V permet d'anticiper sur les phases ultérieures de développement du produit en particulier les plans de test de qualifications et de performance. Parmi les méthodes traditionnelles, nous pouvons citer : SADT, CORIG ...*

## L' APPROCHE AGILE

*Cette approche est définie par les concepts suivants : la simplicité, la légèreté, la souplesse, le lien fort avec le client. C'est dans cette optique que certains apparentent le développement agile aux notions de flexibilité, de rétroaction et d'adaptation au changement rapide et continu.*

*Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients et en anticipant sur les risques. Il y'a continuellement des aller et retour avec le client. L'application logicielle est livrée par version incrémentale. Les versions successives sont aussi fiables que le livrable final en termes de tests et de validation. En quelque sorte le processus est déroulé comme un enchaînement de « minicascades ». A chaque nouvelle itération, l'ensemble de l'architecture et de la conception logicielle est reconsidéré, le code est retravaillé.*

*Les méthodes agiles aspirent donc à améliorer la réactivité et l'adaptabilité des sociétés de logiciels et constituent un moyen de survie dans un environnement instable en s'accompagnant des valeurs suivantes :*

- *Les individus et les interactions plutôt que les processus et les outils ;*
- *L'application fonctionnelle plutôt que la documentation compréhensive ;*
- *La collaboration avec le client plutôt que la négociation des contrats ;*
- *La réponse au changement plutôt que le suivi d'un plan.*

*L'agilité comprend plusieurs courants de pensée qui ont conduits à des méthodes différentes, reposant sur les mêmes concepts mais présentant des singularités. La méthode Scrum est l'un des exemples de ces méthodes.*

## LA METHODE SCRUM

*Scrum est une méthode agile de gestion de projet qui permet de produire la plus grande valeur métier dans la durée la plus courte. Elle a pour objectif d'améliorer la cohésion de l'équipe et la rapidité du processus de développement. Le nom Scrum renvoie à une pratique généralement connue au rugby signifiant la « mêlée ».*

Cette méthode qualifie un ensemble de rôles, d'instruments de gestion et de pratiques managériales favorisant un environnement basé sur la transparence, l'inspection, le suivi et l'adaptation. Le cycle de vie d'un projet Scrum peut être découpé en trois parties :

- La phase d'initiation ou démarrage : il s'agit d'une phase linéaire où l'on définit le périmètre fonctionnel du système et la liste des fonctionnalités (Backlog) agencées par ordre de priorité, d'effort, de complexité et de risque. C'est aussi à ce niveau que l'architecture est définie.
- La phase de développement est un processus empirique : le projet est découpé en cycles itératifs d'une durée de deux semaines ou sprints. Chaque sprint regroupe une ou plusieurs fonctionnalités du Backlog. Tout au long de cette phase, le travail réalisé est mesuré et contrôlé et une amélioration constante du prototype est faite.

La phase de clôture est une phase linéaire de gestion de la livraison du produit final. La figure suivante montre l'articulation générale de Scrum :

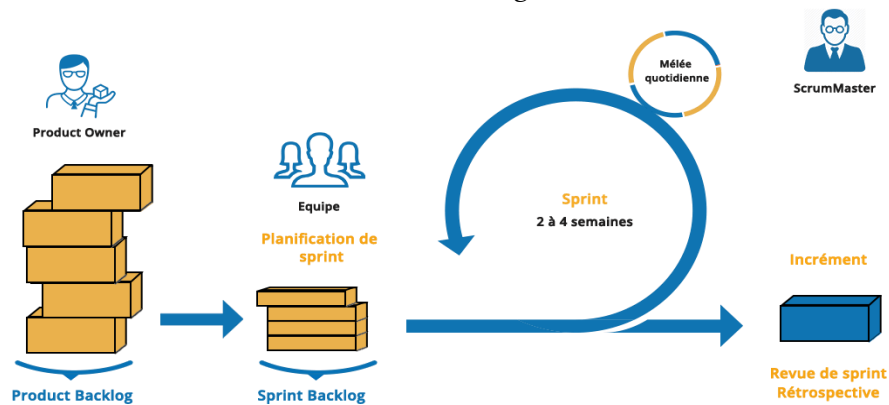


Figure 1 : Scrum

## CHOIX DE LA METHODE DE DEVELOPPEMENT : SCRUM

Après avoir présenté les différentes méthodes de développement, il paraît évident que les méthodes agiles sont mieux adaptées à la conduite de notre projet. Elles permettront ainsi d'éviter les failles causées par les méthodes classiques.

En effet, notre projet est constamment managé à la lumière des quatre variables que sont : délais, coûts, fonctionnalités et qualité. Scrum accorde une importance capitale au partage des connaissances au sein d'une équipe. Ceci est bien entendu un facteur qui améliore la productivité et la capitalisation des savoirs. C'est la raison pour laquelle nous avons adopté la méthode Scrum dans la conduite de ce projet.

Scrum est une méthode à succès quand il s'agit de projets de complexité mineure. La réalisation du reporting relatif à la situation statistique de chaque opération représentera une itération, au

*bout de laquelle, un incrément ou livrable est produit et soumis à l'appréciation du client. Dans la conduite de notre projet, la communication, les feedbacks, le contrôle des livrables sont constants et privilégiés*

## D. UN PROCESSUS DE MODELISATION AVEC UML

### 1. UML (UNIFIED MODELING LANGUAGE)

*UML se définit comme un langage de modélisation graphique et textuel destinée à comprendre et décrire les besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.*

### 2. BASES D'UML

*UML unifie à la fois les notations et les concepts orienté objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. Ce langage est certes issu du développement logiciel mais pourrait être appliqué à toute science fondée sur la description d'un système. Dans l'immédiat, UML intéresse fortement les spécialistes de l'ingénierie système.*

*UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la spécification jusqu'au codage. Dans ce cadre, un concept appartenant aux besoins des utilisateurs projette à sa réalité dans le modèle de conception et dans le codage.*

### 3. DIFFERENT TYPES DE DIAGRAMME D'UML

*UML s'articule maintenant autour des 13 diagrammes différents, dont quatre nouveaux diagrammes introduits par UML 2.0 Chacun d'eux est dédié à la représentation d'un système logiciel suivant un point de vue particulier. Par ailleurs, UML modélise le système suivant deux modes de représentation : l'un concerne la structure de système pris « au repos », l'autre concerne sa dynamique de fonctionnement. Les deux représentations sont*

*nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composants fonctionnent entre eux.*

