

République Algérienne Démocratique et Populaire
Université Abou Bekr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'études Pour l'obtention du diplôme de Master en
Informatique

Option : Génie Logiciel (G.L)

Thème

**Développement et réalisation d'une application mobile
d'échange de services « Echange + »**

Réalisé par :

- TALEB Amel
- ZEROUALI Nawel

Présenté le 29 juin 2025 devant le jury composé de :

- | | |
|----------------------------|----------------|
| - BEKKOUCHE Amina | (Président) |
| - Benmansour Asma | (Examinatrice) |
| - Benmansour Fassila Lemya | (Encadrante) |
| -Ethchiali abdelhak | (Expert I2E) |

Remerciements

Tout d'abord, nous tenons à exprimer notre profonde gratitude à ALLAH Tout-Puissant, qui nous a accordé la force, la patience et la persévérance nécessaires à la réalisation de ce mémoire.

Nous adressons nos sincères remerciements à notre encadrante, Mme BENMANSOUR Fassila Lemya, pour son précieux soutien, ses conseils avisés et sa présence constante tout au long de la réalisation notre projet de fin d'études.

Nous exprimons également notre sincère gratitude aux membres du jury, Mme BEKKOUCHE Amina, Mme BENMANSOUR Asma, Mr EICHIALI Abdelhak pour le temps précieux qu'ils ont consacré à l'évaluation de notre mémoire et pour leurs précieux commentaires.

Nous sommes reconnaissants à Mr SOULEIMAN Soufiane, directeur de l'incubateur du centre I2E, pour les formations de qualité qu'il nous a offertes dans le cadre de notre projet innovant.

Nous remercions tous les enseignants du département d'informatique pour les efforts qu'ils ont déployés et les connaissances qu'ils nous ont transmises tout au long de notre parcours universitaire.

Nous sommes également reconnaissants à nos parents et à nos frères et sœurs, dont l'amour, le soutien moral et les encouragements ont été une source constante d'inspiration.

Enfin, nous tenons à exprimer notre sincère gratitude à nos amis pour leur présence, leur aide et leur soutien tout au long de cette phase importante de notre démarche.

Dédicace

Je dédie ce mémoire à toutes les personnes, famille, amis et encadrants, dont le soutien et la bienveillance m'ont permis d'atteindre cet objectif.

A mes très chers parents, Latifa et Mustafa, les mots me manquent pour exprimer ma gratitude, mon amour et ma reconnaissance. Ils ont été mon véritable soutien et mon plus grand encouragement tout au long de mon parcours universitaire, en particulier ma mère, qui m'a accompagné à chaque étape. Je leur dois tous les succès que je remporte aujourd'hui.

A mes chers frères, Abdessamed, Abderrahmene et à ma belle-sœur, Feirrouz.

A TALEB Amel, ma binôme, sœur et très chère amie à la fois, que je n'oublierai jamais pour les encouragements qu'elle m'a apportés et son aide toute particulière durant les quatre années que nous avons passées ensemble, qui ont été vraiment profondes. Vraiment, elle a eu un impact profond sur moi. Et sa petite famille.

A notre encadrante, Mme BENMANSOUR Fassila Lemya, Pour l'aide et les conseils qu'elle nous a donnés, ainsi que pour son suivi régulier tout au long du processus de PFE. Son accompagnement attentif et ses remarques utiles ont eu une grande influence sur le résultat final de ce mémoire.

A mon enseignante préférée, FEKIH Siham. Elle était non seulement une enseignante pour moi, mais aussi une mère attentionnée qui soutenait et encourageait sincèrement sa fille. Elle m'a profondément marqué.

A tous les membres de ma famille, et en particulier à mon oncle Aboubaker, pour sa générosité, ses conseils et ses encouragements constants. Il a toujours été un soutien fidèle et sa présence a eu un impact profond sur mon parcours.

À mes deux chers amis, BENDEHINA Bouchra, qui m'a soutenu tout au long de mon parcours universitaire avec soutien et assistance, et ISLI Imane, avec qui j'ai été rapproché par le Coran et l'université. Et ses petites familles.

À tous mes amis et connaissances et surtout Tesnim et Yasser pour ces aides,

À tous ceux qui m'aiment et que j'aime.

ZEROUALI Nawel

Dédicace

Je dédie ce mémoire à toutes les personnes qui ont marqué mon parcours, soutenu mes rêves et accompagné mes efforts.

À ma binôme Nawel, qui a été bien plus qu'une partenaire de travail une véritable sœur dans ce monde parfois difficile. Merci pour ta patience, ton soutien indéfectible et ta présence rassurante.

À Imen, cette main tendue qui ne manque jamais de m'aider, surtout dans les moments où je vacille, malade ou perdue. Tu es l'une des personnes les plus gentilles que j'ai jamais connues.

À Aya, ma petite sœur, dont le sourire est une force silencieuse qui m'encourage à continuer, à me relever et à ne jamais abandonner.

À Rayane, mon petit frère, source d'espoir et d'idées lumineuses malgré son jeune âge. Et à mon frère Abderrahmane, dont le courage m'inspire.

À Mes Parents, pour leurs sacrifices, leur foi en moi et leur investissement dans mes études. Ce mémoire est aussi le fruit de leur amour et de leur persévérance.

À notre encadrante, pour sa patience, son accompagnement précieux et ses efforts constants.

À Monsieur Chikh, qui a su transformer notre perception du management de projet lors du Master 1 une vraie révélation.

À Monsieur Yasser, notre créateur passionné, dont la vidéo exceptionnelle a marqué nos esprits.

Enfin, à toute notre promotion. Merci de m'avoir appris que rien n'est impossible, que nous sommes capables de nous surpasser et de transformer la douleur et l'échec en moteur de réussite, grâce à l'esprit de défi que nous avons partagé.

We love the pain, We live the pain.

Taleb Amel

Table des matières

<i>Liste des figures</i>	I
<i>Liste des tableaux</i>	III
<i>Liste des acronymes</i>	IV
Introduction générale	1
La Problématique	2
Les Objectifs	2
Le Plan du Mémoire	3

Chapitre I : Les applications mobiles

I.1. Introduction	5
I.2. Présentation des Applications	5
I.2.1. Définition des Applications	5
I.2.2. Les Fonctionnalités des Applications	6
I.2.3. Les principales caractéristiques des applications	6
I.2.4 Les avantages et les inconvénients d'une application	7
I.3. Types des Applications	8
I.3.1. Applications Natives	8
I.3.2 Applications Web	10
I.3.3 Applications Hybrides	13
I.3.4. Les applications multiplateformes	15
I.4. L'Architecture d'une Application	16
I.4.1. Couche d'interface utilisateur	17
I.4.2. Couche de données	18
I.4.3. Couche de domaine	19
I.5. Organisation Modulaire d'une Application Mobile	20
I.5.1. Modules de données	20
I.5.2. Modules de fonctionnalités	21
I.5.3. Modules d'applications	21
I.5.4. Modules courants	21
I.5.5. Modules de test	21
I.6. Conclusion	22

Chapitre II : Analyse et conception de l'application « Echange+ »

II.1. Introduction	23
II.2. Etude de l'existant	23
II.3 Spécification des besoins	27
II.3.1 Les acteurs de l'application	28
II.3.2 Les besoins du client	29
II.3.2.1 Besoins liés à l'utilisateur	29
II.3.2.2 Besoins liés à l'administrateur	30
II.3.3 Les exigences fonctionnelles	30
II.3.4 Les exigences non fonctionnelles :	33
II.4 Modélisation des besoins fonctionnels	34
II.4.1 Diagrammes de cas d'utilisation	35
II.4.1.1 Le diagramme de cas d'utilisation de l'administrateur	35
II.4.1.2 Le diagramme de cas d'utilisation de l'utilisateur	36
II.4.2 Diagrammes de séquence	37
II.4.2.1 Diagramme de séquence des cas d'utilisation de l'administrateur	37
II.4.2.2 Diagramme de séquence des cas d'utilisation de l'utilisateur	39
II.4.3 Base de données NoSQL :	43
II.4.3.1 Définition :	43
II.4.3.2 Les caractéristiques de base de données NoSQL	43
II.4.3.3 Structure de la base de données de l'application Echange+	44
II.5 Conclusion	48

Chapitre II : Réalisation et implémentation de l'application « Echange+ »

III.1 Introduction	50
III.2. Architecture de notre application	50
III.2.1. Définition du modèle MVVM	50
III.2.2. Structure du modèle MVVM	51
III.3. Structure de notre application :	51
III.4. Processus de développement	53
III.4.1. Définition de la méthode SCRUM	54

III.5. Diagramme de Gantt	54
III.6. Les Outils et les langages utilisés	55
III.7. Défis Du Projet	56
III.8. Présentation de notre application “Echange +”	59
III.9. Conclusion	72

Chapitre IV : Business Model Canvas

IV.1. Introduction	73
IV. 2. Les piliers du Business Model Canvas :	73
IV. 2.1. Propositions de Valeur (Value Propositions)	73
IV. 2.2. Segments de clients (Customer Segments)	74
IV. 2.3. Relations avec les clients (Customer Relationships)	75
IV. 2.4. Canaux (Channels)	76
IV. 2.5. Partenaires clés (Key Partners)	77
IV. 2.6. Activités clés (Key Activities)	78
IV. 2.7. Ressources clés (Key Resources)	79
IV. 2.8. Charges et coûts (Cost Structure)	80
IV. 2.9. Revenus (Revenue)	81
IV. 3. Business Model Canvas :	84
IV. 4. Conclusion :	84
Conclusion Générale	85
Bibliographie	88

Liste des figures

Figure 1: Schéma d'une architecture d'application classique.....	17
Figure 2: Rôle de la couche d'interface utilisateur dans l'architecture de l'application .	18
Figure 3: Rôle de la couche de données dans l'architecture de l'application.....	19
Figure 4: Rôle de la couche de domaine dans l'architecture de l'application.....	20
Figure 5: plateforme Simbi.....	24
Figure 6: Application Fiverr.....	24
Figure 7: Application Youchoz	25
Figure 8: Application Youchoz Pro	26
Figure 9: Diagramme de cas d'utilisation des tâches de l'administrateur	36
Figure 10: Diagramme de cas d'utilisation des tâches de l'utilisateur	37
Figure 11: Diagramme de séquence du cas d'utilisation « vérifier la conformité des annonces »	38
Figure 12: Diagramme de séquence du cas d'utilisation « superviser les échanges ».....	39
Figure 13: Diagramme de séquence du cas d'utilisation « publier une annonce »	40
Figure 14: Diagramme de séquence du cas d'utilisation « faire un échange »	42
Figure 15: Caractéristiques de base de données NoSQL [24]	43
Figure 16: Collections « achat_point », « categories »	44
Figure 17: Collection « annonces », « echange_argent »	44
Figure 18: Collection « echange_points », « echange_service »	45
Figure 19: Collection « exchange_proposals ».....	45
Figure 20:Collection « notification »	46
Figure 21: Collection « platform_commissions », « platform_commissions_argent », « reports_probleme»	47
Figure 22: Collection « reports_user », « user_ratings »	48
Figure 23: Collection « users »	48
Figure 24: Interaction (Model — View — ViewModel) dans le cadre du pattern MVVM [24]	50
Figure 25: Architecture de notre application “Echange+”	52
Figure 26: Logo FlutterFire.....	52
Figure 27: Logo Firebase Authentification.	53
Figure 28: logo Firebase Cloud Firestore.....	53
Figure 29: Diagramme de Gantt de l'application ” Echange+”.....	55
Figure 30: Logo de l'application mobile “Echange +”	59
Figure 31: Interfaces de connexion et d'inscription ”Echange+”.....	60
Figure 32: Interfaces de gestion des propositions et demandes utilisateur.	60
Figure 33: Interfaces de Recherche et proposition d'Échange	61
Figure 34: Interfaces de profil utilisateur et gestion des annonces.....	62
Figure 35: Interfaces d'échange service par service	63
Figure 36: Interfaces d'échange service par points.....	64
Figure 37: Interfaces d'achat de points par l'argent	64
Figure 38: Interfaces de Proposition, Notification et Suivi d'Échanges Monétaires.....	65

Figure 39: Interfaces de finalisation d'échange service par argent et d'évaluation Utilisateur	66
Figure 40: Interfaces de recherche et de signalement utilisateur	66
Figure 41: Gestion des utilisateurs et signalements (Panneau Admin)	67
Figure 42: Panneau administrateur - Supervision des échanges et calendrier	68
Figure 43: Panneau administrateur - Statistiques, revenus et services	69
Figure 44: Panneau administrateur - Gestion des annonces et commentaires	70
Figure 45: Panneau administrateur - Gestion des signalements et problèmes	70
Figure 46: Panneau administrateur - Gestion complète	71
Figure 47: Le modèle économique et sociale de l'application "Echange+"	84

Liste des tableaux

Tableau 1: Exemples d'applications mobiles 100% natives	8
Tableau 2: Exemples d'applications 100% Web.....	11
Tableau 3: Exemples d'applications mobiles 100% hybrides	13
Tableau 4: Exemples d'applications 100% multiplateformes	15
Tableau 5: Tableau comparatif entre les applications de services	27

Liste des acronymes

LES ACRONYMES	SIGNIFICATIONS
ACID	Atomicity, Consistency, Isolation, Durability
API	Application Programming Interface
BMC	business model Canvas
BTP	Bâtiment et Travaux Publics
CSS	Cascading Style Sheets
FAAS	Function as a Service
FAQ	Frequently Asked Questions
FCM	Firebase Cloud Messaging
GPS	Global Positioning System
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
MVVM	Model – View – ViewModel
NOSQL	Not Only SQL
OS	Operating System
REST	Representational State Transfer
SAAS	Software as a Service
SDK	Software Development Kit
SMS	Short Message Service
SQL	Structured Query Language
SQLITE	Structured Query Language – Lite
UI	User Interface
UML	Unified Modeling Language
URL	Uniform Resource Locator
UX	User Experience
WEBRTC	Web Real-Time Communication

Introduction Générale

Introduction générale

Les technologies mobiles, devenues incontournables dans notre quotidien, offrent des solutions accessibles et pratiques pour simplifier une multitude de tâches, telles que la communication, la gestion des services, ou encore l'accès à l'information. Parmi ces usages, les applications mobiles d'échange entre particuliers connaissent un véritable essor, portées par une volonté croissante de consommer autrement, de partager et de valoriser les ressources existantes.

Ces plateformes d'échange représentent une alternative intéressante, notamment dans des contextes où les moyens financiers sont limités. En utilisant d'autres modes de transaction tels que l'échange direct, un système de points convertibles en argent ou des paiements électroniques, offrant ainsi une alternative souple et adaptée aux besoins des utilisateurs. En parallèle, des fonctionnalités telles que la géolocalisation, les évaluations et les notifications viennent renforcer la confiance entre utilisateurs et améliorer l'expérience globale. De plus, chaque échange permet aux utilisateurs d'accumuler des points, qui, une fois convertis en argent, ajoutent une dimension économique attractive et motivante pour encourager la participation active au sein de la plateforme.

C'est dans cette dynamique que s'inscrit notre projet de développement d'une application mobile dédiée à l'échange de services entre utilisateurs. L'application Echange+ offre une plateforme intuitive permettant aux utilisateurs de publier des annonces, consulter les offres disponibles, effectuer des demandes d'échange et finaliser les transactions par différents moyens : l'échange, points ou paiement via Edahabia. Des fonctionnalités complémentaires, telles que la géolocalisation, les notifications en temps réel et l'évaluation des prestataires, enrichissent l'expérience utilisateur et assurent la fiabilité du système.

L'objectif principal de ce mémoire est de présenter les différentes étapes de conception, de développement de l'application Echange+, en mettant en lumière les choix techniques, fonctionnels et ergonomiques effectués. Nous analyserons également l'impact de cette solution sur le comportement des utilisateurs et son potentiel à répondre à des besoins locaux spécifiques, dans un contexte où l'échange et la solidarité prennent une dimension de plus en plus importante.

La Problématique

Face à un contexte économique où les inégalités s'accroissent et où les contraintes budgétaires touchent particulièrement les jeunes et les populations vulnérables, le recours à des solutions alternatives comme l'échange de services devient de plus en plus pertinent. Pourtant, ces pratiques restent souvent informelles, peu organisées, et limitées à un cercle restreint, ce qui freine leur portée et leur efficacité.

Face à cette réalité, il devient essentiel de se demander comment les technologies mobiles peuvent être mobilisées pour structurer et dynamiser ces échanges de manière fiable, pratique et accessible à tous. L'enjeu est de proposer une plateforme numérique qui facilite les interactions entre utilisateurs, renforce la confiance et propose des modes de transaction souples (échange, points, paiement mobile), tout en tenant compte des spécificités sociales, culturelles et économiques locales.

Ainsi, la problématique centrale de ce mémoire est la suivante : **Comment concevoir une application mobile intuitive et sécurisée qui permette de valoriser les échanges de services entre particuliers, tout en s'adaptant aux besoins et aux réalités du contexte local algérien ?**

Les Objectifs

L'objectif principal de ce projet est de concevoir et de développer une application mobile appelée Echange+, destinée à faciliter l'échange de services entre utilisateurs, en mettant en avant différentes formes de transaction.

De manière plus spécifique, ce projet vise à :

- Créer une plateforme intuitive qui permet aux utilisateurs de publier et consulter des annonces de services.
- Faciliter la mise en relation entre les utilisateurs intéressés par un échange via un système de demandes et de propositions d'échange.
- Intégrer plusieurs modes de transaction, notamment l'échange direct, un système de points et un paiement électronique.
- Ajouter la géolocalisation pour visualiser les offres disponibles à proximité et améliorer la pertinence des échanges.

- Mettre en place un système de notifications pour informer les utilisateurs en temps réel des nouvelles offres, demandes ou évaluations.
- Offrir une évaluation des prestataires à travers des étoiles et des commentaires, pour garantir la fiabilité et la sécurité des échanges.
- Concevoir une interface simple, moderne et ergonomique, accessible à tout type d'utilisateur.
- Mettre en place un modèle économique viable, à travers des services premium, de la publicité ciblée ou des partenariats, afin de générer des revenus pour les administrateurs de la plateforme.

Avec cette application, notre but principal est d'aider les personnes qui ont peu de moyens, en leur donnant la possibilité d'échanger des services ou des biens facilement, sans forcément utiliser de l'argent. L'application permet aussi de répondre à des besoins du quotidien de façon simple et pratique. En plus de cela, elle peut nous permettre, en tant qu'administrateurs, de générer des revenus grâce à certaines options payantes ou partenariats, ce qui rend le projet utile et durable.

Le Plan du Mémoire

Notre rapport de mémoire est organisé en quatre chapitres, précédé d'une introduction générale et suivi par une conclusion générale :

- **Dans le chapitre 01**, intitulé « **les applications mobiles** », nous présentons les bases des applications mobiles : leur définition, leurs types (natives, web, hybrides, multiplateformes), ainsi que leurs avantages et inconvénients. Nous évoquons également leur architecture générale, basée sur la séparation des couches (interface, logique, données).
- **Dans le chapitre 02**, intitulé « **Analyse et conception du l'application Echange+** », nous identifions les besoins fonctionnels et non fonctionnels du système, définissons les cas d'utilisation et présentons une modélisation UML détaillée afin de planifier la structure de notre application.
- **Dans le chapitre 03**, intitulé « **Réalisation et implémentation de l'application Échange+** », nous décrivons les outils, les technologies et les langages choisis pour le développement. Ce chapitre contient également des captures des interfaces principales, une démonstration des fonctionnalités clés et une discussion sur les résultats obtenus.

- **Dans le chapitre 04**, intitulé « **Business Model Canvas** », nous étudions le modèle économique de l'application. Ce chapitre présente ses fondations à travers les neuf blocs du canevas, tels que la proposition de valeur, les segments de clientèle, les sources de revenus ou encore les partenariats clés. L'objectif est de démontrer la viabilité et la pérennité du projet sur le plan organisationnel et financier. Chaque pilier est analysé pour mieux structurer le fonctionnement futur de la plateforme. Enfin, un tableau synthétique résume visuellement l'ensemble du modèle.

Chapitre I :
Les applications mobiles

I.1. Introduction

L'essor du numérique a profondément transformé notre quotidien, rendant les applications mobiles indispensables dans divers domaines. De la communication à la gestion financière, en passant par la santé et l'éducation, ces outils offrent un accès rapide et personnalisé aux services essentiels, facilitant ainsi la vie des utilisateurs.

Dans ce cadre, les applications mobiles jouent un rôle majeur dans l'évolution des modes de consommation et d'interaction. Elles encouragent la digitalisation des services et favorisent l'économie collaborative en permettant un échange plus fluide de biens et de compétences entre particuliers et professionnels.

Toutefois, cette transformation soulève des défis en matière de sécurité et de protection des données. L'adoption croissante de ces technologies impose de trouver un équilibre entre innovation et respect de la vie privée, afin d'assurer une expérience numérique à la fois performante et sécurisée.

I.2. Présentation des Applications

De nos jours, les applications mobiles jouent un rôle central dans la vie quotidienne, facilitant l'accès à une variété de services. Elles offrent des solutions pratiques et instantanées, allant de la communication à la gestion des finances, en passant par le shopping et la santé. Dans cette section, nous explorerons les différents types d'applications mobiles et leurs caractéristiques essentielles, ainsi que leur impact sur les utilisateurs et les entreprises.

I.2.1. Définition des Applications

Une **application mobile** (ou application pour mobile) est un logiciel développé spécifiquement pour être utilisé sur de petits appareils informatiques sans fil, tels que les **smartphones** et les **tablettes**, plutôt que sur des ordinateurs de bureau ou portables.

Les applications mobiles sont parfois classées en fonction de leur nature :

- **Applications web**, accessibles via un navigateur,
- **Applications natives**, créées spécifiquement pour une plateforme donnée,
- Et une **troisième catégorie**, les **applications hybrides**, qui combinent des éléments des deux types précédents.
- Et une **quatrième catégorie**, les **applications multiplateformes**, qui permettent de créer un seul code fonctionne sur iOS, Android, web.

À l'ère du numérique, les applications mobiles sont devenues **indispensables dans la vie quotidienne** de la plupart des gens. Qu'il s'agisse de **réseaux sociaux**, de **divertissement**, de **productivité** ou d'**activités professionnelles**, elles jouent un rôle essentiel dans la manière dont nous interagissons avec la technologie [1].

I.2.2. Les Fonctionnalités des Applications

Une application mobile est un logiciel conçu pour offrir divers services aux utilisateurs à travers des interfaces interactives et optimisées pour les appareils mobiles [2] :

- **Plateforme mobile-first** : Optimisée pour smartphones/tablettes.
- **Personnalisation** : Logo, couleurs et identité visuelle.
- **Fonction chat** : Communication rapide et sécurisée.
- **Messagerie instantanée** : Échanges discrets et efficaces.
- **Partage de contenu** : Collaboration et centralisation des infos.
- **Notifications push** : Infos urgentes et ciblées en temps réel.
- **Actualités internes** : Suivi de la vie de l'entreprise.
- **Formation en ligne** : Inscription et suivi via mobile.
- **Enquêtes employées** : Feedbacks anonymes et engagement.
- **Système de notifications** : Rappels, félicitations, alertes.
- **Annuaire interne** : Recherche rapide de contacts.
- **FAQ** : Réponses aux questions fréquentes.
- **Modules de formation** : Vidéos, textes, accès mobile.
- **Dashboard commercial** : Classement et motivation des équipes.
- **Recherche intégrée** : Gain de temps, accès rapide à l'info.
- **Compatibilité iOS/Android** : BYOD (Bring Your Own Device).
- **Répertoire de liens** : Accès rapide aux outils essentiels.
- **Authentification unique** : Connexion simplifiée et sécurisée.
- **Multilingue** : Accès aux contenus dans la langue de l'employé.
- **Calendrier d'équipe** : Suivi des tâches et des échéances.

I.2.3. Les principales caractéristiques des applications

Une application mobile se distingue par cinq caractéristiques clés de succès [3] :

- **Design attrayant (UX/UI)**
Interface intuitive, navigation fluide, fonctionnalités accessibles.

- **Évolutivité**
Capacité à s'adapter aux besoins futurs et aux retours utilisateurs.
- **Bon marketing**
Plan de communication solide, ciblage précis, visibilité maximale.
- **Valeur ajoutée**
Résout un problème réel, apporte un bénéfice clair à l'utilisateur.
- **Bon développeur**
Choix de technologies adaptées, partenaires fiables et expérimentés.

I.2.4 Les avantages et les inconvénients d'une application

Les applications offrent de nombreux avantages, mais présentent aussi certains inconvénients selon le contexte d'utilisation [4] :

Avantages :

- **Communication facilitée avec les clients** : Les applications permettent une interaction directe et efficace avec les utilisateurs, améliorant le service client et renforçant la relation client.
- **Accessibilité hors ligne** : Certaines applications offrent des fonctionnalités utilisables sans connexion Internet, augmentant leur praticité pour les utilisateurs.
- **Fonctionnalités spécifiques** : Les applications peuvent intégrer des fonctionnalités natives des appareils mobiles, telles que le GPS ou l'appareil photo, enrichissant l'expérience utilisateur.

Inconvénients :

- **Coûts de développement et de maintenance élevés** : La création et l'entretien d'une application nécessitent des investissements significatifs en termes de temps et d'argent.
- **Compatibilité multiplateforme** : Assurer le bon fonctionnement de l'application sur différents systèmes d'exploitation peut être complexe et coûteux.
- **Mises à jour fréquentes requises** : Les applications doivent être régulièrement mises à jour pour rester compatibles avec les nouvelles versions des systèmes d'exploitation et répondre aux attentes des utilisateurs.
- **Problèmes de sécurité** : Les applications mobiles peuvent être vulnérables aux cyberattaques, nécessitant des mesures de sécurité robustes et une surveillance continue.

I.3. Types des Applications

Une application est un logiciel qui vous permet d'échanger des informations avec des clients et de les aider à accomplir des tâches spécifiques. Les différents types d'applications se distinguent par leur méthode de développement et leur fonctionnalité interne. Les applications web sont diffusées via un navigateur internet. Les utilisateurs n'ont pas besoin de les installer sur leurs appareils. Les applications natives, quant à elles, sont conçues pour une plateforme ou un type de dispositif spécifique. L'utilisateur doit installer la bonne version du logiciel sur l'appareil de son choix. Les applications hybrides sont natives, mais intègrent un navigateur web [5]. Enfin, les applications **multiplateformes** sont conçues pour fonctionner sur plusieurs systèmes avec un seul code source, facilitant la maintenance. Chaque type présente des avantages selon le contexte d'utilisation et les attentes des utilisateurs.

I.3.1. Applications Natives

Une application native est une application mobile conçue spécifiquement pour un système d'exploitation particulier utilisé par les smartphones et les tablettes, tels qu'iOS, Android, ect. Les applications natives sont développées en utilisant des langages de programmation natifs, tels qu'Objective-C pour iOS.

Les applications natives sont généralement plus performantes que les applications web ou hybrides, car elles sont conçues pour utiliser directement les ressources de l'appareil, au lieu de passer par un navigateur ou un conteneur. Les développeurs ont également plus de contrôle sur l'apparence et les fonctionnalités de l'application, ce qui permet une meilleure expérience utilisateur [6]. Le tableau 1 suivant donne trois exemples d'applications natives ainsi que leurs caractéristiques techniques.

Application	Plateforme	Langage	100% Natif ?	Pourquoi ?
Apple Mail	iOS	Swift / Obj-C	✓ Oui	APIs iOS uniquement, intégrée au système Apple
Google Fit	Android	Kotlin / Java	✓ Oui	Capteurs Android, Google Services, pas de code partagé
Samsung Notes	Android	Kotlin / Java	✓ Oui	Spécifique à Samsung, fonctionnalités matérielles natives

Tableau 1: Exemples d'applications mobiles 100% natives

Les applications natives présentent certains avantages et quelques inconvénients [7] :

Avantages :

- **Le natif pour une expérience utilisateur optimale**

Selon une étude menée par Oracle, 55 % des utilisateurs abandonneraient une application mobile si l'expérience de navigation est médiocre.

Les applications natives sont créées pour une plateforme précise. Elles sont donc optimisées pour cette dernière. Il en ressort donc un niveau de performance bien accru. La compilation des applis mobiles natives se faisant avec des API et un langage de programmation de base de l'OS spécifique, **elles sont plus réactives et surtout plus rapides**. Les plateformes que sont Android et iOS mettent par ailleurs davantage l'accent, sur le User Experience (UX) et le User Interface (UI), pour un environnement graphique riche.

Les applications mobiles natives exploitent enfin, l'identité du système d'exploitation auquel elles sont destinées, pour garantir une expérience utilisateur optimale. Elles suivent des directives qui enlèvent d'ailleurs la courbe d'apprentissage de l'utilisateur.

Un possesseur d'iPhone qui télécharge une application iOS ne souhaite pas se familiariser avec des gestes d'un autre OS comme Android par exemple.

- **Le natif, pour un accès aux fonctionnalités natives du téléphone**

Pendant le développement d'une application native, les éditeurs ont directement accès aux fonctionnalités natives du téléphone telles que la caméra, l'accéléromètre, le GPS, le statut de la batterie, la géolocalisation et bien d'autres. Ils peuvent ainsi donner vie à des fonctionnalités avancées, dans leurs apps. Les ressources de l'application peuvent par ailleurs être stockées dans le téléphone quand il s'agit du natif. L'appli peut donc fonctionner aisément même sans accès permanent à internet.

- **Le natif, pour moins de bugs pendant le développement et plus de sécurité**

Les applications mobiles natives intègrent la gestion numérique des droits (GND). Cela s'avère être une nécessité pour le stockage en toute sécurité des données sur le téléphone, l'affichage de notifications push hors application ou encore l'accès à plusieurs services multimédias.

Contrairement aux WebApps, qui reposent par exemple sur des technologies comme le CSS, le HTML ou JavaScript, les applis natives garantissent une meilleure fiabilité de la protection des données.

Ses outils de développements n'étant pas multiplateformes, il y a par ailleurs **moins de dépendances, donc moins de bugs, qu'avec les apps hybrides**. La sortie de nouvelles versions des systèmes d'exploitation mobiles donne lieu généralement à la publication de nouveaux kits de développement. Ces derniers permettent aux éditeurs d'applications mobiles natives d'implémenter les nouvelles fonctionnalités dans leurs créations dans un délai bien plus réduit.

Inconvénients :

- **Le natif ne rime pas forcément avec « coût » bas**

Le développement d'applications natives requiert des ressources, qu'elles soient financières ou humaines. Contrairement aux solutions hybrides ou aux web apps qui nécessitent en majorité des développeurs ayant des aptitudes dans des langages connus comme le HTML, le CSS ou encore JavaScript, le natif, lui, requiert plus d'expertise. Il faut en effet une équipe de développeurs spécialisés en développement iOS et une autre en Android lorsque vous voulez une application déployable sur les deux OS. Il en est de même pour sa maintenance.

- **Application native et « coût du Store »**

L'un des inconvénients des applications mobiles natives est la compatibilité qu'elles doivent avoir avec les différents stores. Une application téléchargée sur l'App Store ou la Play Store doit en effet être respectivement compatible avec un téléphone iOS ou Android. Elle doit donc respecter les règles d'approbation des mises à jour ou de diffusion sur ces stores respectifs. Il faut enfin intégrer dans les frais de développement de son application, les coûts d'abonnement sur la plateforme de publication spécifique.

I.3.2 Applications Web

Une application web, que l'on appelle aussi Web App, est une application hébergée en ligne sur un serveur et utilisable simplement à l'aide d'un navigateur. On peut donc utiliser l'application sans avoir à la télécharger sur son ordinateur ou quelconque appareil comme smartphone ou une tablette [8]. Le tableau 2 suivant donne trois exemples d'applications web ainsi que leurs caractéristiques techniques.

Application	Plateforme	Langage	100% web ?	Pourquoi ?
Moteur de recherche Google	Web	JavaScript, Python	✓ Oui	Accessible via navigateur sans installation, traitement serveur, interface web

Google Maps	Web	JavaScript, Python	✓ Oui	Fonctionne entièrement dans un navigateur avec API web et cartes interactives
Airbnb	Web	JavaScript, Ruby on Rails	✓ Oui	Utilisable à 100% depuis le web, sans installation d'application

Tableau 2: Exemples d'applications 100% Web

Les applications web présentent certains avantages et quelques inconvénients [8] :

Avantages :

- **L'accessibilité**

Une application web est plus rapidement accessible qu'une application native. L'utilisateur n'a pas besoin de télécharger une application sur son appareil, il peut l'utiliser instantanément. Lorsqu'il souhaite interagir avec l'application, il lui suffit d'une connexion internet et de rentrer l'URL de celle-ci dans un navigateur web, qu'il soit sur son ordinateur, sa tablette ou son smartphone.

- **La compatibilité**

L'application web est conçue de manière à être compatible sur tous les systèmes d'exploitation et responsives sur tous les appareils. Étant donné que celle-ci fonctionne grâce à un navigateur, elle ne dépend ni de l'appareil utilisé, ni du système d'exploitation et elle est faite pour s'adapter à tous les formats d'écrans.

- **La mobilité**

La mobilité est améliorée grâce aux applications web. Où que vous soyez, votre application est accessible tant que vous possédez une connexion internet. Vous pouvez passer d'un ordinateur à l'autre, de votre ordinateur à votre smartphone, etc... Vos données sont stockées dans l'application, ce qui vous permet de reprendre votre activité à tout moment et de n'importe où !

- **Le travail en équipe**

Dans le cadre d'une entreprise, les applications web simplifient le travail d'équipe. Elles permettent de créer des espaces collaboratifs sur lesquels il est très facile de partager le travail avec une équipe entière. Toutes les données sont stockées en ligne et il suffit de partager un lien ou de créer un accès pour accéder aux espaces de collaborations. Plus besoin de travailler en local et d'envoyer les documents par mails à chaque modification !

- **Pas de contrainte des magasins d'applications**

Le développement d'une application web permet d'éviter certaines contraintes imposées par les magasins d'applications natives. Votre application étant hébergée en ligne, elle n'est pas soumise aux différentes règles de conformité ou à la vérification des magasins d'application comme Google Play Store ou l'App Store. Cela permet de gagner beaucoup de temps sur le développement.

- **Mise à jour et sécurité**

L'application web est toujours à jour. Contrairement aux applications natives qu'il faut mettre à jour manuellement, les applications web ont l'avantage d'être toujours à jour pour tout le monde. La mise à jour se fait côté serveur et non du côté de l'utilisateur, aucune action n'est donc requise par l'utilisateur. Cela permet de renforcer la sécurité. Lorsqu'une faille est détectée, elle est corrigée pour tous les utilisateurs en même temps.

- **Stockage**

Plus de soucis à se faire concernant la limite de stockage de votre ordinateur ou de votre smartphone. Les applications web sont hébergées en ligne et vous n'avez pas de données à stocker directement sur votre appareil. Les applications peuvent aussi permettre d'enregistrer votre travail, vos documents, ou n'importe quels fichiers directement sur le cloud qui va proposer une capacité de stockage quasiment infinie.

Inconvénients :

- **Dépendance à internet**

Une application web ne fonctionne que si vous avez accès à internet. Sans connexion, vous ne pourrez pas vous rendre sur votre navigateur et interagir avec la Web app. Aujourd'hui, rares sont les situations où aucune connexion internet n'est disponible mais dans ce cas précis, l'application ne vous sera d'aucune utilité.

- **La compatibilité**

Un des avantages de l'application web est sa compatibilité sous différents formats. Mais pour cela, il faut développer une application parfaitement responsive et compatible au niveau technique. C'est une tâche qui peut parfois compliquer le développement et demander beaucoup de temps.

- **Moins de visibilité**

Une application web ne bénéficie pas de la visibilité des magasins d'applications. Celle-ci a la même visibilité qu'un site web classique sur lequel il faut travailler le référencement.

- **Accès restreint à certaines fonctionnalités**

Une application web a un accès restreint aux fonctionnalités de l'appareil contrairement aux applications natives qui ont un accès simplifié aux fonctionnalités matérielles comme le GPS, la caméra, le micro, etc.

I.3.3 Applications Hybrides

Une application mobile hybride est une application conçue pour être utilisée sur plusieurs plates-formes. Une application hybride est un mélange d'une application native et d'une application web [9].

Ces applications sont créées avec des technologies web telles que HTML, CSS et JavaScript, mais peuvent être déployées sur plusieurs systèmes d'exploitation mobiles (iOS, Android) en utilisant des wrappers ¹ natifs. Le tableau 3 suivant donne trois exemples d'applications hybrides ainsi que leurs caractéristiques techniques.

Application	Plateforme	Langage	100% Hybride ?	Pourquoi ?
Gmail (mobile)	iOS, Android	HTML, JavaScript (WebView)	✓ Oui	Utilise WebView pour afficher l'interface, entièrement basée sur du code web
Evernote	iOS, Android	HTML, JavaScript (Cordova)	✓ Oui	À l'origine conçue comme une app 100% hybride avec Cordova pour mutualiser le code
Untappd	iOS, Android	Ionic Framework	✓ Oui	App sociale bien connue dans le domaine de la bière, faite 100% en Ionic

Tableau 3: Exemples d'applications mobiles 100% hybrides

Les applications hybrides présentent certains avantages et quelques inconvénients [10] :

Avantages :

- **Allier l'expérience utilisateur avec l'agilité du cycle de développement et des coûts maîtrisés.**

¹ Un **wrapper** est un composant logiciel qui encapsule une autre fonctionnalité ou un autre module pour **simplifier son utilisation** ou **le rendre compatible avec un environnement différent**. Il agit comme une **interface intermédiaire** entre l'utilisateur (ou un autre programme) et une bibliothèque ou un système plus complexe.