*La figure suivante présente les différents types de diagramme de l'UML.*

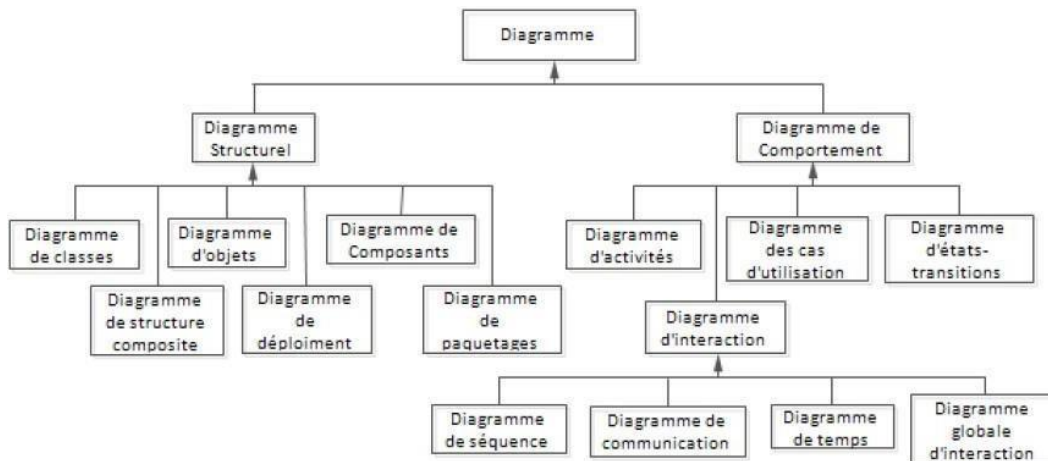


Figure 2 : Différent Type De Diagramme D' Uml

*Dans notre projet nous allons utiliser seulement les diagrammes suivants :*

### 3.1 DIAGRAMME DE CAS D'UTILISATION :

*Un diagramme de cas d'utilisation est la représentation la plus simpliste possible entre un utilisateur et les différents cas d'utilisation auxquels il peut prendre part. Ce diagramme peut permettre d'identifier les potentiels utilisateurs d'un système. Il permet également un point de vue plus simple du système, utile par exemple en communiquant avec un client.*

### 3.2 DIAGRAMME DE SEQUENCE :

*Un diagramme de séquence représente les interactions entre les différents objets qui composent le système. La temporalité est respectée de manière verticale sur le diagramme.*

### 3.3 DIAGRAMME DE CLASSE :

*Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles. Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction.*

## E. CONCLUSION DU CHAPITRE

*Dans ce chapitre nous avons présenté les principales approches et méthodes de développement logiciel. Nous sommes parvenus à porter notre choix sur la méthode Scrum en raison des nombreux avantages qu'elle présente et de son adéquation avec le contexte de notre projet. En fin nous avons fait une présentation du langage UML.*



## CHAPITRE 3 : LES FONCTIONNALITES

*Voici quelques fonctionnalités qu'une application de réservation d'appartements peut offrir à ses utilisateurs:*

*Recherche facile : une fonction de recherche facile à utiliser pour trouver rapidement des appartements disponibles dans la région souhaitée, en fonction du budget et des préférences de l'utilisateur.*

*Filtres de recherche : possibilité d'appliquer des filtres de recherche avancés pour affiner les résultats de recherche en fonction des critères tels que le nombre de chambres, la présence d'une piscine ou d'un balcon, etc.*

*Photos et descriptions détaillées : chaque annonce doit avoir des photos et une description détaillée pour aider les utilisateurs à avoir une idée claire de l'appartement avant de réserver.*

*Calendrier de disponibilité : un calendrier précis des dates de disponibilité pour chaque appartement .*

*Processus de réservation facile : un processus de réservation facile et sécurisé .*

## CHAPITRE 4 : CONCEPTION DE LA SOLUTION

*La démarche de conception est une étape fondamentale dans le processus de développement puisqu'elle fait correspondre la vision applicative (le modèle d'analyse) à la vision technique (l'environnement de développement et d'exécution). [3]*

*Ce chapitre vise à illustrer la phase de conception et les modèles UML associés.*

*Nous commençons par établir les diagrammes de séquence des cas d'utilisation, ensuite nous allons présenter des définitions très utiles pour la compréhension des termes utilisées, une architecture logicielle de la solution, les outils et technologies utilisés et l'architecture technique du système.*

### A. DIAGRAMMES DE SEQUENCES DES CAS D'UTILISATIONS

#### 1. DIAGRAMME DE SEQUENCE

*L'utilisateur doit s'authentifier en saisissant son identifiant et mot de passe, puis le système vérifie les informations introduites, si l'une des coordonnées est non valide le système affiche un message d'erreur sinon il envoie une requête de vérification au serveur qui vérifie l'existence de l'utilisateur dans le SGBD, en cas d'erreur le système affiche un message d'erreur, sinon ce dernier affiche une interface correspondant au rôle de l'utilisateur.*