- **Contourner les limitations de l'App store Apple :** pour déployer une application sur l'App store Apple, il faut soumettre l'application et attendre sa validation. Le temps de validation variera en fonction du moment de l'année, mais aussi de s'il s'agit d'une première version de l'application ou bien d'une mise à jour. Il est généralement compris entre 1 à 7 jours. Les applications hybrides présentent donc un grand avantage pour les développeurs souhaitant pouvoir mettre à jour leur application fréquemment puisqu'il n'est pas nécessaire de resoumettre la nouvelle version si les éléments modifiés ne touchent pas au code natif.
- **Trouver des talents :** la plupart des applications ont une version iOS et une version Android. Elles sont donc développées en utilisant le langage de programmation correspondant : Objective-C ou Swift pour iOS, Java pour Android. Les applications hybrides permettent quant à elles l'utilisation de langages utilisés de façon fréquente par les développeurs web (HTML, JavaScript et CSS) qui peuvent donc utiliser leurs connaissances. Il est ainsi plus facile de trouver des talents pour développer une application hybride.
- **Réutiliser le code de la partie web app :** le code est écrit une fois et déployé ensuite sur toutes les plateformes mobiles.
- **Réduire les temps de développement et les coûts :** le code étant écrit une fois, cela réduit considérablement les temps de développement et les coûts par rapport aux applications natives nécessitant d'une part un développement pour iOS et d'autre part un développement pour Android.

Inconvénients :

- **Une interface utilisateur limitée :** les applications hybrides ont en effet un design qui n'a pas le ressenti d'un design natif. L'interface utilisateur n'est donc pas aussi fluide. Les possibilités (3D par exemple) sont aussi limitées du fait de l'utilisation du WebView ne permet pas d'exploiter tout le potentiel de l'appareil.
- **La difficulté ou l'impossibilité d'exploiter toutes les capacités des plateformes :** chaque plateforme, iOS ou Android, a des capacités uniques qu'un développeur peut souhaiter exploiter. Si tel est le cas, c'est en utilisant une combinaison de plugins et de code spécifique à la plateforme que cela pourra être réalisé pour une application hybride. Cela complexifie d'autant le projet de développement de l'application.
- **Des performances et des transitions entre les pages plus lentes.**
- **Une dépendance vis à vis de la vitesse du navigateur.**

I.3.4. Les applications multiplateformes

Une application multiplateforme, également connue sous le nom d'application cross-plateforme, est une application logicielle qui peut être utilisée sur différentes plateformes et systèmes d'exploitation, tels que iOS, Android, Windows, etc.

Une application multiplateforme peut être développée une seule fois pour être exécutée sur différents types de dispositifs sans nécessiter de modifications importantes ou de réécriture pour chaque plateforme. Les applications multiplateformes sont souvent développées à l'aide de frameworks et de langages de programmation tels que React Native, Flutter, Xamarin, ou encore Cordova/PhoneGap, qui permettent de partager le code entre différentes plateformes, et de réduire les coûts et les temps de développement [11]. Le tableau 4 suivant donne trois exemples d'applications multiplateformes ainsi que leurs caractéristiques techniques.

Application	Plateforme	Langage	100% multiplateformes ?	Pourquoi ?
WhatsApp	iOS, Android, Web	React Native, WebRTC	✓ Oui	Une seule base de code pour toutes les plateformes
Telegram	iOS, Android, Web	React Native, JavaScript	✓ Oui	Fonctionne sur toutes les plateformes avec un seul code
Figma	iOS, Android, Web	Web technologies (HTML, CSS, JS)	✓ Oui	Code partagé pour mobile et web

Tableau 4: Exemples d'applications 100% multiplateformes

Les **applications multiplateformes** présentent certains avantages et quelques inconvénients [12] :

Avantages :

- **Réduction des coûts de développement :**

Une seule base de code réduit les coûts en développement et maintenance.

- **Déploiement plus rapide :**

Le temps de développement est réduit, facilitant le lancement simultané sur plusieurs plateformes.

- **Cohérence de l'expérience utilisateur :**

Une interface uniforme renforce la fidélité des utilisateurs sur toutes les plateformes.

- **Accès à un plus grand marché :**

Cibler simultanément Android et iOS maximise la portée et le retour sur investissement.

Inconvénients :

- **Performance parfois inférieure :**

Les applications multiplateformes peuvent être moins performantes que les applications natives.

- **Accès limité aux fonctionnalités spécifiques :**

Certaines intégrations natives peuvent nécessiter du code spécifique à chaque plateforme.

- **Problèmes de compatibilité :**

Les mises à jour fréquentes des systèmes peuvent entraîner des bugs ou des incompatibilités.

- **Expérience utilisateur moins optimisée :**

L'interface peut ne pas tirer pleinement parti des spécificités de chaque système.

I.4. L'Architecture d'une Application

L'architecture d'une application permet de mieux comprendre son fonctionnement et de comment ses différentes parties sont organisées. Chaque application doit comporter au moins deux couches [13] :

- La couche d'interface utilisateur qui affiche les données de l'application à l'écran.
- La couche de données qui contient la logique métier de votre application et expose les données de l'application.

Il est possible d'ajouter une couche, appelée couche de domaine, pour simplifier et réutiliser les interactions entre l'interface utilisateur et les couches de données. La figure 1, ci-dessous montre l'approche minimale pour structurer une application.

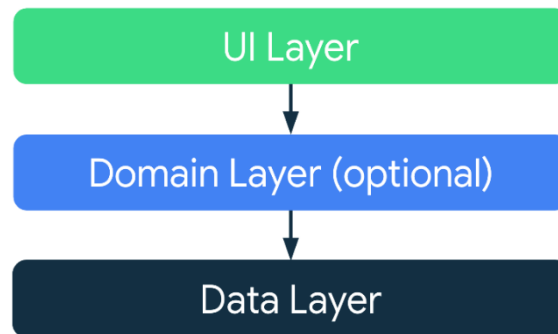


Figure 1: Schéma d'une architecture d'application classique

Couches :

- **UI Layer** : gère l'affichage et les interactions utilisateur (écrans, composants, animations).
- **Domain layer (optionnel)** : contient la logique métier pure (règles de la gestion, cas d'utilisation).
- **Data layer** : gère les données (bases locales, API, repositories)

I.4.1. Couche d'interface utilisateur

Le rôle de la couche de l'interface utilisateur (ou *couche de présentation*) consiste à afficher les données de l'application à l'écran. Chaque fois que les données changent, soit à la suite d'une interaction de l'utilisateur (par exemple, si celui-ci appuie sur un bouton) ou d'une entrée externe (telle qu'une réponse du réseau), l'UI doit être mise à jour pour refléter les modifications.

La couche d'UI contient les deux éléments suivants :

- Éléments d'interface utilisateur qui affichent les données à l'écran. Ces éléments sont créés à l'aide des fonctions View ou Jetpack Compose.
- Les conteneurs d'état (tels que les classes ViewModel) qui contiennent des données les exposent à l'interface utilisateur et gèrent la logique.

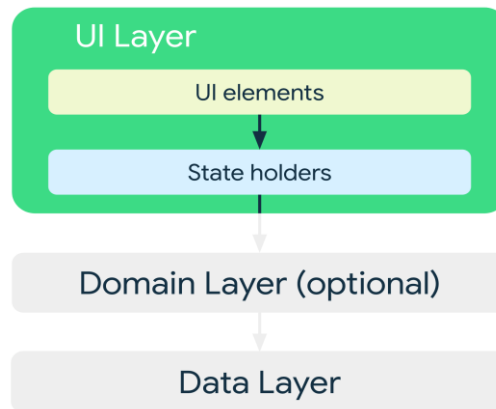


Figure 2: Rôle de la couche d'interface utilisateur dans l'architecture de l'application

Cette figure 2 met en évidence le rôle central des State Holders (tels que HomeViewModel) dans la médiation des échanges entre l'interface utilisateur (UI) et les données de l'application. Les State Holders s'appuient sur des Use Cases pour exécuter la logique métier, et sur des repositories pour accéder aux sources de données de manière abstraite.

La couche UI (User Interface) se compose principalement de deux éléments :

- **Les éléments de l'interface utilisateur** : il s'agit des composants visuels (boutons, listes, champs, etc.) avec lesquels l'utilisateur interagit directement.
- **Les State Holders (par exemple, les ViewModels dans le modèle MVVM)** : ils assurent la gestion de l'état de l'interface et servent d'intermédiaires entre les composants visuels et les couches Domain et Data.

I.4.2. Couche de données

La couche de données d'une application contient la *logique métier*. La logique métier est ce qui donne de la valeur à votre application. Elle repose sur des règles qui déterminent la manière dont votre application crée, stocke et modifie les données.

La couche de données est constituée de *dépôts* pouvant contenir de zéro à plusieurs *sources de données*. Vous devez créer une classe de dépôt pour chaque type de données que vous gérez dans votre application. Par exemple, vous pouvez créer une classe `MoviesRepository` pour les données liées aux films, ou une classe `PaymentsRepository` pour les données liées aux paiements.

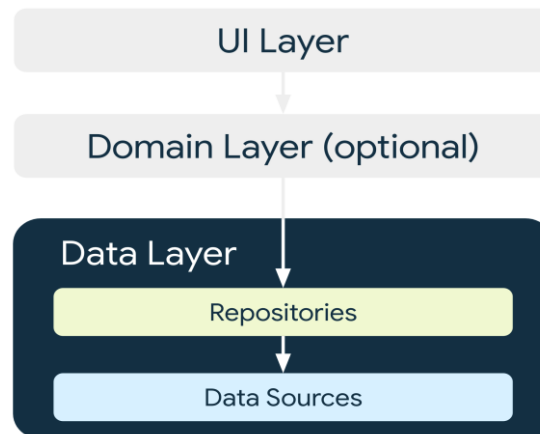


Figure 3: Rôle de la couche de données dans l'architecture de l'application.

La Figure 3 illustre la structure approfondie de la **Data Layer**, avec ses sous-composants.

- **Repositories** : Centralisent l'accès aux données (ex : BooksRepository). Ils masquent la provenance des données (locale/distante).
- **Data Sources** : Implémentations concrètes (ex : BooksLocalDataSource pour SQLite, BooksRemoteDataSource pour API).

Les classes de dépôt sont responsables des tâches suivantes :

- Présenter les données au reste de l'application
- Centraliser les modifications apportées aux données
- Résoudre les conflits entre plusieurs sources de données
- Extraire des sources de données du reste de l'application
- Contenir la logique métier

Chaque classe de source de données doit avoir la responsabilité de travailler avec une seule source de données, à savoir un fichier, une source réseau ou une base de données locale. Les classes de sources de données font le lien entre l'application et le système pour les opérations de données.

I.4.3. Couche de domaine

La couche de domaine est une couche facultative qui se trouve entre l'UI et les couches de données.

La couche de domaine est chargée d'encapsuler une logique métier complexe, ou une logique métier simple qui est réutilisée par plusieurs ViewModels. Cette couche est facultative, car ces exigences ne s'appliquent pas à toutes les applications. Vous ne devez l'utiliser que lorsque cela est nécessaire, par exemple pour gérer la complexité ou favoriser la réutilisation.

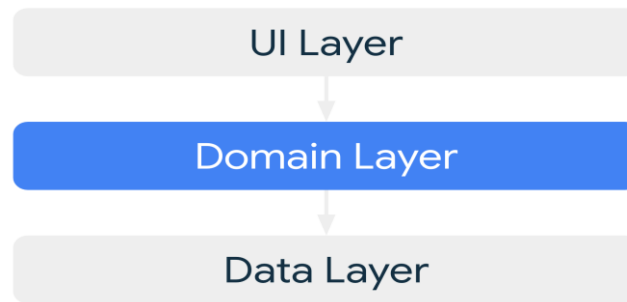


Figure 4: Rôle de la couche de domaine dans l'architecture de l'application.

La Figure 4 illustre comment la Domain Layer sert de médiateur entre l'interface utilisateur et les données. En centralisant les règles métiers (ex : validation des montants d'échange), elle garantit une logique cohérente, testable et indépendante des choix techniques UI/Data.

Les classes de cette couche sont communément appelées use cases (cas d'utilisation) ou interactors (interacteurs). Chaque cas d'utilisation doit être responsable d'une seule fonctionnalité. Par exemple, votre application peut avoir une classe `GetTimeZoneUseCase` si plusieurs ViewModels utilisent les fuseaux horaires pour afficher le bon message à l'écran.

I.5. Organisation Modulaire d'une Application Mobile

Un module dans une application mobile est un composant spécialisé qui regroupe un ensemble de fonctionnalités dédiées à un aspect spécifique du fonctionnement de l'application. La façon dont vous organisez vos modules dépend principalement de l'architecture de votre application, généralement, il existe 5 modules [14] :

I.5.1. Modules de données

Un module de données contient généralement un dépôt, des sources de données et des classes de modèle. Les trois principales responsabilités d'un module de données sont les suivantes :

1. Encapsuler toutes les données et la logique métier d'un domaine donné.
2. Exposer le dépôt en tant qu'API externe.
3. Masquer toutes les informations concernant l'implémentation et les sources de données pour l'extérieur.

Exemple : un module qui gère les données utilisateur (profil, préférences, historique) dans une application de messagerie

I.5.2. Modules de fonctionnalités

Une fonctionnalité est une partie isolée du fonctionnement d'une application correspondant généralement à un écran ou à une série d'écrans étroitement liés, comme un processus d'inscription ou de règlement. Si votre application dispose d'une barre de navigation inférieure, il est probable que chaque destination soit une fonctionnalité.

Exemple : un module pour la gestion du panier dans une application e-commerce.

I.5.3. Modules d'applications

Les modules d'application constituent un point d'entrée de l'application. Ils dépendent de modules de fonctionnalités et fournissent généralement une navigation racine. Un module d'application unique peut être compilé pour plusieurs binaires grâce à des variantes de compilation.

Exemple : un module principal pour Android et un module allégé pour les montres connectées ou Android TV.

I.5.4. Modules courants

Les modules courants, également appelés modules principaux, contiennent du code que d'autres modules utilisent fréquemment. Ils réduisent la redondance et ne représentent aucune couche spécifique dans l'architecture d'une application.

Voici des exemples de modules courants :

- Module d'interface utilisateur.
- Module d'analyse.
- Module réseau.
- Module utilitaire.

I.5.5. Modules de test

Les modules de test sont des modules Android utilisés uniquement à des fins de test. Les modules contiennent du code, des ressources et des dépendances de test qui ne sont nécessaires que pour l'exécution de tests et qui ne sont pas requis pendant l'exécution de l'application. Les modules de test sont créés pour séparer le code spécifique au test de l'application principale, ce qui facilite la gestion et la maintenance du code du module.

Exemple : un module de test dans une application de messagerie vérifie l'envoi et la réception des messages via des tests unitaires et d'intégration.

I.6. Conclusion

Dans ce premier chapitre, nous avons abordé en détail le domaine des applications mobiles en commençant par leur définition, leurs fonctionnalités et leurs principales caractéristiques. Nous avons étudié les avantages et inconvénients des applications, tout en mettant en évidence les différents types : natives, web, hybrides et multiplateformes, chacun ayant ses spécificités et cas d'utilisation.

L'architecture d'une application mobile repose sur la séparation des responsabilités en couches distinctes (interface utilisateur, logique métier, données), favorisant ainsi la maintenabilité et la testabilité. La modularisation divise l'application en modules indépendants, facilitant la réutilisation du code et la collaboration entre équipes.

Ce chapitre a donc posé les bases essentielles pour comprendre le fonctionnement et les exigences des applications mobiles modernes. Le prochain chapitre se concentrera sur l'analyse et la conception de notre système « *Echange +* », qui est une application de type **Multiplateforme, détaillant**, ainsi, le processus de développement et les spécifications nécessaires à sa mise en œuvre.

Chapitre II : Analyse et conception de l'application « Echange+ »

II.1. Introduction

La conception est une étape importante dans le développement d'un système, permettant de définir son architecture et ses principales fonctionnalités. Ce chapitre est consacré à l'étude conceptuelle de notre projet afin de garantir une modélisation rigoureuse et adaptée aux besoins identifiés. Dans un premier temps, nous réalisons une comparaison avec les applications existantes afin d'analyser leurs fonctionnalités, leurs limites et leurs points forts. Cette analyse nous permet d'identifier les améliorations et les innovations à apporter à notre solution. Ensuite, nous définissons l'architecture du système à travers des modèles UML, notamment les diagrammes de cas d'utilisation et de séquence, afin de structurer et clarifier les interactions entre les différentes entités du système. Cette approche vise à assurer une conception optimisée et cohérente en vue de la phase de développement.

II.2. Etude de l'existant

Dans cette section, nous allons explorer et analyser les applications existantes d'échange de services entre utilisateurs, afin de mieux comprendre les fonctionnalités proposées par ces plateformes. Cette étude vise à identifier les forces et les faiblesses des solutions actuelles, à comparer leurs fonctionnalités aux besoins spécifiques de notre projet. A travers cette analyse, nous serons en mesure de définir les exigences fonctionnelles et non fonctionnelles de notre propre application, afin de proposer une solution innovante, efficace et adaptée aux attentes des utilisateurs.

II.2.1. Simbi

Simbi est une plateforme mondiale de services en ligne lancée en 2016. Il permet à ses utilisateurs d'offrir et de recevoir certains services (livres de cuisine, conseils, design, etc.) sans utiliser d'argent réel. Simbi utilise une monnaie virtuelle (Simbi) que les membres utilisent pour échanger leurs compétences en gagnant et en dépensant des crédits. Il permet également d'échanger directement des services. Accessible sur son site web, la plateforme s'adresse principalement aux consommateurs américains, mais est ouverte à tous les consommateurs du monde entier [15]. **La figure 5** représente la plateforme de Simbi.

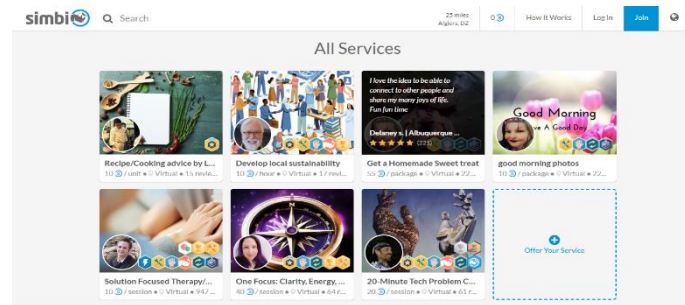


Figure 5: plateforme Simbi

II.2.2. Fiverr

Cette plateforme en ligne de services digitaux freelance, créée en 2010, fonctionne à travers une application mobile et un site web. Elle permet aux utilisateurs de proposer et d'acheter divers services numériques dans plusieurs domaines, tels que le graphisme, la rédaction, la programmation, le marketing, et bien d'autres. Présente sur le marché international, **Fiverr** est utilisée dans plus de 160 pays [16]. En 2020, la plateforme comptait environ 3.4 millions d'acheteurs actifs [17].

Parmi ces fonctionnalités principales, on peut citer la messagerie intégrée qui facilite la communication entre les prestataires de services (freelances) et les clients, ainsi que des systèmes sécurisés de paiement, notamment par cartes bancaires et PayPal. La figure 6 représente l'application de Fiverr[18].

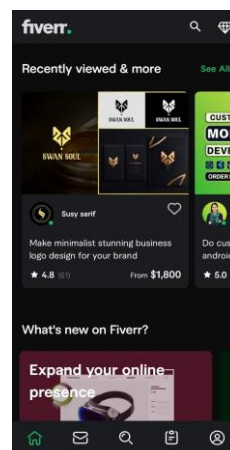


Figure 6: Application Fiverr

II.2.3. Youchoz

Youchoz est une plateforme web de mise en relation entre particuliers à la recherche de professionnels du bâtiment et des experts du BTP ayant besoin de nouvelles opportunités professionnelles. L'objectif de la plateforme est de faciliter la recherche d'artisans, d'obtenir un devis détaillé en ligne rapidement, de consulter les photos des chantiers précédents du professionnel ainsi que les avis qui auront été laissés par d'autres clients. Ce projet, lancé par trois étudiants en génie civil [19]. Parmi ces fonctionnalités principales, on peut citer la messagerie intégrée qui facilite la communication entre les utilisateurs, ainsi qu'un système de géolocalisation pour trouver facilement des prestataires proches. Le paiement des services ne passe pas par la plateforme et se fait directement entre les utilisateurs. La figure 7 représente l'application de Youchoz [20].



Figure 7: Application Youchoz

II.2.4. Youchoz Pro

Youchoz Pro est la version professionnelle de **Youchoz**, spécialement conçue pour les prestataires de services professionnels tels que les entreprises, les artisans et les travailleurs indépendants. Elle propose des fonctionnalités adaptées aux besoins des professionnels, comme la création de profils détaillés, l'affichage des prix, des outils de gestion pour organiser les demandes et les disponibilités, ainsi que la possibilité de générer des devis et des factures personnalisées. Cependant, Youchoz Pro ne dispose pas de système de paiement en ligne, les paiements se font directement entre les utilisateurs en dehors de la plateforme. La figure 8 représente l'application de Youchoz Pro [21].

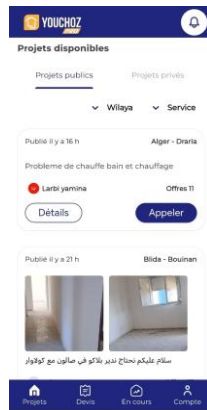


Figure 8: Application Youchoz Pro

II.2.5. Comparaison entre les différentes applications

Afin de comparer les différentes plateformes d'échange de services existantes, nous avons retenu un ensemble de critères fonctionnels et techniques essentiels, permettant d'évaluer leur efficacité, leur couverture fonctionnelle et leur adaptation aux besoins des utilisateurs. Ces critères sont les suivants :

- Présence d'une application mobile.
- Existence d'un site web.
- Possibilité d'échanger un service contre un autre.
- Intégration d'un paiement en ligne sécurisé.
- Présence d'un système de points (avec ou sans expiration).
- Géolocalisation des prestataires ou des annonces.
- Messagerie intégrée pour faciliter la communication.
- Système d'évaluation des utilisateurs.
- Signalement des utilisateurs ou des problèmes rencontrés.
- Existence d'une interface d'administration pour la gestion, la modération et le suivi.

Ces critères nous ont permis de réaliser une comparaison structurée et équitable, mettant en lumière les forces et faiblesses de chaque plateforme.

Le tableau 5 ci-après est un tableau comparatif entre les applications cités ci-dessus et notre application nommé **Echange+** :

Application	Fiverr	Simbi	Youchoz	Youchoz Pro	Echange+
Date de sortie	2010	2016	2021	2021	2025
Origine	internationale	internationale	algérienne	algérienne	algérienne
Application mobile	✓	✗	✓	✓	✓
Site web	✓	✓	✓	✗	✗
Echange par service	✗	✓	✗	✗	✓
paiement en ligne	✓	✗	✗	✗	✓
Système de points	✗	✓	✗	✗	✓
Géolocalisation	✓	✗	✓	✓	✓
Messagerie intégrée	✓	✓	✗	✓	✗
Système d'évaluation	✓	✓	✓	✓	✓
Signalement des utilisateurs	✓	✗	✗	✓	✓
Signalement des problèmes	✓	✗	✗	✗	✓
Expiration des points	✗	✗	✗	✗	✓
Interface admin	?	?	?	?	✓

Tableau 5:Tableau comparatif entre les applications de services

II.2.6. Discussion

L'analyse comparative a mis en évidence plusieurs limites fonctionnelles dans les plateformes existantes, qui peuvent réduire la fluidité, la sécurité ou l'accessibilité pour les utilisateurs.

Fiverr est très développé à l'échelle internationale, mais ne propose pas d'échange sans argent, ce qui exclut les utilisateurs ayant peu de moyens. Il n'intègre pas de système de points, et son modèle reste purement commercial.

Simbi, malgré son orientation vers l'échange sans argent, n'offre pas d'application mobile, ni de géolocalisation, ni de système de signalement. Cela limite son usage dans un contexte local et communautaire.

Youchoz se concentre principalement sur les métiers du bâtiment. Il ne permet pas l'échange de services, ne propose ni paiement intégré, ni système de points, et ne dispose pas d'un système de signalement ou de modération, ce qui limite la sécurité et la liberté des utilisateurs.

Youchoz Pro apporte des améliorations en intégrant la messagerie et le signalement des utilisateurs. Cependant, il ne propose pas de paiement en ligne, ni de système de points, ce qui laisse un doute sur sa capacité à gérer efficacement la plateforme à grande échelle.

Face à ces constats, **Echange+** se positionne comme une solution algérienne innovante complète, flexible et inclusive, pensée pour combler ces lacunes. Elle intègre :

- Trois modes d'échange : service contre service, service contre points, et paiement via Edahabia.
- Un système de points avec expiration pour encourager les échanges actifs.
- Des fonctionnalités clés : géolocalisation, notifications, évaluations, messagerie, signalement des utilisateurs et problèmes.
- Une interface d'administration dédiée, pour gérer les utilisateurs, les annonces, les échanges, et assurer un suivi en temps réel.

Echange+ réunit ainsi toutes les fonctionnalités dispersées ou absentes dans les autres plateformes, pour offrir une expérience sécurisée, adaptée au contexte local et accessible même sans argent.

II.3 Spécification des besoins

Dans cette phase, notre objectif est de comprendre en profondeur le contexte de l'application. Cela consiste à identifier les fonctionnalités essentielles et les acteurs principaux, tout en analysant les risques potentiels. Il s'agit aussi d'une étape où le client définit ses besoins.

II.3.1 Les acteurs de l'application

Notre application implique deux acteurs clés :

- **Utilisateur**

L'utilisateur est l'acteur principal du système, qu'il soit client ou prestataire de services. Il utilise l'application pour rechercher des services, consulter ses échanges et suivre son solde de points et argent. Grâce aux fonctionnalités de l'application, il peut faire l'échange via trois modes : par service, points accumulés ou par paiement en argent. Après chaque échange, il a la possibilité d'évaluer les prestataires de services afin de garantir la fiabilité et la qualité des

échanges. Son expérience est optimisée par une **interface ergonomique** et un processus sécurisé, garantissant des échanges rapides et fiables.

- **Administrateur**

L'administrateur est responsable de la gestion. C'est lui qui assure le bon déroulement des activités sur la plateforme au quotidien. Son travail consiste principalement à faire respecter les règles établies (termes et conditions), à superviser les échanges et traiter les signalements des utilisateurs et à intervenir en cas de problèmes afin de garantir un environnement sécurisé et équitable pour tous les utilisateurs. Il dispose également d'un accès aux données statistiques et revenus de la plateforme.

II.3.2 Les besoins

Dans cette partie, on va identifier les besoins essentiels auxquels l'application doit répondre. Ces besoins reflètent ce que l'utilisateur et l'administrateur attendent de l'application pour bien l'utiliser. Ils servent de base pour définir les fonctionnalités à développer.

II.3.2.1 Besoins liés à l'utilisateur

- L'utilisateur peut effectuer des échanges selon plusieurs modes (services, points, argent) et suivre leur état en temps réel.
- L'utilisateur peut consulter l'historique de ses échanges et recevoir des notifications relatives à ces derniers.
- L'utilisateur peut ajouter une carte Edahabia pour effectuer des paiements en argent et recevoir un bonus de 500.00 DA.
- L'utilisateur peut être informé de l'expiration de ses points et acheter des points via la carte Edahabia.
- L'utilisateur peut gérer ses annonces (publier, modifier, supprimer) et rechercher, filtrer et consulter la localisation des annonces et utilisateurs selon différents critères.
- L'utilisateur peut contacter un autre utilisateur uniquement après un échange accepté et évaluer cet échange.
- L'utilisateur peut signaler des problèmes ou des utilisateurs si nécessaires.
- L'utilisateur peut partager des annonces et profils sur les réseaux sociaux.

II.3.2.2 Besoins liés à l'administrateur

- L'administrateur peut surveiller et gérer le contenu de la plateforme, incluant la consultation des annonces en détail (pour vérifier la conformité), la gestion de catégories, et la suppression des avis utilisateurs pour assurer le respect des conditions.
- L'administrateur peut superviser et gérer les échanges et les revenus, en ayant une vue globale des modes (service, point, argent), en visualisant les statistiques sur l'activité globale et les revenus générés, et en appliquant automatiquement une commission sur les échanges (points et argent). Il peut également ajuster manuellement le solde de points d'un utilisateur pour corriger des erreurs.
- L'administrateur peut gérer les utilisateurs et les signalements, ce qui inclut le traitement des signalements (et la mise à jour de leur statut), la consultation de la liste des utilisateurs signalés et des signalements associés, la réactivation d'un compte suspendu après un délai défini (15 jours), et la possibilité de contacter un utilisateur via e-mail, téléphone ou SMS pour enquêter ou résoudre un litige.
- L'administrateur peut accéder à un tableau de bord complet pour surveiller l'ensemble des données de la plateforme (utilisateurs, échanges, revenus, annonces, commentaires et signalements), et effectuer des recherches et des filtres sur les utilisateurs, les annonces, les signalements et les échanges pour une gestion efficace des données.

II.3.3 Les exigences fonctionnelles

Dans cette partie, nous allons présenter les exigences fonctionnelles de notre application en considérant les différents intervenants de l'application. Il doit répondre au besoins de l'utilisateur et de l'administrateur.

L'application doit offrir les fonctionnalités suivantes :

Pour l'utilisateur :

Pour accéder aux différentes fonctionnalités de l'application, l'utilisateur doit s'authentifier en utilisant un courriel et un mot de passe, ou via un compte Gmail ou Facebook.

Gestion des annonces : il s'agit d'un journal interactif, il doit être affiché sur la page d'accueil, contenant les annonces des services des utilisateurs.

- L'application doit permettre à l'utilisateur de publier des annonces (en renseignant titre, catégorie, description, mode d'échange et localisation), de les consulter, de les modifier ou de les supprimer à tout moment.

Échange de services et moyen de paiement : consiste à offrir différents modes d'échange tout en assurant un suivi précis des échanges.

- L'application doit permettre à l'utilisateur de proposer des échanges (par services, points, ou argent via un moyen électronique), d'afficher son solde de points et d'argent en temps réel.
- L'application doit également inclure un suivi en temps réel des échanges pour consulter et modifier leur état, et confirmer la réception du service.
- L'application doit permettre l'ajout d'une carte Edahabia pour les paiements en argent, offrant un bonus de 500,00 DA.

Recherche et filtrage :

- L'application doit intégrer une barre de recherche et de filtres pour les annonces (par catégorie, localisation, mode d'échange) et permettre de retrouver un autre membre par son nom et prénom pour accéder à son profil, annonces et évaluations.
- L'application doit permettre de filtrer les propositions de services et les demandes des clients par leur statut (en attente, acceptée, terminée) et par date.

Évaluation et avis : permet à l'utilisateur d'évaluer les prestataires et d'assurer la transparence des services.

- L'application doit mettre en place un système d'évaluation permettant de noter les prestataires (étoiles et commentaires) après un échange, de calculer et d'afficher la moyenne des évaluations sur le profil du prestataire, et permettre la modification ou la suppression des avis.

Géolocalisation et accessibilité : permet à l'utilisateur de trouver plus facilement des services en fonction de leur emplacement.

- L'application doit pouvoir détecter la localisation de l'utilisateur pour proposer des services adaptés à sa position géographique et afficher la localisation géographique du service dans chaque annonce.

Notifications :

- L'application doit envoyer des notifications à l'utilisateur lorsqu'il reçoit une proposition d'échange ou lorsque sa demande est acceptée ou refusée.

Sécurité et modération : garantit un environnement sûr en offrant des outils de signalement et de modération.

- L'application doit gérer un système de points avec une période de validité de 6 mois après le dernier échange valide, afficher une alerte sur le temps restant avant l'expiration des points, et proposer l'achat de points via un moyen de paiement électronique si le solde est insuffisant.

Moyen de contact entre utilisateurs :

- L'application doit permettre à un utilisateur de contacter un autre utilisateur par numéro de téléphone, SMS ou Gmail, mais uniquement après un échange accepté, afin de garantir un usage sécurisé.

Pour l'administrateur :

Pour accéder aux différentes fonctionnalités de l'application, l'administrateur doit s'authentifier en utilisant un courriel et un mot de passe.

Gestion des annonces, des catégories et des avis :

- L'application doit permettre à l'administrateur de consulter les détails complets de chaque annonce, de vérifier leur conformité aux règles et de supprimer celles ne respectant pas les conditions d'utilisation.
- L'application doit permettre à l'administrateur de créer, modifier et supprimer des catégories de services.
- L'application doit permettre à l'administrateur de vérifier la conformité des commentaires des utilisateurs et de supprimer ceux ne respectant pas les règles de la plateforme.

Supervision des échanges de services : permet à l'administrateur de gérer et de suivre l'ensemble des échanges réalisés sur la plateforme.

- L'application doit permettre à l'administrateur de superviser les échanges selon les modes d'échange disponibles (services, points et argent), en affichant pour chaque opération : l'état, le client, le prestataire, la date et le service concerné.

Sécurité et modération : permet à l'administrateur de garantir un environnement sécurisé en contrôlant les signalements, en gérant les utilisateurs et en assurant le respect des règles de la plateforme.

- L'application doit permettre à l'administrateur d'accéder à la liste et à l'historique des signalements, de les traiter et de mettre à jour leur statut, afin de prendre les mesures appropriées.

- L'application doit permettre à l'administrateur de réactiver les comptes suspendus après une durée spécifique (15 jours).

Gestion et surveillance : fournit à l'administrateur des outils pour superviser et gérer l'ensemble des activités sur la plateforme.

- L'application doit fournir à l'administrateur un tableau de bord complet pour consulter la liste des utilisateurs (inscrits, signalés), gérer les catégories, les signalements, superviser les échanges, les revenus, et vérifier les annonces et les commentaires.
- L'application doit permettre à l'administrateur d'ajuster manuellement le solde de points d'un utilisateur pour corriger des erreurs techniques.
- L'application doit générer et afficher des statistiques automatiques sur l'activité globale de la plateforme (nombre d'échanges, d'annonces, d'utilisateurs signalés, actifs, suspendus).

Gestion des revenus et des bénéfices : consiste à permettre à l'administrateur de suivre et gérer les revenus, les commissions et la répartition des points dans l'application.

- L'application doit permettre à l'administrateur de suivre les revenus générés par les échanges (points, argent, achat de points) pour évaluer la rentabilité.
- L'application doit permettre à l'administrateur de percevoir une commission automatique de 20 % sur les échanges par points et de 25 % sur les échanges par argent.

Recherche et filtrage :

- L'application doit permettre à l'administrateur de rechercher des utilisateurs (par nom, prénom, e-mail) et de filtrer les annonces (par statut : validées, supprimées), les signalements (par statut, gravité), et les échanges (par statut et date pour les trois modes).

Moyen de contact :

- L'application doit permettre à l'administrateur de contacter les utilisateurs via leur adresse Gmail, leur numéro de téléphone ou SMS.

II.3.4 Les exigences non fonctionnelles :

Les exigences non fonctionnelles de notre application se concentrent sur la performance, la sécurité, la disponibilité, la fiabilité, l'utilisabilité, l'extensibilité, la maintenabilité, l'interopérabilité, l'évolutivité et la compatibilité afin d'assurer une expérience utilisateur optimale. En particulier, notre application doit satisfaire les exigences suivantes :

- **Performance et évolutivité :**

L'application doit pouvoir supporter un grand nombre d'utilisateurs actifs simultanément et gérer une augmentation progressive du nombre d'utilisateurs et d'annonces sans dégradation des performances. Elle doit garantir des temps de réponse optimaux pour les interactions clés comme la consultation des annonces et la gestion des échanges de services.

- **Sécurité et fiabilité :**

Des mesures de protection doivent être mises en place pour assurer la confidentialité des données, limiter l'accès aux informations sensibles, et sécuriser les échanges. L'application doit également garantir sa stabilité en réduisant les risques de défaillance et en assurant la continuité des services essentiels (publication d'annonces, paiements).

- **Disponibilité et utilisabilité :**

L'application doit être accessible en continu (24h/24, 7j/7). Elle doit offrir une interface fluide et ergonomique qui facilite la navigation et l'interaction, permettant aux utilisateurs de rechercher des services, proposer des offres, et gérer leurs transactions sans difficulté.

- **Extensibilité et maintenabilité :**

L'application doit adopter une architecture modulaire qui permet l'ajout progressif de nouvelles fonctionnalités. Son code source doit être clairement structuré et documenté pour simplifier la maintenance, la correction des erreurs, et les mises à jour.

- **Interopérabilité et compatibilité :**

L'application doit pouvoir se connecter à des services tiers comme **Firestore** (gestion utilisateurs, authentification via Google/Facebook, stockage annonces), intégrer la carte **Edahabia** pour les paiements, et utiliser la géolocalisation. Elle doit être adaptée aux plateformes Android et iOS.

II.4 Modélisation des besoins fonctionnels

Cette section est dédiée à la modélisation des besoins fonctionnels d'Échange+, une étape fondamentale dans le processus de conception de l'application. Nous avons choisi le langage UML, qui permet de représenter de manière formelle et structurée les fonctionnalités de l'application. L'objectif est d'identifier et d'organiser rigoureusement les fonctionnalités attendues, en tenant compte des exigences des utilisateurs et des spécificités du domaine. Grâce à cette modélisation, nous assurons une compréhension claire et exhaustive des besoins opérationnels, facilitant ainsi le développement d'une solution optimisée et parfaitement adaptée aux attentes des utilisateurs.