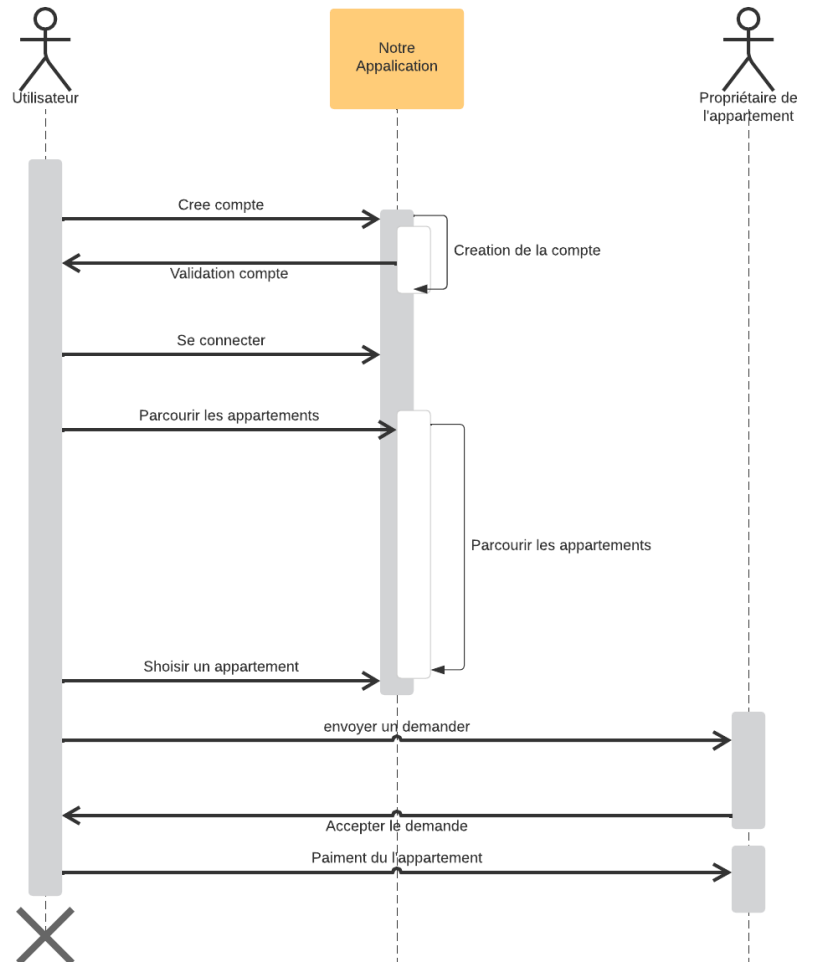


Figure 3 : Diagramme De Séquence

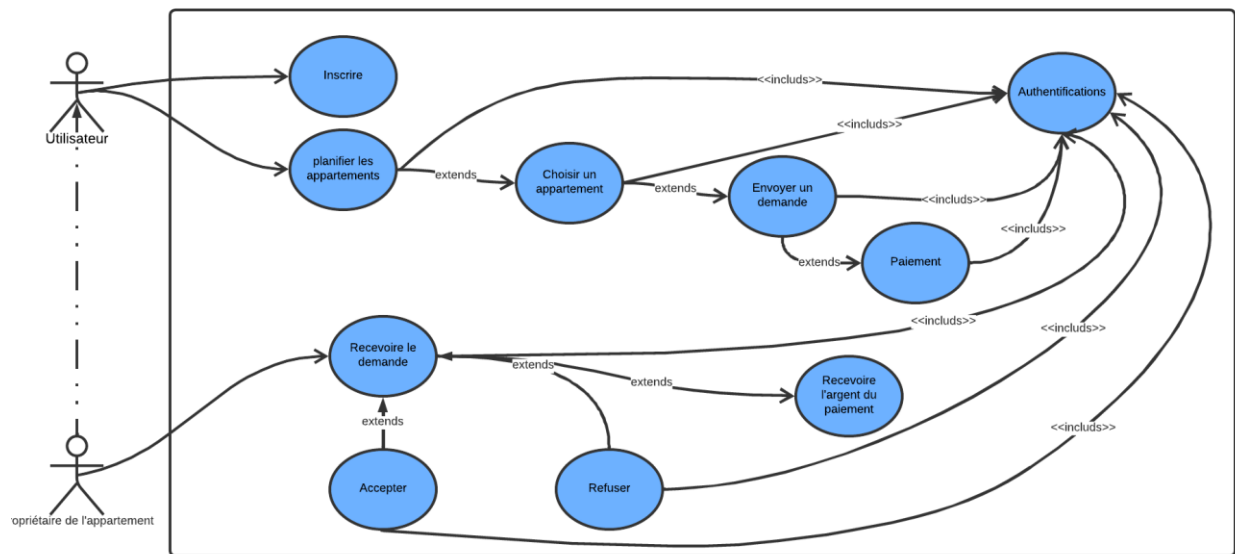


Figure 4 : Diagramme De Cas D'utilisation

Dans ce chapitre, nous allons faire des choix sur le codage. Nous sommes arrivés pratiquement à la fin du processus de développement. Nous commençons par la présentation de l'environnement de développement, les outils et technologies choisis. Enfin nous terminerons par la présentation des réalisations effectuées au cours de ce projet.

### A. OUTILS ET TECHNOLOGIES

#### 1. FLUTTER



Figure 5 : Flutter

*Flutter est un kit de développement de 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. Dans notre cas nous utilisons la partie front end.*

#### 2. DART



Figure 6 : 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.*

### 3. UML

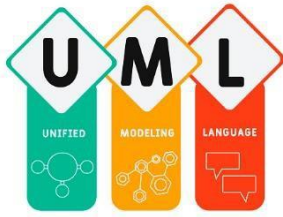


Figure 7 : Uml

*Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement.*

### 4. ANDROID STUDIO



Figure 8 : Android studio

*Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, MacOS, Chrome OS et Linux*

### VISUAL STUDIO CODE

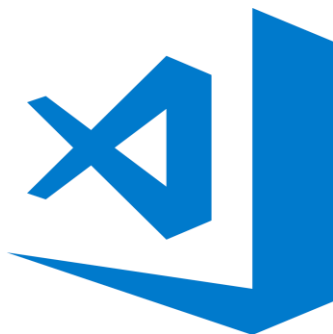


Figure 9 : Visual Studio Code

*Est un éditeur de code simplifié, qui est gratuit et développé en open source par Microsoft. Il fonctionne sous Windows, Mac OS et Linux*

## FIREBASE



Figure 10 : Firebase

*Firebase est une plateforme développée par Google pour créer des applications mobiles et web. C'était à l'origine une société indépendante fondée en 2011. En 2014, Google a acquis la plateforme et c'est maintenant leur offre phare pour le développement d'applications.*