II.4.1 Diagrammes de cas d'utilisation

Un diagramme de cas d'utilisation est un outil de modélisation qui représente les interactions entre les acteurs (utilisateurs ou systèmes externes) et un système. Il illustre les fonctionnalités proposées par le système sous forme de cas d'utilisation, en mettant en lumière la manière dont ces fonctionnalités sont utilisées par les acteurs. Ce type de diagramme est particulièrement utile pour analyser les besoins des utilisateurs et les exigences fonctionnelles du système. Il est couramment employé lors de la phase de conception pour visualiser les attentes et les interactions avec le système, sans détailler les aspects techniques ou les séquences d'actions [22].

Dans la section suivante, nous présenterons les différents diagrammes de cas d'utilisation correspondant à chaque acteur identifié.

II.4.1.1 Le diagramme de cas d'utilisation de l'administrateur

L'administrateur a pour rôle de gérer les différentes fonctionnalités de l'application **Échange+**, notamment le contrôle des annonces et des commentaires, la supervision des transactions, ainsi que l'administration des comptes utilisateurs. Il est également responsable du suivi des statistiques et des revenus générés, ainsi que de la gestion du système de points. **La figure 9** ci-dessous représente le diagramme de cas d'utilisation des tâches de l'administrateur.

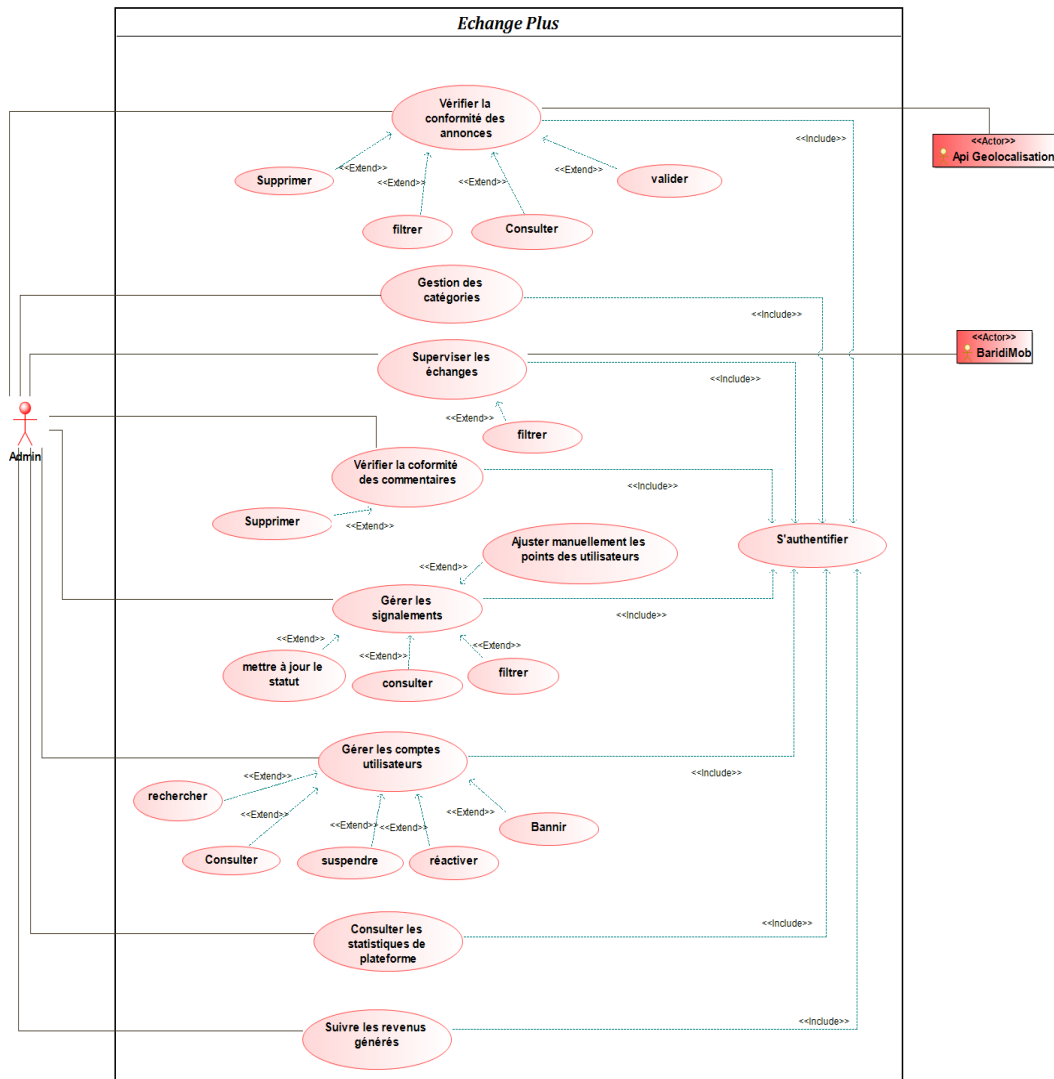


Figure 9: Diagramme de cas d'utilisation des tâches de l'administrateur

II.4.1.2 Le diagramme de cas d'utilisation de l'utilisateur

L'utilisateur joue un rôle central dans l'application **Échange+**, il peut publier des annonces et rechercher des services, recevoir des notifications et effectuer des échanges via plusieurs modes. Il peut aussi évaluer les prestataires et signaler des problèmes. **La figure 10** ci-dessous présente le diagramme de cas d'utilisation des tâches de l'utilisateur.

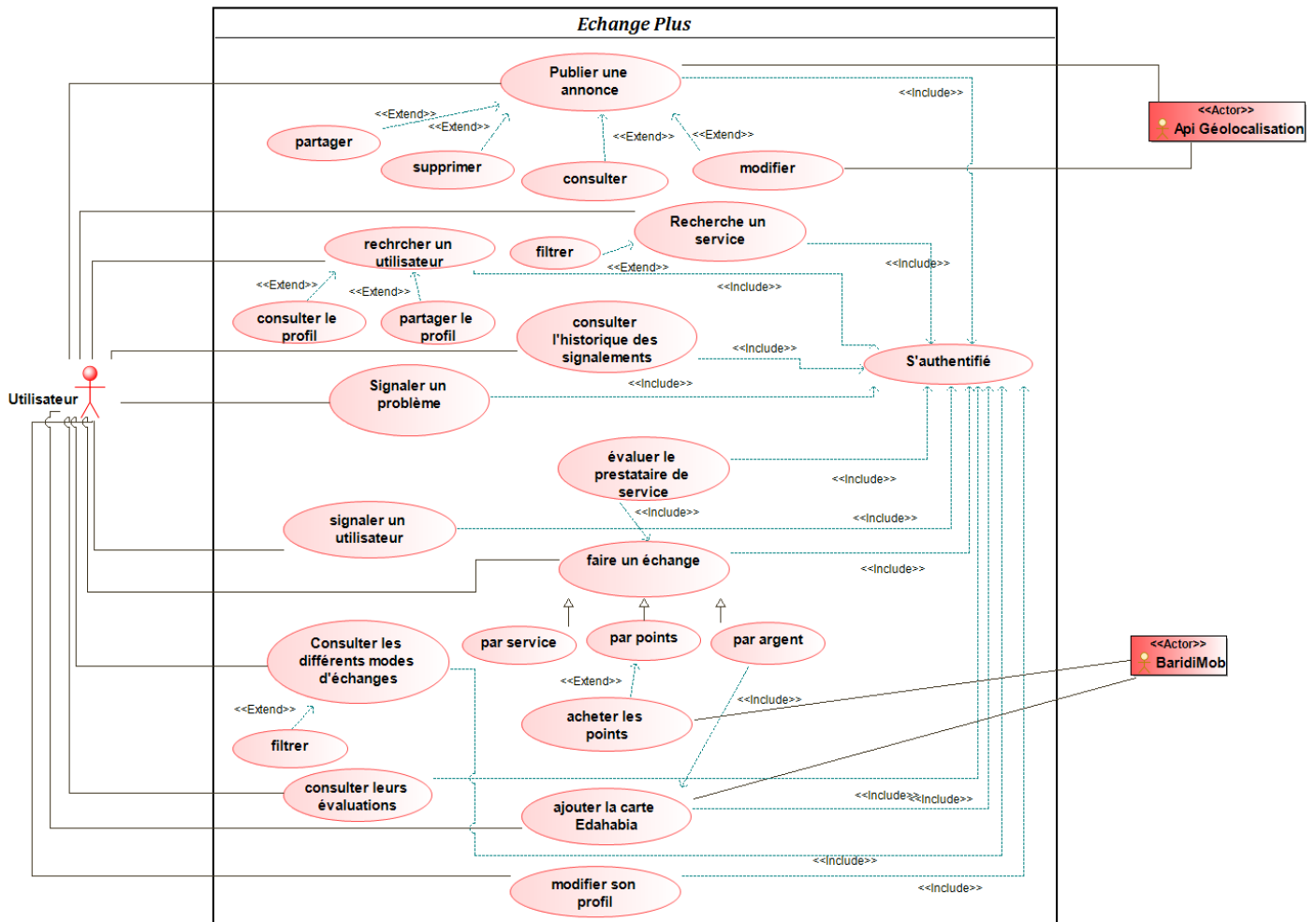


Figure 10: Diagramme de cas d'utilisation des tâches de l'utilisateur

II.4.2 Diagrammes de séquence

Un diagramme de séquence est un outil de modélisation UML qui représente les interactions entre les objets d'un système sous forme de messages échangés au fil du temps. Il illustre l'ordre des interactions et le déroulement des processus, facilitant ainsi la compréhension du comportement dynamique du système. Cet outil est particulièrement utile pour analyser et concevoir les scénarios d'exécution du système [23].

II.4.2.1 Diagramme de séquence des cas d'utilisation de l'administrateur

Dans cette partie, nous allons présenter les diagrammes de séquences de quelques cas d'utilisation de l'administrateur :

Vérifier la conformité des annonces : La figure 11, ci-dessous, représente les interactions liées au cas d'utilisation "Vérifier la conformité des annonces". L'administrateur a la possibilité de contrôler les annonces publiées en accédant à la fonctionnalité de gestion des annonces. Il peut consulter la liste des annonces, afficher les détails d'une annonce spécifique et décider de

sa conformité. Si l'annonce est jugée non conforme, elle est supprimée, et une notification est envoyée à l'utilisateur concerné. Dans le cas contraire, l'annonce reste inchangée.

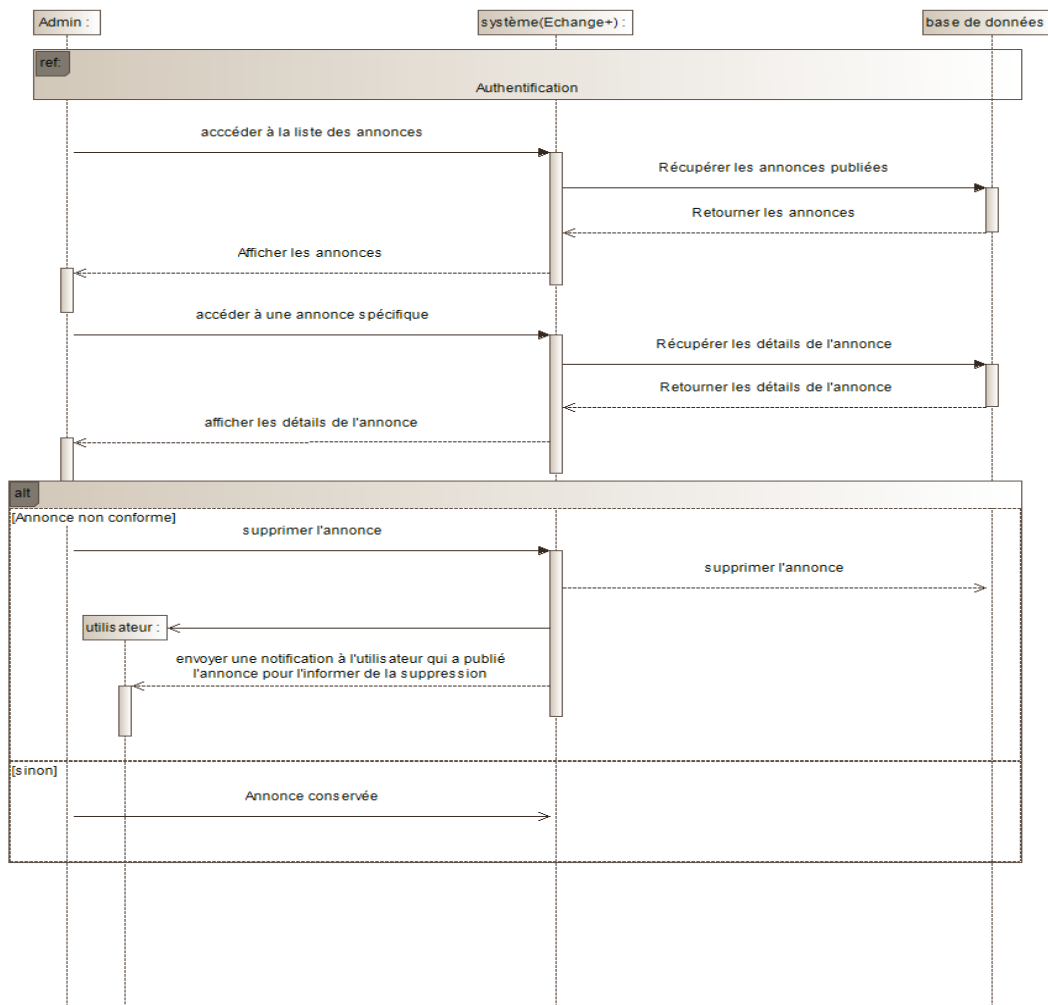


Figure 11: Diagramme de séquence du cas d'utilisation « vérifier la conformité des annonces »

Superviser les échanges : la figure 12 illustre les interactions du cas d'utilisation "Superviser les échanges". L'administrateur s'authentifie, accède à la liste des échanges et peut en consulter les détails. Selon le mode d'échange sélectionné (Service-Service, Service-Points ou Service-Argent), le système interroge la base de données pour récupérer la liste correspondante et l'affiche à l'administrateur.

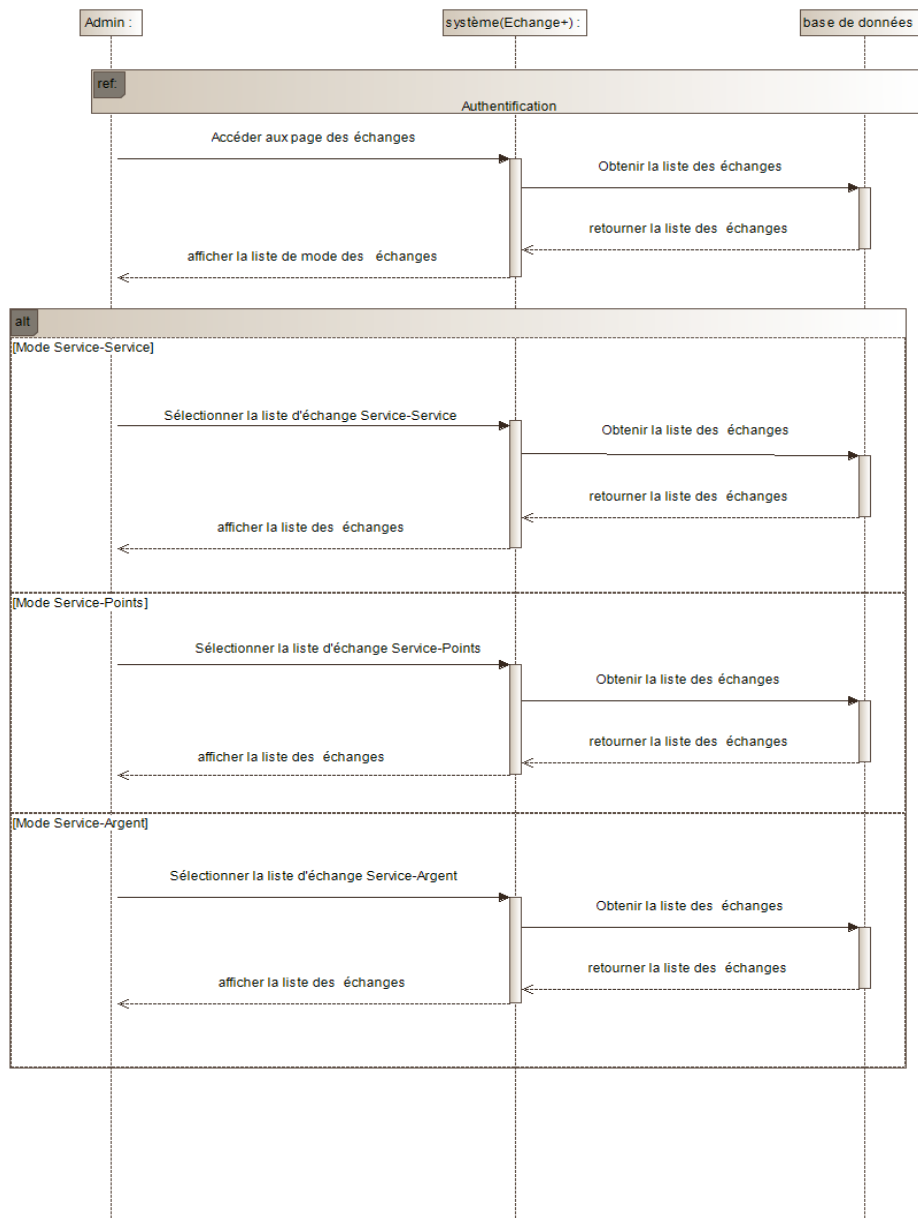


Figure 12: Diagramme de séquence du cas d'utilisation « superviser les échanges »

II.4.2.2 Diagramme de séquence des cas d'utilisation de l'utilisateur

Les diagrammes de séquence suivants illustrent quelques cas d'utilisation de l'utilisateur :

Publier une annonce : La figure 13, ci-dessous, représente les interactions liées au cas d'utilisation "Publier une annonce". L'utilisateur peut accéder à la fonctionnalité de publication d'annonce en s'authentifiant. Il remplit un formulaire contenant les détails requis. Si le formulaire est incomplet, un message d'erreur s'affiche, et l'utilisateur doit le compléter avant de soumettre à nouveau l'annonce. Une fois le formulaire correctement rempli, le système valide les informations et enregistre l'annonce. L'utilisateur reçoit alors une confirmation de mise en ligne et peut consulter son annonce publiée.

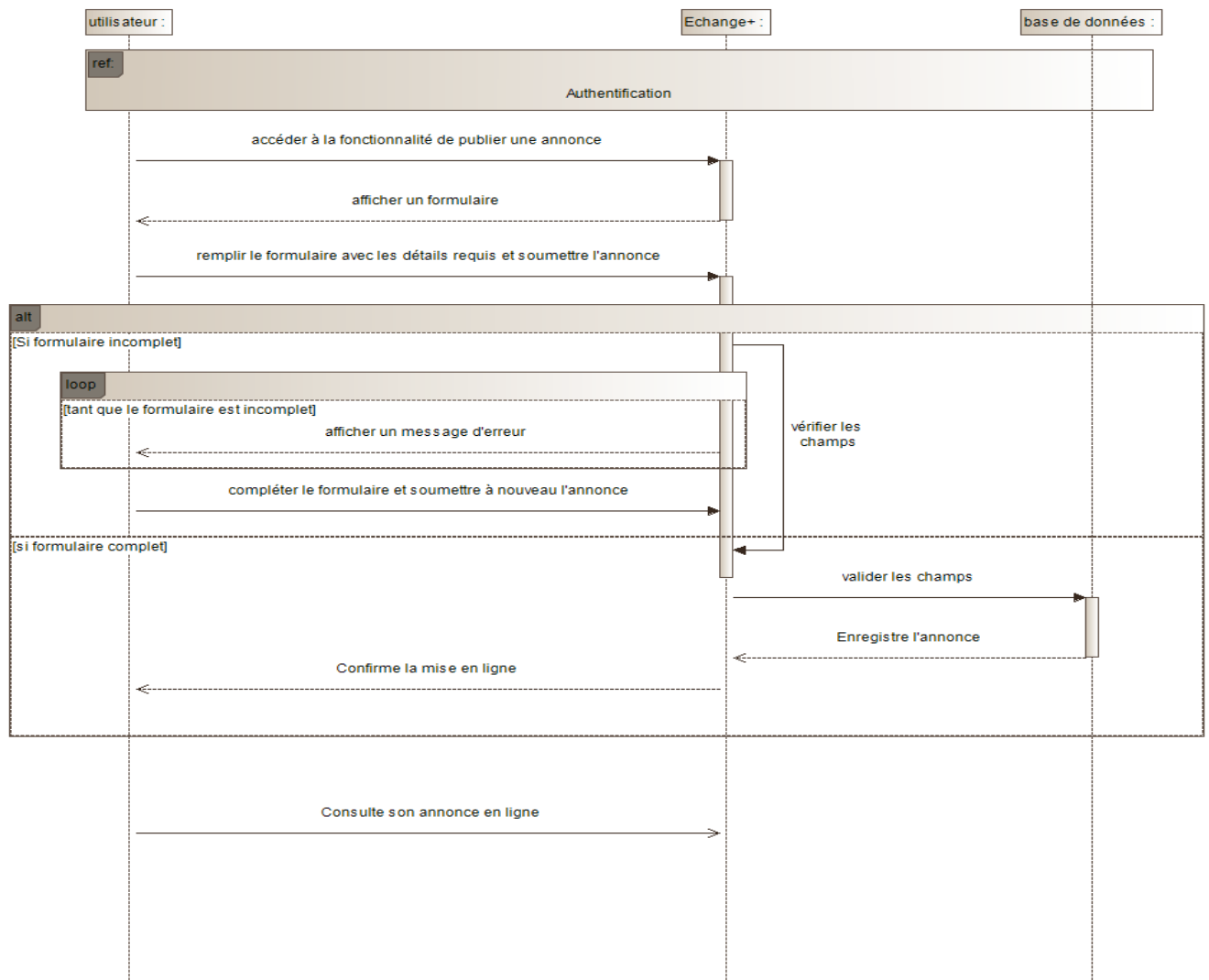
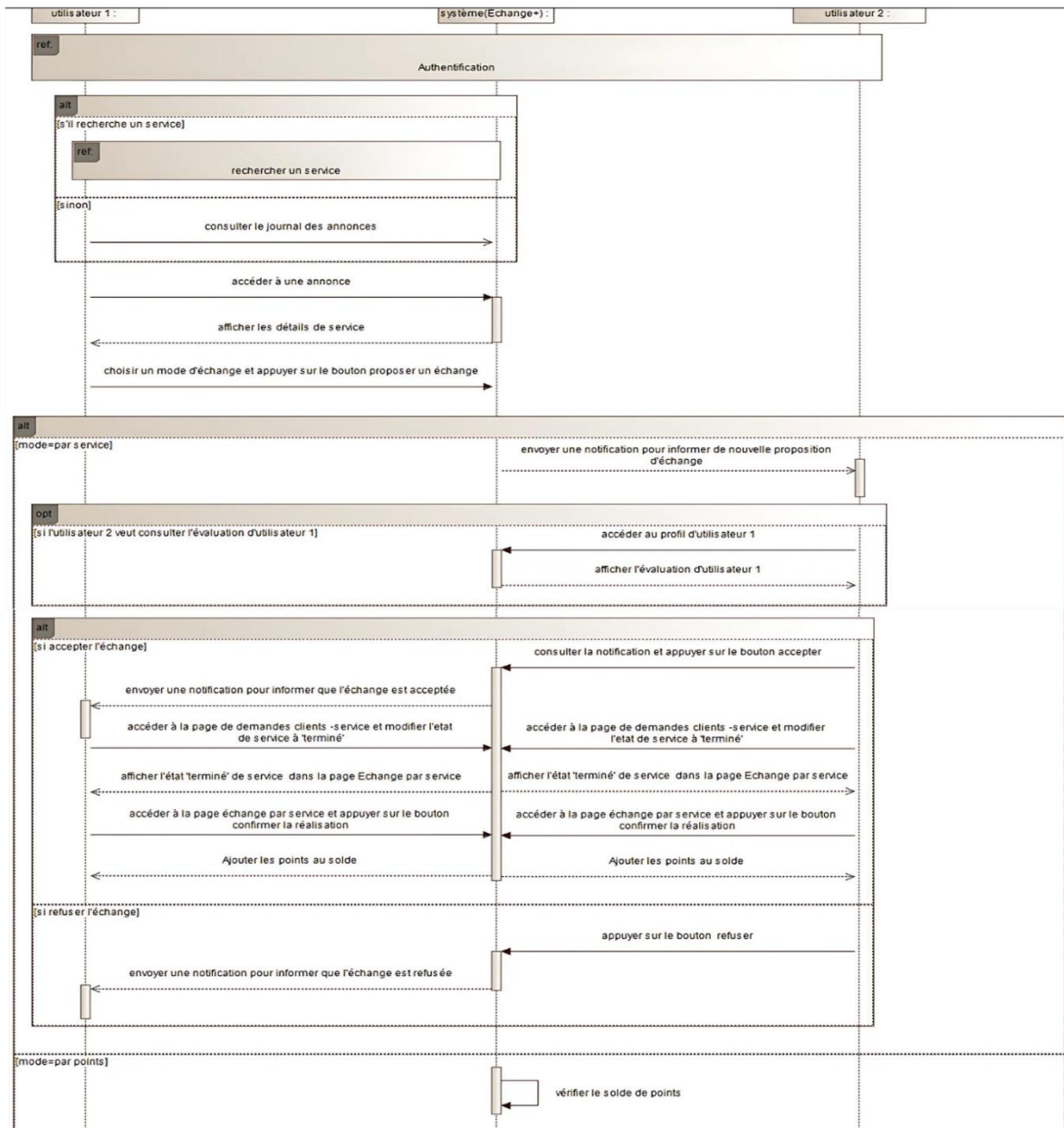


Figure 13: Diagramme de séquence du cas d'utilisation « publier une annonce »

Faire un échange : La figure 14 illustre les interactions du cas d'utilisation "Faire un échange". L'utilisateur s'authentifie, sélectionne un service et propose un échange selon l'un des modes suivants :

- **Service :** Le prestataire accepte ou refuse. En cas d'acceptation, l'échange est validé.
- **Points :** Le solde est vérifié. Si insuffisant, une boîte de dialogue apparaît, sinon l'échange est validé.

- **Argent** : L'utilisateur ajouter sa carte Edahabia, en cas de solde insuffisant, une boite de dialogue apparaît, sinon l'échange est validé.



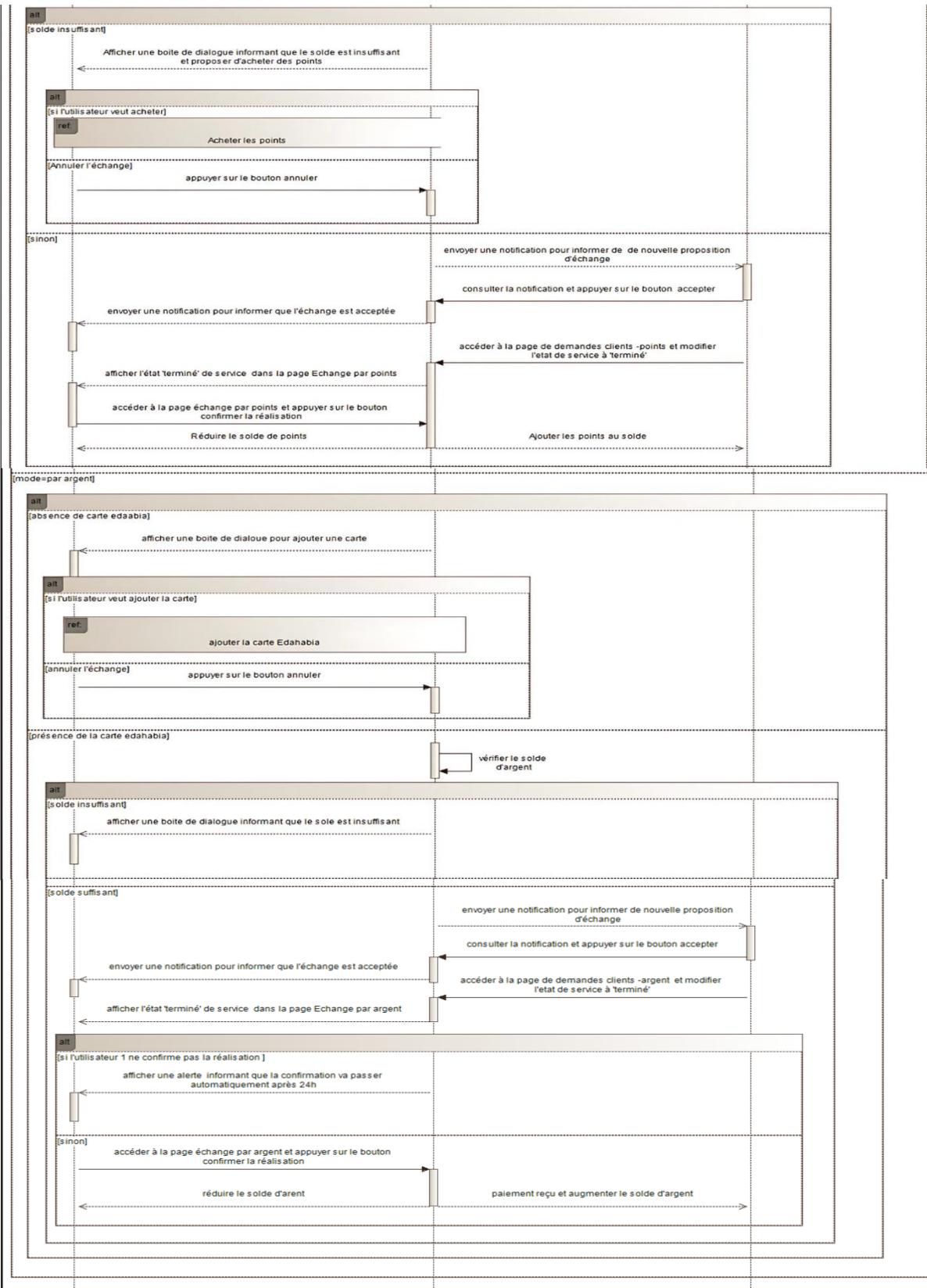


Figure 14: Diagramme de séquence du cas d'utilisation « faire un échange »

II.4.3 Base de données NoSQL :

II.4.3.1 Définition :

Une base de données NoSQL est une base de données « non relationnelle ». Il est possible d'y stocker des données sous une forme non structurée, sans suivre de schéma fixe. Les jointures ne sont plus nécessaires, et la scalabilité est facilitée [24].

II.4.3.2 Les caractéristiques de base de données NoSQL

- **Modèle non relationnel** : La principale particularité des bases de données NoSQL est qu'elles ne suivent pas le modèle relationnel et ne présentent pas de tableaux sous forme de colonnes fixes. Ces bases de données ne nécessitent pas de normalisation de données ou de mapping relationnel. Il est possible d'interagir sans utiliser de langages de requête complexe.
- **Schéma flexible ou absent** : Une autre particularité est l'absence ou la flexibilité des schémas. Il n'est pas nécessaire de définir de schéma des données, et les données de différentes structures peuvent donc être regroupées sur un même système.
- **Interface simple via APIs** : Les bases de données non relationnelles se distinguent aussi par une interface simple d'utilisation pour le stockage et la requête de données. Des APIs permettent de manipuler les données avec diverses méthodes de sélection. Les protocoles, basés sur le texte, reposent principalement sur HTTP REST avec JSON. On utilise en général un langage de requête NoSQL.
- **Système distribué et scalable** : La dernière caractéristique d'une base de données NoSQL est d'être distribuée. De multiples bases NoSQL peuvent être exécutées de façon distribuée, offrant des capacités de mise à l'échelle automatique et de basculement. Le concept ACID peut être délaissé au profit de l'élasticité et des performances. **La figure 15** représente les caractéristiques de base de données NoSQL.

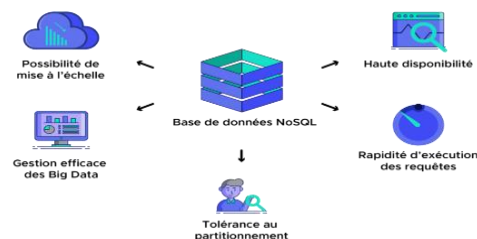


Figure 15: Caractéristiques de base de données NoSQL [24]

II.4.3.3 Structure de la base de données de l'application Echange+

Voici la structure de notre système, incluant des collections avec document, en pièce jointe:

La figure 16 illustre les collections "achat_point" et "categories". La collection "achat_point" gère les transactions liées à l'achat de points par les utilisateurs, tandis que "categories" organise les différentes catégories d'annonces ou de services disponibles sur la plateforme. Ces collections sont essentielles pour structurer les offres et faciliter les échanges entre utilisateurs.

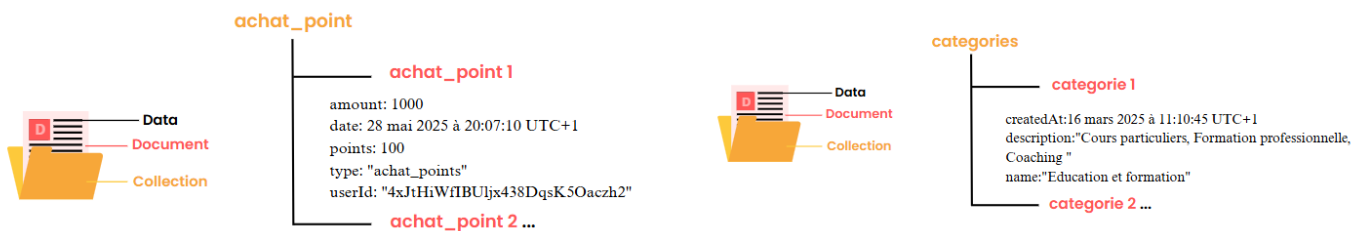


Figure 16: Collections « achat_point », « categories »

La figure 17 présente les collections "annonces" et "echange_argent". La collection "annonces" contient les publications des utilisateurs pour proposer des biens ou services, tandis que "echange_argent" gère les transactions monétaires entre utilisateurs. Ces collections sont centrales pour le fonctionnement des échanges sur la plateforme.

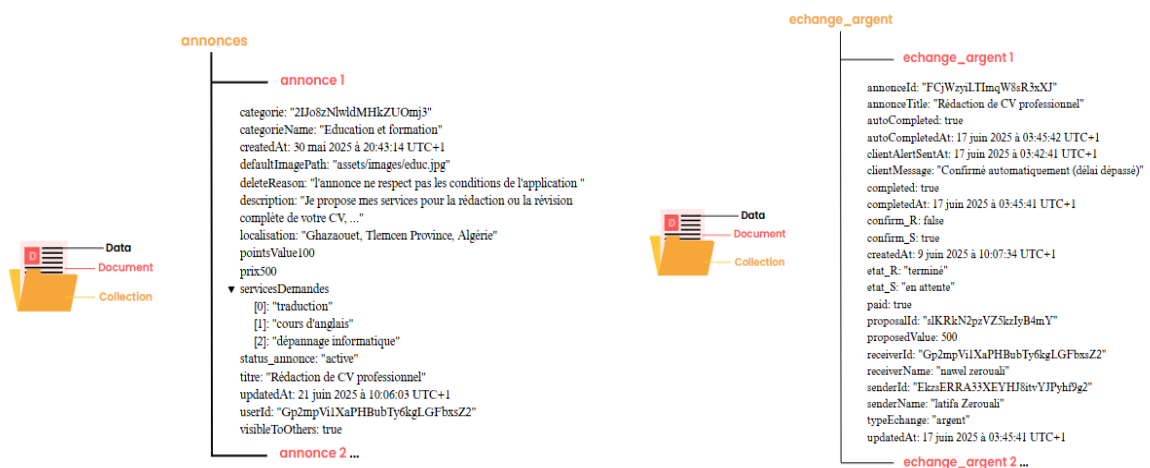


Figure 17: Collection « annonces », « echange_argent »

La figure 18 montre les collections "échange_points" et "échange_service". La première gère les échanges basés sur des points, un système potentiel de récompense ou de monnaie virtuelle, tandis que la seconde traite des échanges de services entre utilisateurs. Ces collections permettent de diversifier les modes d'échange sur la plateforme.