## B. INTERFACES CLIENT

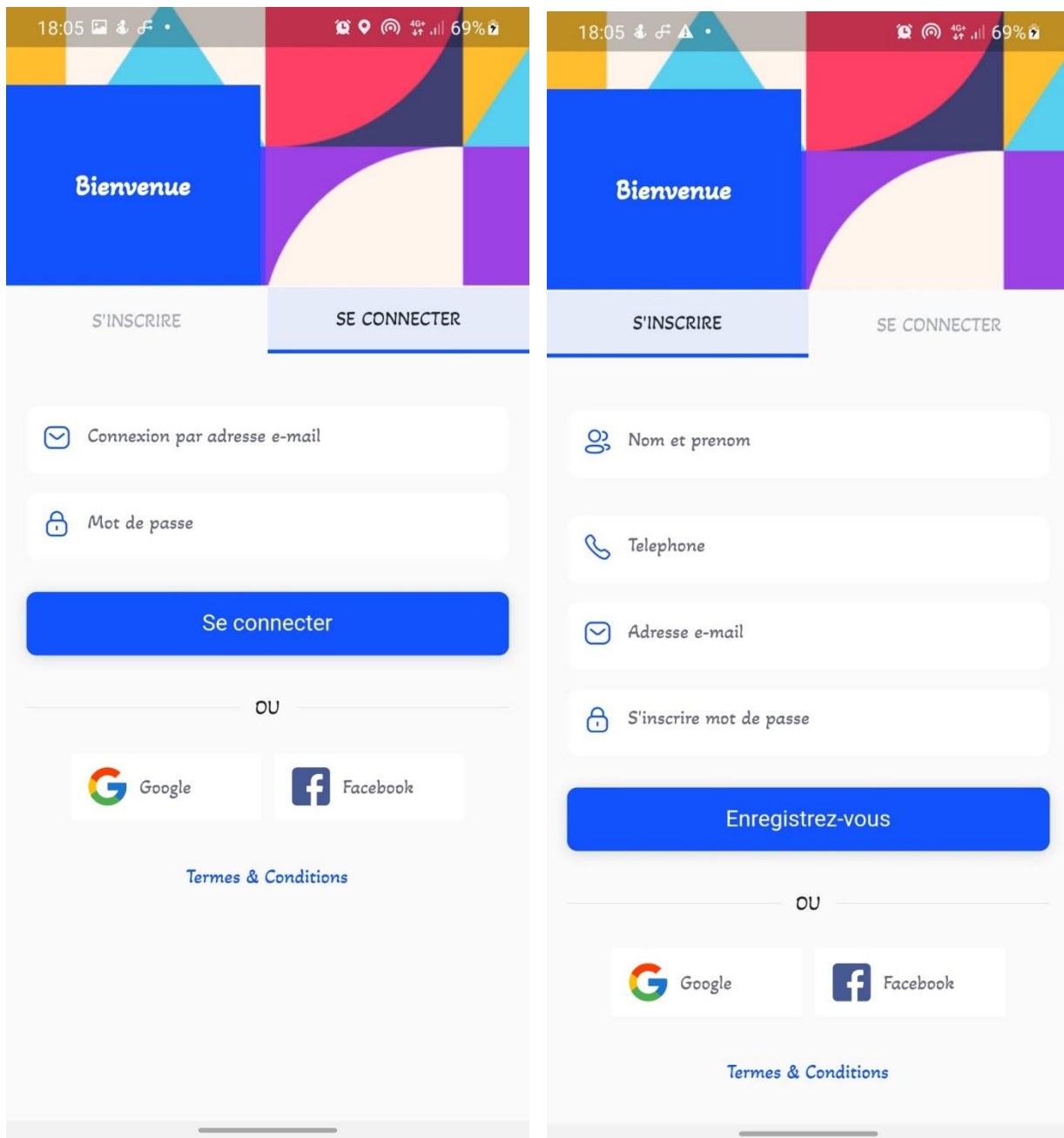
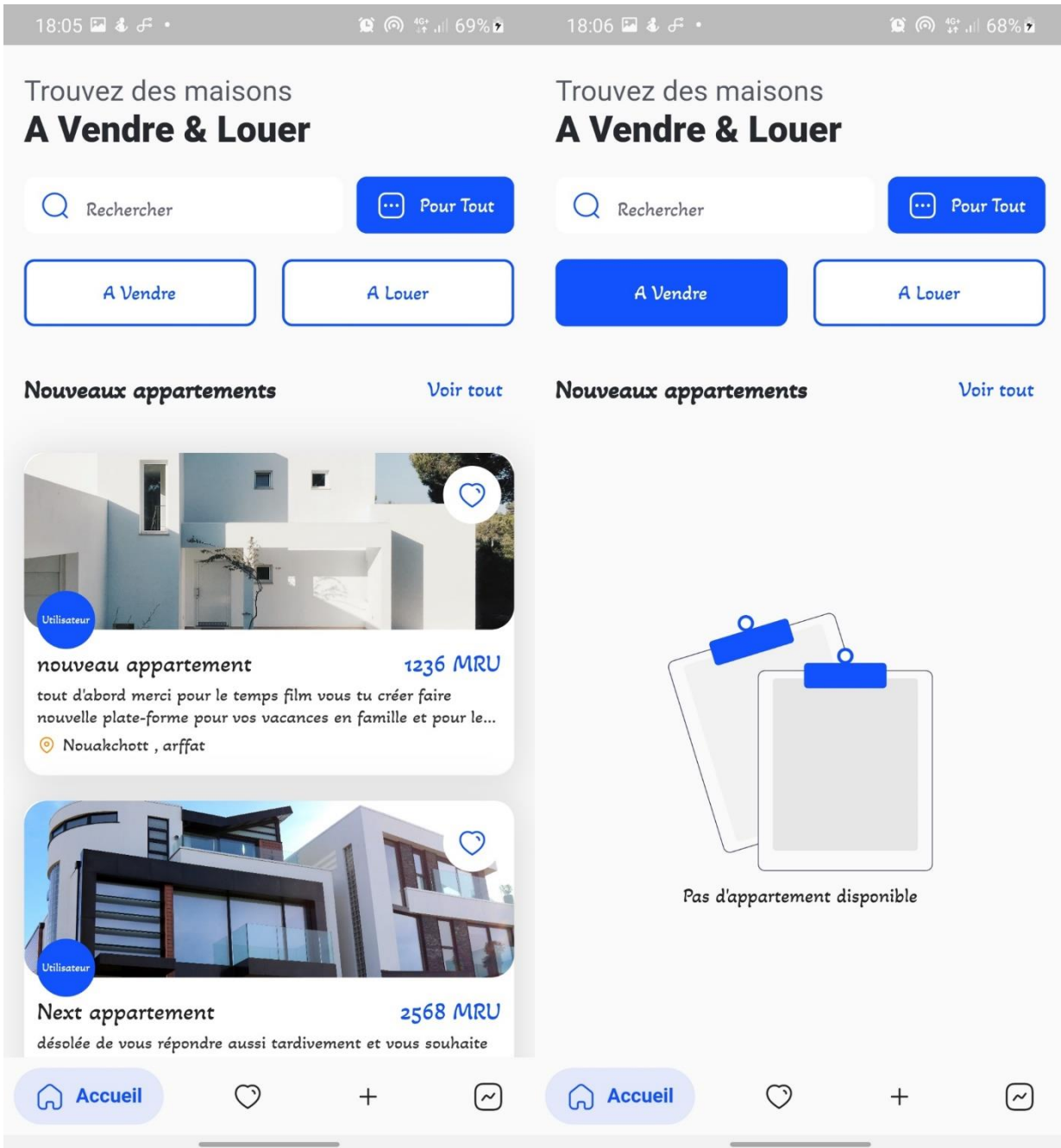
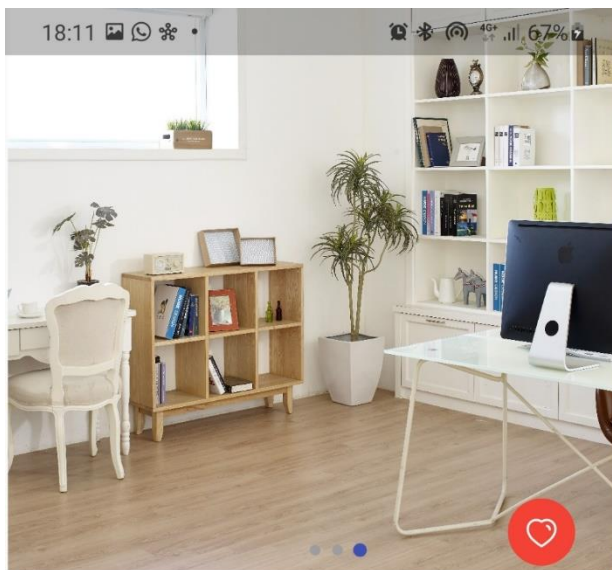