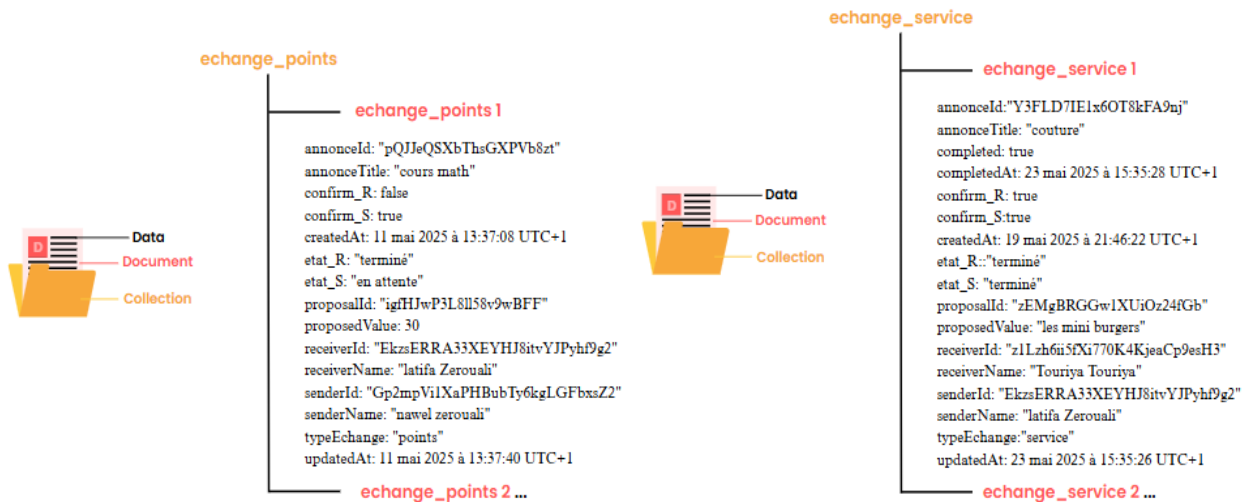


Figure 18: Collection « échange_points », « échange_service »

La figure 19 présente la collection "échange_proposals" est dédiée à la gestion des propositions d'échange entre utilisateurs. Elle joue un rôle clé dans la négociation et la validation des échanges, en stockant les détails des services et des contre-services .

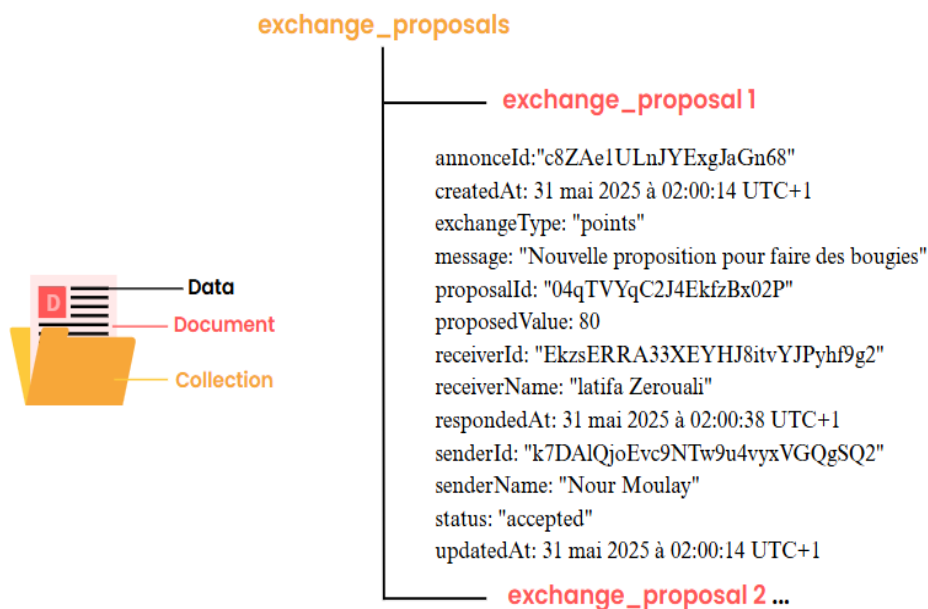


Figure 19: Collection « échange_proposals »

La figure 20 représente la collection "notification", qui stocke les alertes et messages envoyés aux utilisateurs pour les informer des activités pertinentes sur la plateforme. Elle gère principalement deux types de notifications :

1-Nouvelle proposition d'échange : Alertes envoyées lorsqu'un utilisateur reçoit une nouvelle offre concernant une annonce qu'il a publiée.

2-Réponse à une proposition : Notifications informant un utilisateur que sa proposition d'échange a été acceptée, refusée l'autre utilisateur.

Cette collection joue un rôle crucial dans la fluidité des interactions entre utilisateurs, en maintenant un suivi en temps réel des échanges et en favorisant l'engagement sur la plateforme.

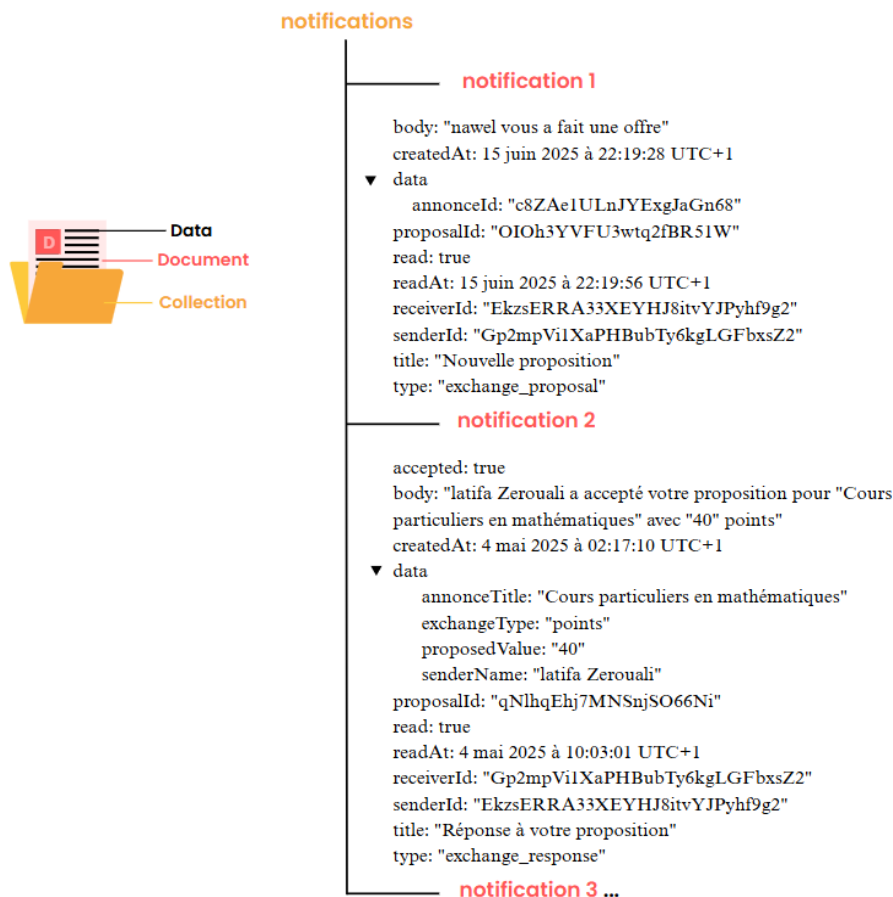


Figure 20: Collection « notification »

La figure 21 présente trois collections clés liées à la gestion financière et à la modération de la plateforme :

La collection « platform_commissions »: Enregistre les commissions prélevées par la plateforme sur les transactions impliquant des échanges de points.

La collection « platform_commissions_argent » : Gère les commissions perçues sur les transactions monétaires (échanges en argent réel).

La collection « reports_probleme » : Stocke les signalements techniques ou fonctionnels remontés par les utilisateurs.

Ces collections garantissent la transparence des revenus de la plateforme et la résolution des problèmes techniques, assurant ainsi un environnement fiable pour les utilisateurs.

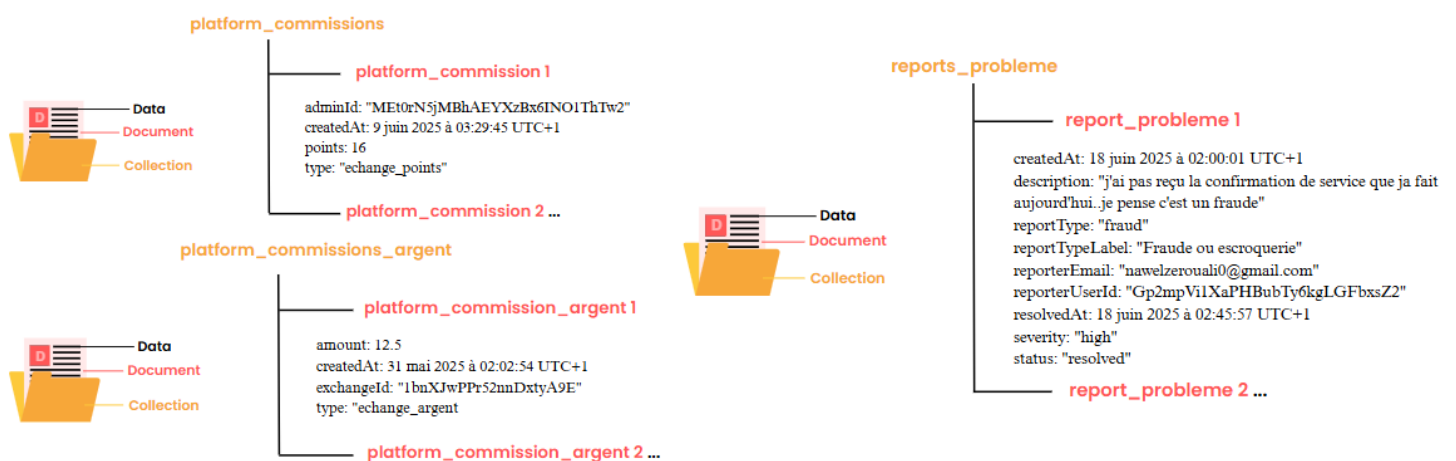


Figure 21: Collection « platform_commissions », « platform_commissions_argent », « reports_probleme »

La figure 22 illustre les collections "reports_user" et "user_ratings". La première gère les signalements d'utilisateurs pour comportement inapproprié, tandis que la seconde stocke les évaluations et commentaires laissés par les utilisateurs après une transaction. Ces collections contribuent à la modération et à la confiance sur la plateforme.

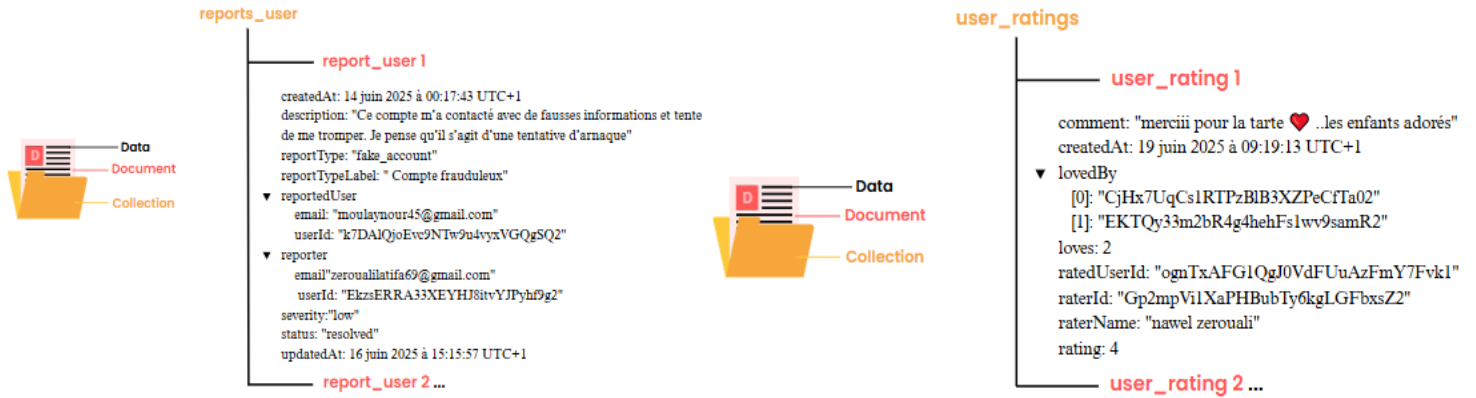


Figure 22: Collection « reports_user », « user_ratings »

La figure 23 présente la collection « users » stockant les données utilisateurs : identifiants, coordonnées . Essentielle pour l'authentification et la gestion des comptes, elle constitue le socle de toutes les interactions sur la plateforme. Sa structure optimisée garantit sécurité et performance.

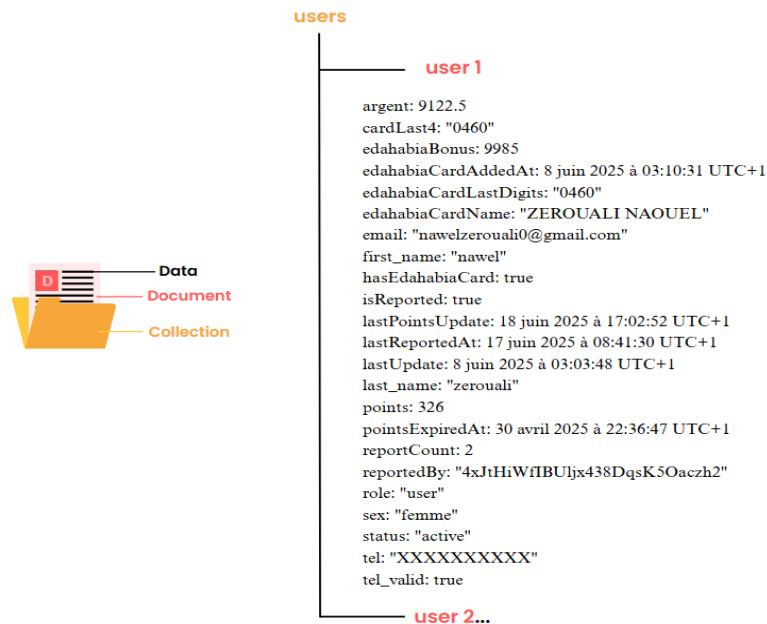


Figure 23: Collection « users »

II.5 Conclusion

La phase de conception est une étape clé dans le développement d'un projet, permettant d'optimiser son architecture et d'assurer sa viabilité. Une conception rigoureuse apporte de nombreux avantages, notamment en termes de clarté fonctionnelle et d'optimisation des interactions entre les différents acteurs du système.

Ce chapitre a offert une vue d'ensemble de notre projet. Nous avons tout d'abord effectué une étude de l'existant afin d'identifier les points forts et les limites des solutions actuelles. Ensuite, nous avons défini la structure et le fonctionnement de notre application à travers plusieurs diagrammes UML. Les diagrammes de cas d'utilisation ont permis de mettre en évidence les principales fonctionnalités, les diagrammes de séquence ont détaillé les échanges entre les acteurs et l'application, tandis que la structure des collections/documents a modélisé les entités et leurs relations.

Dans le prochain chapitre, nous nous concentrerons sur l'implémentation de notre solution, en passant de la conception théorique à la réalisation technique de notre application.

Chapitre III :
Réalisation et implémentation de
l'application « *Echange+* »

III.1 Introduction

Dans ce chapitre, nous présentons les différentes étapes techniques de développement de notre application mobile. Nous commençons par exposer l'**architecture générale** mise en place, avant de détailler le **processus de développement** suivi, fondé sur une méthode agile. Nous poursuivons avec les **outils et langages** que nous avons choisis, les **difficultés rencontrées**, ainsi que les solutions apportées. Ce chapitre inclut également un **planning de réalisation** sous forme de diagramme de Gantt, une **présentation de l'application**, et les **fonctionnalités développées**. L'objectif est de mettre en lumière l'ensemble du travail réalisé, de la conception à la mise en œuvre technique.

III.2. Architecture de notre application

L'architecture de notre application repose sur une organisation bien structurée, assurant à la fois modularité, maintenabilité et performance. Nous avons opté pour une architecture basée sur le modèle MVVM, largement utilisé dans les projets Flutter.

III.2.1. Définition du modèle MVVM

Le modèle MVVM, permet de séparer de manière nette la logique métier et la logique de présentation d'une application de son interface utilisateur (IU) [24]. La **figure 24** ci-dessous illustre l'architecture MVVM (Model — View — ViewModel) :

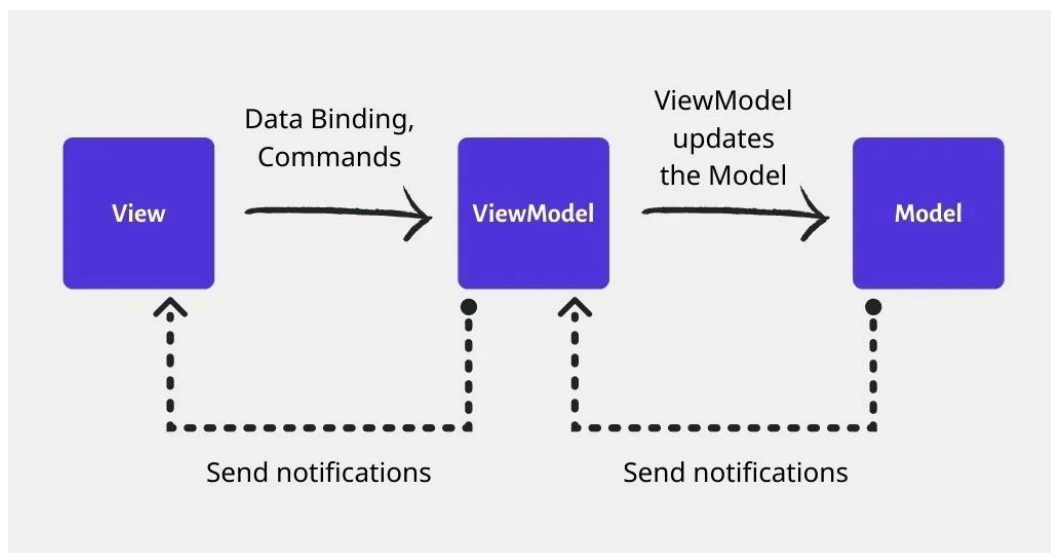


Figure 24: Interaction (Model — View — ViewModel) dans le cadre du pattern MVVM [24]

III.2.2. Structure du modèle MVVM

De manière générale, MVVM est structuré en 3 grandes parties [25] :

- **Model**

Le modèle représente l'ensemble des entités et des modèles de données dont aura besoin la vue pour présenter les informations à l'utilisateur.

- **View**

La Vue gère l'affichage et les interactions utilisateur. Elle montre les données du ViewModel et lui transmet les actions. En MVVM, elle se contente d'observer et d'afficher, sans logique métier, ce qui rend le code plus clair et facile à modifier. Par exemple, un Fragment Android affiche des données et envoie les clics au ViewModel.

- **ViewModel**

Le ViewModel représente la couche qui gère toute la logique métier de l'application, elle concerne les opérations de traitements qui s'opèrent entre la base de données et la vue, par exemple le formatage de données, l'envoi de paramètres vers les requêtes, la réception des données.

III.3. Structure de notre application :

Nous avons utilisé Flutter (Dart) pour développer une application mobile destinée aux utilisateurs, ainsi qu'une interface d'administration développée également avec Flutter. L'application mobile sera disponible sur les plateformes Android et iOS.