Figure 11 : Interface Client







### nouveau appartement

📍 Nouakchott , arffat [Voir Sur Maps](#) 1236 MRU



3

7

2

#### Discreption

tout d'abord merci pour le temps film vous tu créer faire nouvelle plate-forme pour vos vacances en famille et pour les autres



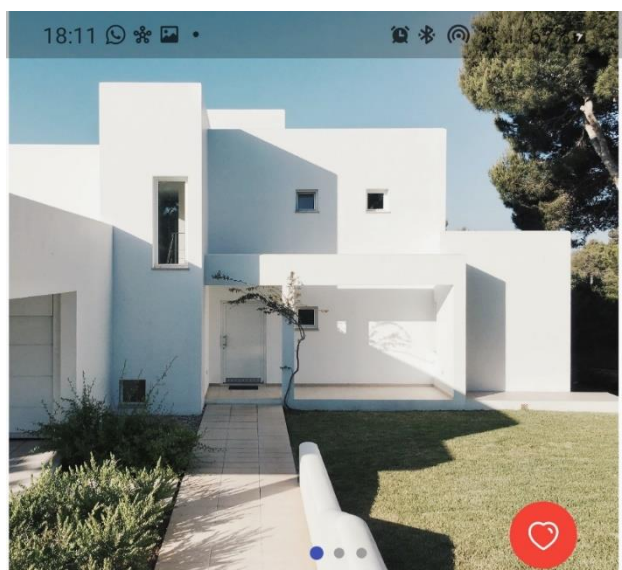
test2

test2@gmail.com

Envoyer un message

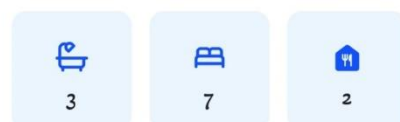


Appel



### nouveau appartement

📍 Nouakchott , arffat [Voir Sur Maps](#) 1236 MRU



3

7

2

#### Discreption

tout d'abord merci pour le temps film vous tu créer faire nouvelle plate-forme pour vos vacances en famille et pour les autres



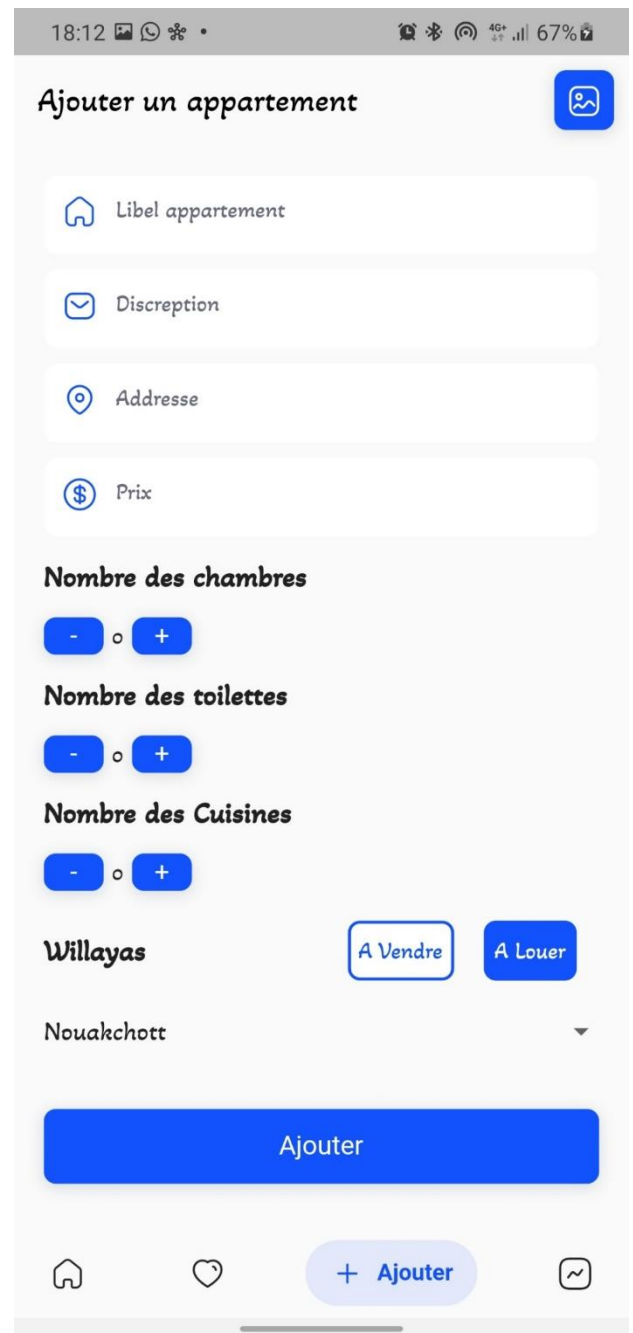
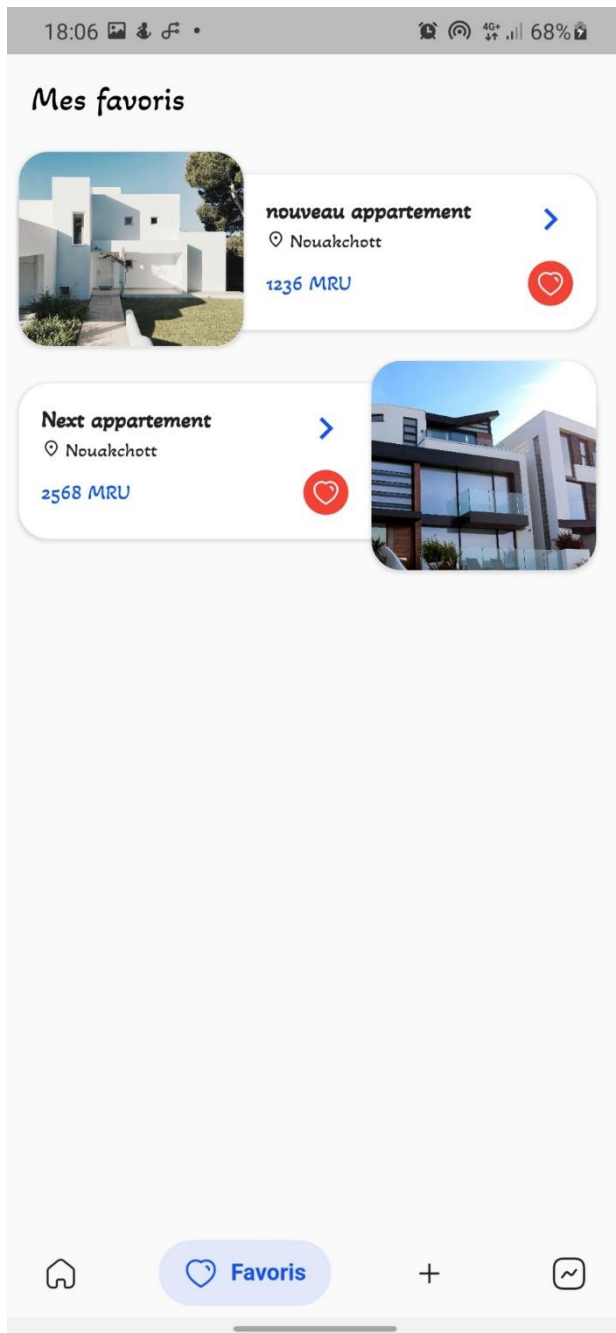
test2

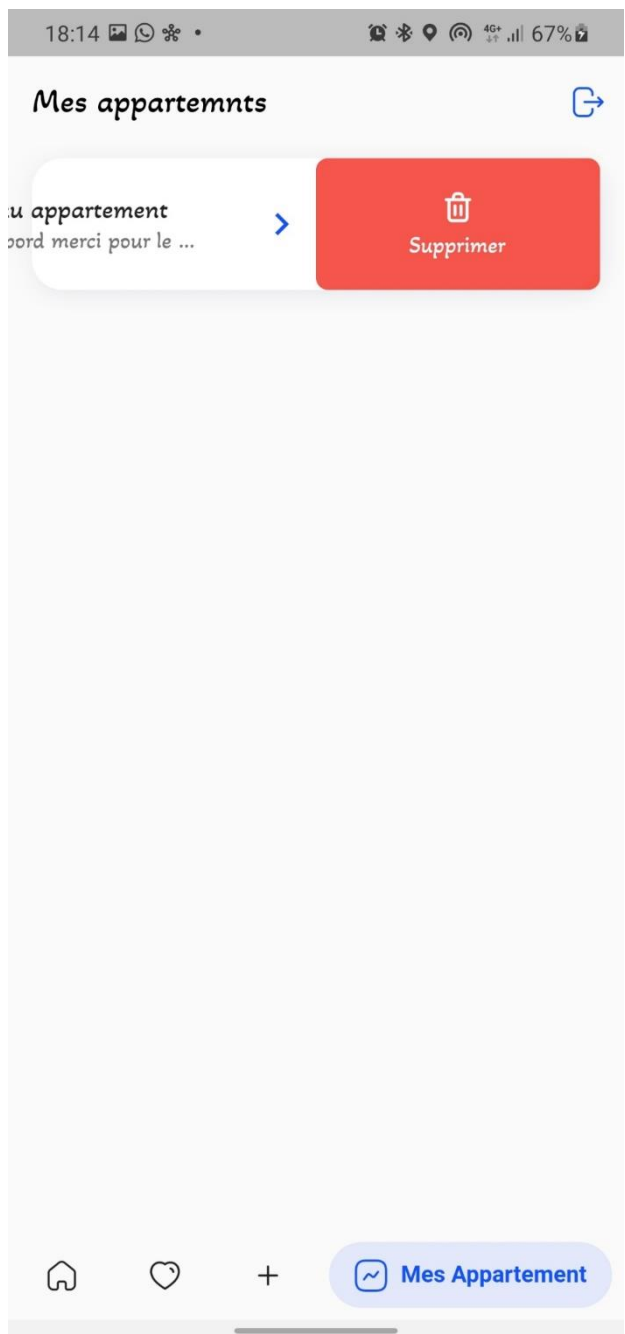
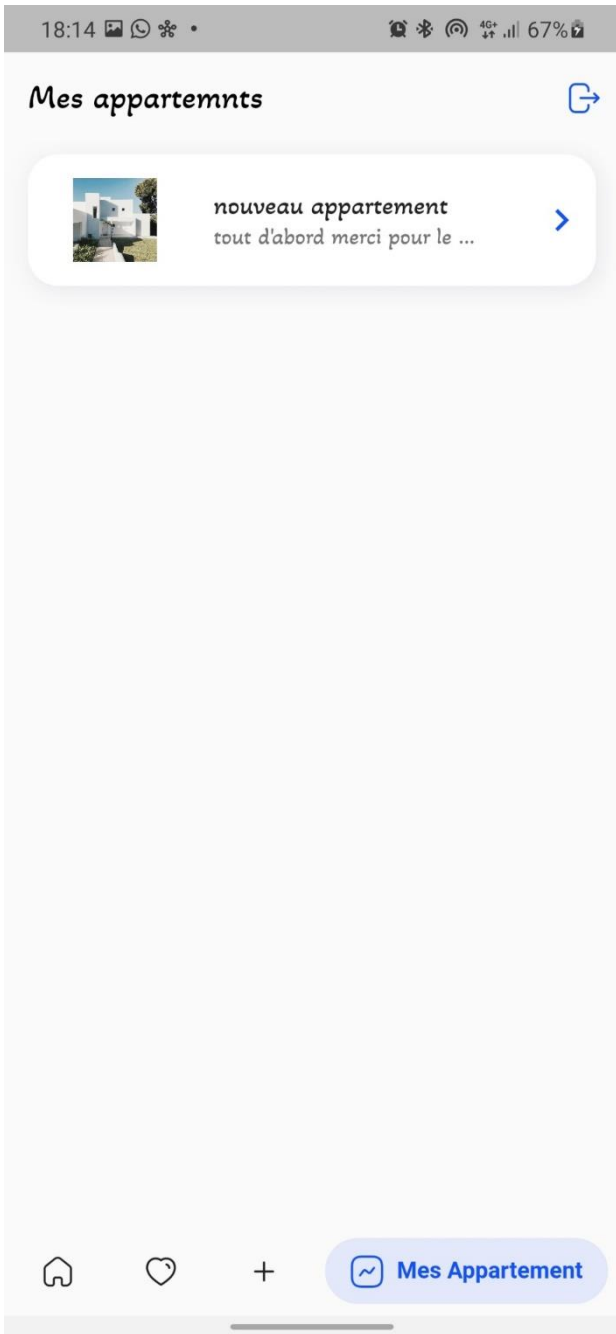
test2@gmail.com

Envoyer un message



Appel





## C. ADMIN

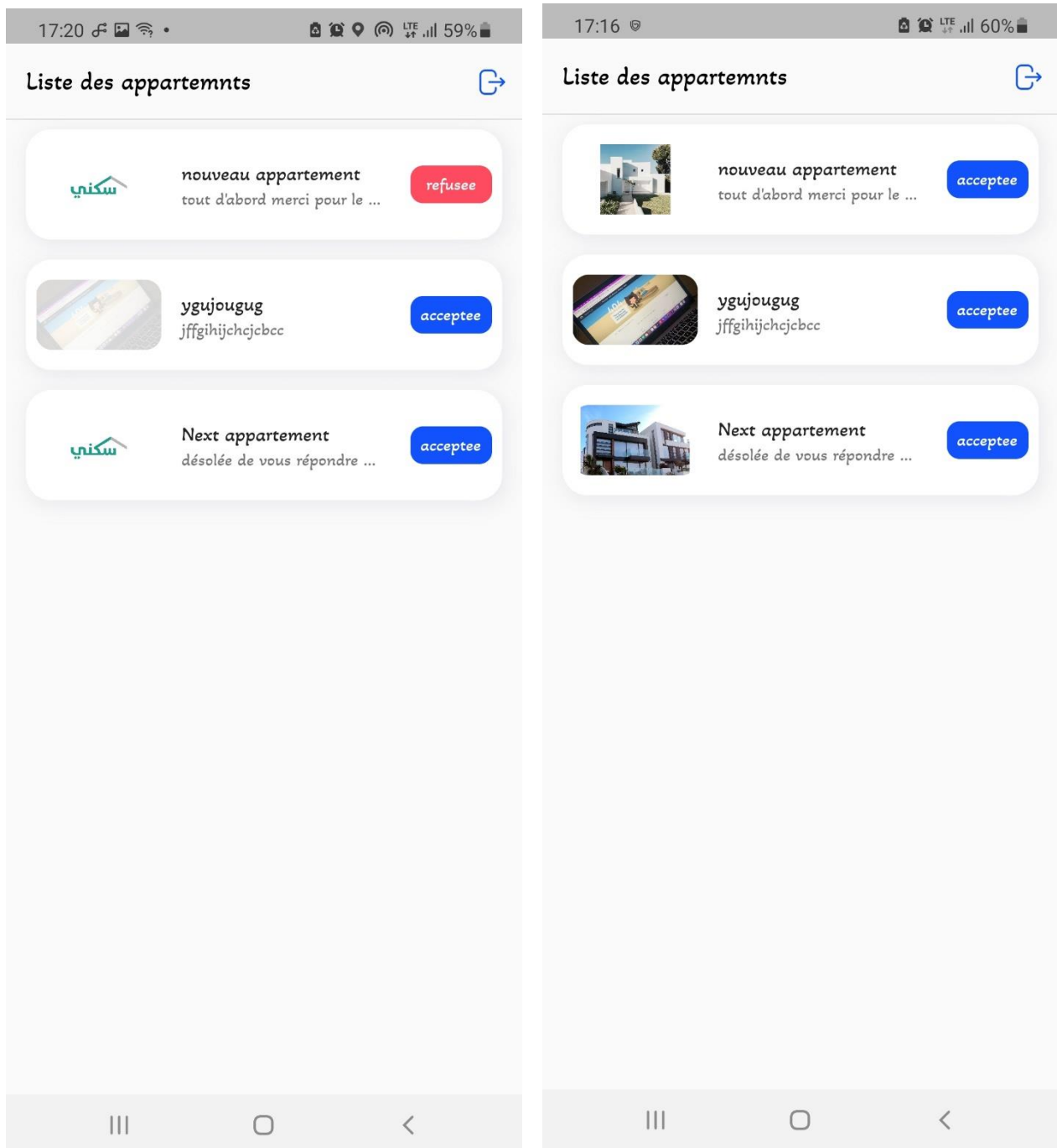
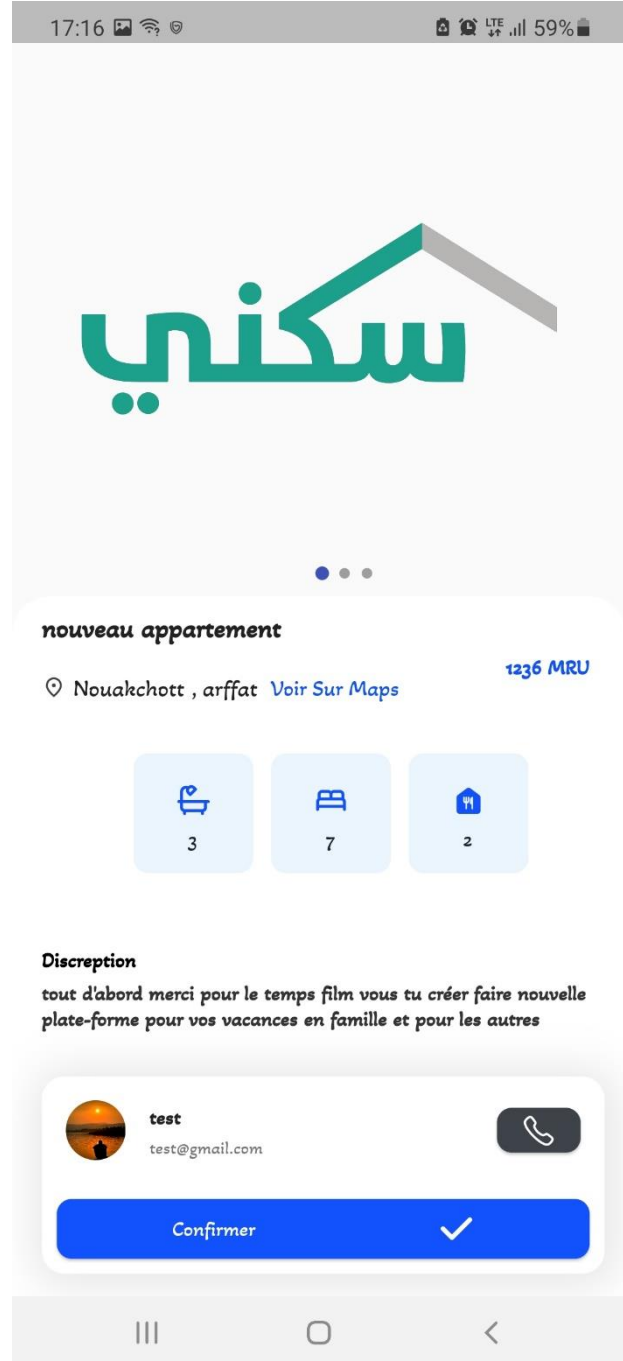
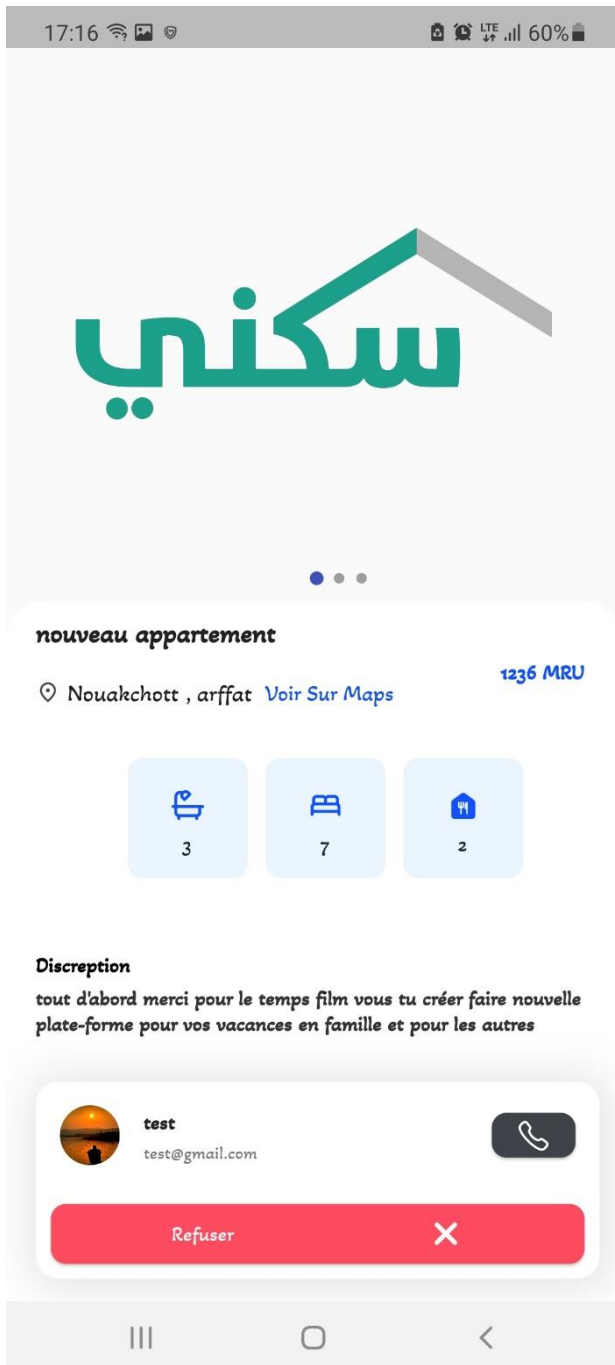


Figure 12 : Interface Admin



## CONCLUSION DU CHAPITRE

*La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application. Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement, les outils et technologies associée à notre système. En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.*

## CONCLUSION ET PERSPECTIVES

*Dans le cadre de notre projet de fin de formation, nous avons développé et réalisé une application mobile qui intègre une solution portable permettant à l'utilisateur de trouver rapidement une solution à son problème.*

*Nous avons d'abord présenté le travail en général ainsi que le contexte, les enjeux et les objectifs du projet. Dans la deuxième partie, nous avons implémenté les différentes étapes du processus de développement d'UP, afin d'implémenter notre solution au problème.*

*Nous avons commencé par identifier les besoins que nous avons ensuite modélisés sous forme de diagrammes de cas d'utilisation, de diagrammes de séquence, et avons terminé par la réalisation de notre application.*

*Au cours de ce travail, nous avons pu appliquer les connaissances théoriques acquises lors de notre formation.*

*Ce projet fait l'objet d'une expérimentation intéressante, il nous a permis de nous rendre compte qu'un projet de création d'application mobile multiplateforme est un ensemble de plusieurs plans planifiés et interdépendants. Toutes les phases de ce projet nous permettent d'enrichir notre expérience, notamment dans les différents outils et techniques dédiés à la programmation des appareils mobiles.*

*Nous avons également soutenu que la réalisation d'une application mobile nécessite une bonne organisation et une cohérence entre les différents acteurs du projet. Parce que tout travail informatique a toujours été un travail d'équipe.*

*Bien que notre application ne soit pas encore terminée, nous veillons à la compléter et à l'améliorer en termes de design (ergonomie), certaines fonctionnalités comme l'optimisation du temps, la gestion des statistiques, le Play Store et l'App Store afin que nous puissions l'utiliser.*



## REFERENCES

1. <https://flutter.dev/> (dernière consultation le 05/06/23).
2. <https://dart.dev/> (dernière consultation le 05/06/23).
3. <https://www.uml.org/> (dernière consultation le 05/06/23).
4. <https://www.latex-project.org/> (dernière consultation le 05/06/23).
5. <https://developer.android.com/> (dernière consultation 05/06/23)