Pour la gestion des données et des fonctionnalités, nous avons choisi d'utiliser Firebase. Cette plateforme propose des services variés tels que l'authentification, le stockage, les fonctions en tant que service (FaaS), l'analyse, et bien plus encore. Certes, Firebase est une plateforme évolutive offrant de nombreuses fonctionnalités essentielles. Son principal avantage réside dans sa rapidité et sa simplicité d'intégration dans une application Flutter (quelques commandes

Chapitre III Réalisation et implémentation de l'application « Echange+ »

suffisent), ainsi que dans la facilité d'interaction entre Flutter et les services Firebase [26]. Comment il est illustré dans la **figure 25** ci-dessous :

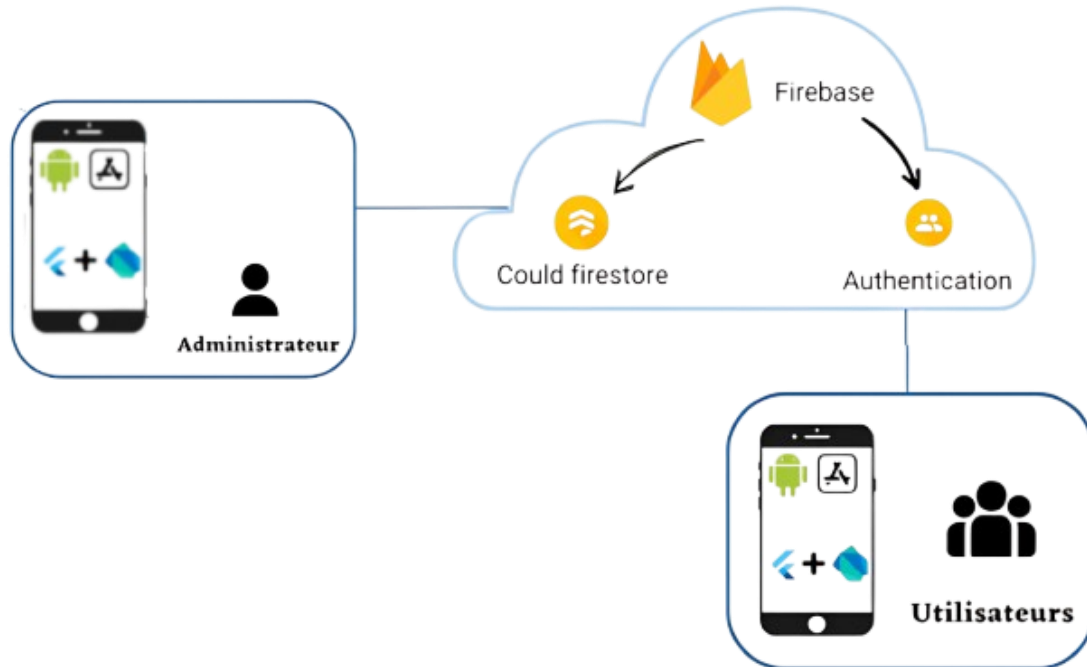


Figure 25: Architecture de notre application “Echange+”

Nous avons accédé à ces services Firebase via FlutterFire, la suite officielle de plugins Flutter pour Firebase, ce qui nous a permis d'intégrer ces services directement dans notre code Flutter de manière native, efficace et sécurisée. Plus précisément, Nous avons utilisé les services suivants de Firebase. La **figure 26** ci-dessous montre le logo de FlutterFire :



Figure 26: Logo FlutterFire



Firebase Authentication

Figure 27: Logo Firebase Authentication.

- **Firebase Authentication** fournit des services de backend, des SDK faciles à utiliser et des bibliothèques d'UI prêtes à l'emploi pour authentifier les utilisateurs dans l'application. Il prend en charge l'authentification à l'aide de mots de passe, de numéros de téléphone, de fournisseurs d'identité fédérés populaires tels que Google, Facebook et Twitter, etc. La **figure 27** ci-dessous montre le *logo Firebase Authentication* [27].

Ce service a été intégré via le plugin `firebase_auth` de FlutterFire.

- **Cloud Firestore** est une base de données flexible et évolutive pour les mobiles, le Web, et le développement de serveurs via Firebase et Google Cloud. Firebase Realtime Database synchronise les données entre les applications clientes en temps réel, et offre un fonctionnement hors connexion pour les mobiles et le Web. Il est possible de créer des applications réactives qui continuent de fonctionner même avec une mauvaise connexion ou sans Internet. Cloud Firestore permet également une intégration parfaite avec les autres produits Firebase et Google Cloud, y compris Cloud Function. La **figure 28** ci-dessous montre le *logo Firebase Cloud Firestore* [28].



Figure 28: logo Firebase Cloud Firestore.

Ce service a été intégré via le plugin `cloud_firestore` de FlutterFire.

III.4. Processus de développement

Nous avons opté pour la méthode **SCRUM** afin de gérer le développement de notre application mobile Échange+, dédiée à l'échange de services entre particuliers. Ce choix repose sur la capacité de cette méthode à offrir une visibilité constante, une adaptabilité optimale, ainsi qu'une amélioration continue à chaque étape du projet. La méthode SCRUM s'est révélée

particulièrement adaptée à un contexte académique, où les contraintes de temps et de coordination exigent une gestion souple et itérative du développement.

III.4.1. Définition de la méthode SCRUM

SCRUM n'est pas une méthode mais un cadre de travail (*framework*) permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible [31].

Le **framework Scrum** permet de travailler en équipe pour faire de l'amélioration continue sur des livraisons itératives incrémentales de produits afin de satisfaire les clients. **Scrum** est fondé sur la théorie du contrôle empirique de processus et soutenu par 3 piliers fondamentaux [31] :

- **La transparence** : le fait d'être honnête, de ne rien avoir à cacher, de travailler ensemble au succès du produit/projet en rendant les aspects importants du processus visibles à tous ceux qui sont responsables des résultats.
- **L'inspection** : le fait de pouvoir s'entraider et inspecter les artefacts Scrum et l'état d'avancement par rapport à un Objectif de Sprint afin de détecter les écarts indésirables
- **L'adaptation** : le fait de s'adapter aux changements en général, changements de produit, changements de la façon de faire...

III.5. Diagramme de Gantt

Les diagrammes de Gantt sont un outil de gestion de projet agile utilisé pour planifier et suivre l'avancement des différentes tâches et activités d'un projet. Ils proposent une chronologie visuelle qui affiche la liste des tâches à effectuer, leurs dates de début et de fin, les étapes importantes, les dépendances entre les tâches et les personnes assignées.

En tant que feuille de route indispensable, les diagrammes de Gantt aident les équipes et les organisations à planifier et coordonner leurs tâches avec efficacité. La mise en valeur des interconnexions entre les tâches favorise une collaboration et un alignement fluide entre toutes les parties prenantes du projet, en veillant à ce que tout le monde soit sur la même longueur d'onde et travaille à atteindre le même objectif [32]. La **figure 29** ci-dessous illustre le diagramme de Gantt de l'application "Echange+".

● Application « Echange+ » - Diagramme de Gantt du projet ●

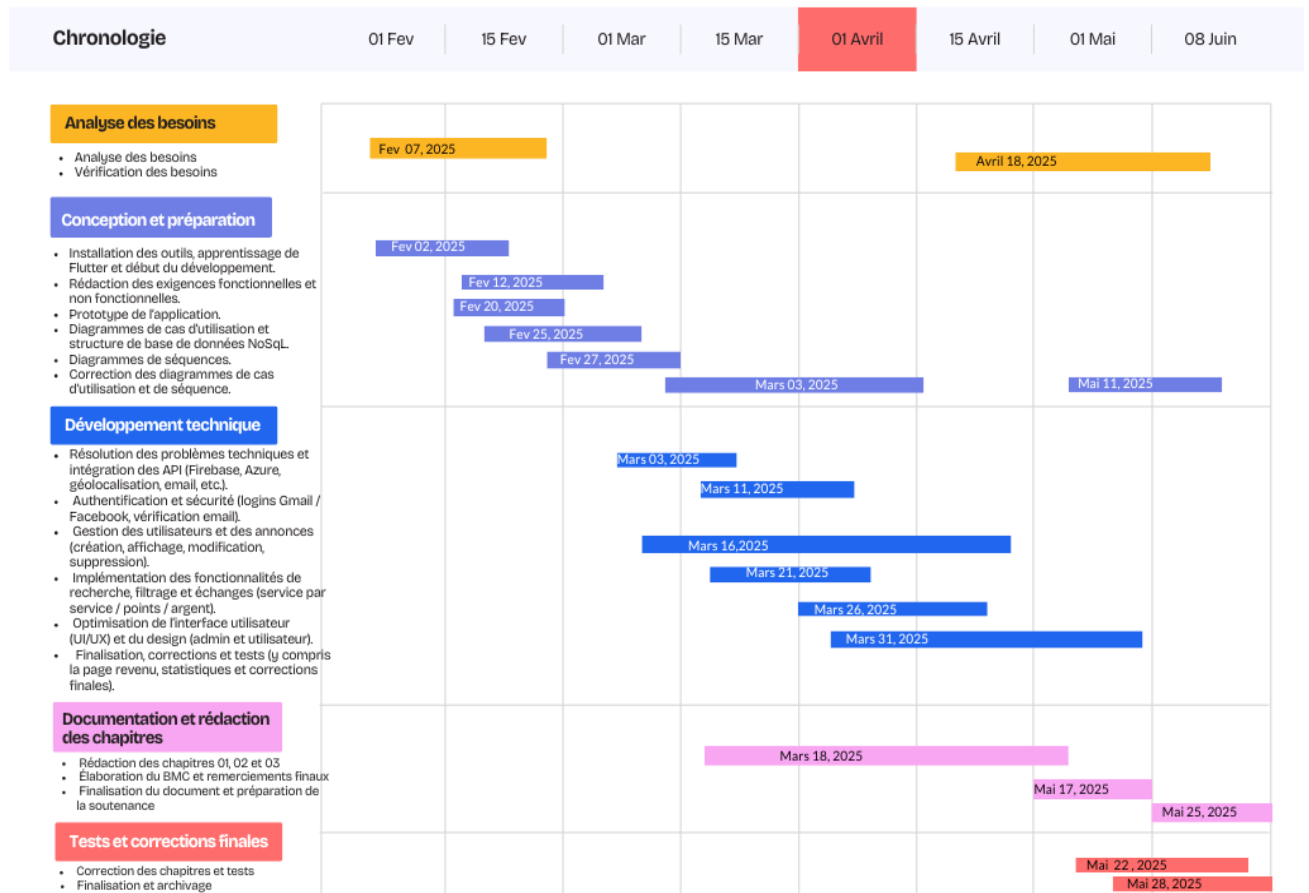


Figure 29: Diagramme de Gantt de l'application " Echange+".

III.6. Les Outils et les langages utilisés

Dans cette section, nous allons vous présenter les outils que nous avons utilisés lors du développement de notre application mobile :

III.6.1. Modelio

Un logiciel open source de modélisation UML, est accessible sur les systèmes d'exploitation Windows, Linux et Mac [33]. On a utilisé ce logiciel pour créer les différents diagrammes de notre projet pour aider à la compréhension du système.

III.6.2. Visual Studio code

Est un éditeur de code simplifié, gratuit et développé en open source par Microsoft. Ses fonctionnalités sont riches et variées. Parmi celles-ci, on retrouve la prise en charge de plusieurs

langages de programmation, notamment C++, Python, Java, et bien d'autres. Il dispose également d'un débogueur intégré, facilitant ainsi la phase d'édition, de compilation et de débogage, ce qui nous a aidé à accélérer le processus de développement de notre module [34].

III.6.3. GitHub

GitHub est une plateforme [open source](#) de gestion de versions et de [collaboration](#) destinée aux développeurs de logiciels. Livrée en tant que logiciel à la demande ([SaaS](#), Software as a Service), la solution GitHub a été lancée en 2008 [35].

III.6.4. Flutter

Flutter est un framework de **développement d'applications multiplateforme**, conçu par Google, dont la première version a été publiée sous forme de projet open source à la fin de l'année 2018. Flutter met à disposition une grande variété de bibliothèques d'éléments d'UI standard pour Android et iOS [36].

III.6.5. Dart

Dart est un langage optimisé pour le client, permettant de développer rapidement des applications sur toutes les plateformes. Son objectif est d'offrir le langage de programmation le plus productif pour le développement multiplateforme, associé à une plateforme d'exécution flexible pour les frameworks d'applications [37].

III.6.6. Canva

Lancé en 2013, Canva est un outil de design et de communication visuelle en ligne dont la mission est de permettre à tout le monde de concevoir et de publier selon ses envies [38].

III.7. Défis Du Projet

Dans cette section nous allons parler des difficultés et défis rencontrés au cours de la réalisation de notre projet.

III.7.1. Problèmes d'intégration de Firebase

Une fois l'environnement de développement stabilisé, une nouvelle série de défis est apparue : l'intégration de Firebase pour l'authentification et la base de données en temps réel.

Les principaux problèmes rencontrés étaient liés à :

- La compatibilité entre la version de Flutter utilisée et les SDK Firebase disponibles,
- Les conflits entre les versions des packages dans le fichier pubspec.yaml,

Chapitre III Réalisation et implémentation de l'application « Echange+ »

- Les erreurs de synchronisation Gradle liées aux dépendances Android (Firebase Authentication, Firebase Core , Cloud Firestore, etc.).

Malgré une tentative d'upgrade de Flutter vers une version plus récente pour espérer régler ces conflits automatiquement, les erreurs persistaient. En réalité, le problème venait du fait que certains packages Firebase avaient besoin de versions précises de Gradle, ou de versions précises du `compileSdkVersion` et `minSdkVersion` dans les fichiers `android/build.gradle` du projet.

Après une analyse manuelle et beaucoup de recherches, nous avons compris qu'il fallait aligner manuellement les versions des packages dans le fichier `pubspec.yaml` et adapter le projet Android aux recommandations de compatibilité imposées par Firebase.

Les actions correctrices entreprises étaient :

- Mise à jour des dépendances Firebase dans `pubspec.yaml`,
- Modification des fichiers `android/build.gradle` pour fixer les versions de `compileSdkVersion` et `minSdkVersion`,
- Nettoyage du projet (`flutter clean`) et reconstruction des dépendances (`flutter pub get`).

Après ces ajustements précis, l'application a pu se connecter correctement à Firebase, permettre l'authentification des utilisateurs, et gérer la base de données sans erreurs critiques.

III.7.2. Problèmes liés au stockage des images :

Lors de la conception de l'application, nous avons rencontré des difficultés concernant le stockage et la récupération des images. Initialement, nous avons prévu d'utiliser Firebase Storage pour stocker les photos de profil et les images des annonces. Cependant, nous avons découvert que certaines fonctionnalités de Firebase (notamment la gestion avancée des images et leur affichage sécurisé) sont devenues payantes au-delà du plan gratuit "Spark", ce qui a représenté une contrainte pour notre projet non financé.

Lorsque nous avons utilisé Azure Blob Storage, nous avons constaté qu'il n'était pas possible de choisir une clé publique pour accéder aux fichiers. Azure ne fournit, dans notre cas, qu'une **clé privée**, ce qui empêche toute récupération directe des images via l'application mobile. Or, pour afficher les photos des utilisateurs dans l'interface, il nous fallait une **clé publique ou un lien accessible sans authentification complexe**, ce que Azure ne permettait pas dans notre configuration.

En résumé, ce problème a mis en lumière l'importance de bien anticiper les choix d'infrastructure cloud et de stockage dès le début du projet, notamment en tenant compte des contraintes budgétaires et des exigences de sécurité.

III.7.3. Difficultés avec Firebase Cloud Messaging (FCM)

La mise en place des notifications via Firebase Cloud Messaging (FCM) était essentielle pour notre application, notamment pour notifier les propositions d'échanges et les réponses (acceptation ou refus). Initialement, le système fonctionnait correctement, mais il a fini par s'interrompre sans avertissement, causant un grand stress dans notre équipe. Ce problème a ralenti l'avancement côté administration (calcul des revenus, suivi des signalements, supervision des modes de paiement, etc.).

Nous avons tenté d'éviter les coûts supplémentaires en utilisant des requêtes HTTP directes, mais la sécurisation des APIs Google Cloud nous a empêchés de continuer gratuitement. À terme, l'impossibilité de déployer certaines fonctionnalités serveurs a impacté le fonctionnement complet de l'application.

III.7.4. API Edahabia

Pour intégrer un système de paiement électronique avec l'API Edahabia (Algérie Poste), nous avons découvert que l'accès officiel nécessitait un registre de commerce. N'ayant pas ce document, et suivant les conseils du président des projets I2E, nous avons opté pour une simulation de la transaction dans notre projet, afin de pouvoir démontrer le fonctionnement théorique sans pour autant intégrer l'API réelle.

III.7.5. Envoi des notifications push

L'envoi des notifications devait être géré par les Cloud Functions de Firebase. Cependant, cette solution nécessite le passage à une formule payante (le Blaze Plan), ce qui n'était pas possible dans le contexte du projet.

Pour contourner cette limitation, un petit serveur a été développé avec Node.js, afin d'envoyer les notifications directement à partir du backend. La configuration a été soigneusement mise en place : le token FCM est bien généré et enregistré pour chaque utilisateur, les autorisations sont activées côté mobile, les canaux de notifications sont bien définis.

Malgré cela, un problème persiste. Les notifications arrivent bien dans l'application, mais elles ne s'affichent pas sur l'appareil (aucune alerte visible). Autrement dit, elles sont interceptées silencieusement par l'application sans déclencher d'affichage système.

III.8. Présentation de notre application “Echange +”

L'objectif de cette section est de démontrer comment chaque utilisateur interagit avec le système et de souligner les fonctionnalités associées à chaque rôle. Notre application mobile “Echange +” propose une plateforme dynamique dédiée à l'échange de services ou de biens entre particuliers. Elle distingue plusieurs rôles : l'utilisateur et l'administrateur du système. Chaque profil accède à un ensemble de fonctionnalités spécifiques, assurant une utilisation claire, sécurisée et adaptée à ses besoins. L'interface a été pensée pour être à la fois intuitive, ergonomique et réactive, avec un design fluide qui s'adapte à toutes les tailles d'écran.

L'application intègre des fonctionnalités essentielles : gestion des annonces (création, modification, suppression), recherche par localisation, système d'évaluation par étoiles et commentaires, notifications en temps réel, et suivi des échanges. Un espace administrateur permet de superviser les utilisateurs, de modérer le contenu et de consulter les revenus. Chaque fonctionnalité est conçue pour renforcer la confiance, fluidifier les interactions, et encourager les utilisateurs à participer activement à la plateforme.

III.8.1. Logo de l'application



Figure 30: Logo de l'application mobile “Echange +”

III.8.2. Les Interfaces de l'application

Cette section explore en détail toutes les interfaces de notre application "Echange +". Nous y présenterons les parcours utilisateur et administrateur, en illustrant les interactions clés et les fonctionnalités spécifiques associées à chaque rôle.

III.8.2.1. Côté Utilisateur

Cette section présente les **interfaces et le parcours utilisateur au sein de l'application mobile "Echange+"**. Elle décrira les fonctionnalités permettant aux utilisateurs de s'inscrire, se connecter, rechercher et publier des annonces, proposer et gérer des échanges, suivre leurs transactions, consulter leur profil et interagir avec d'autres membres.

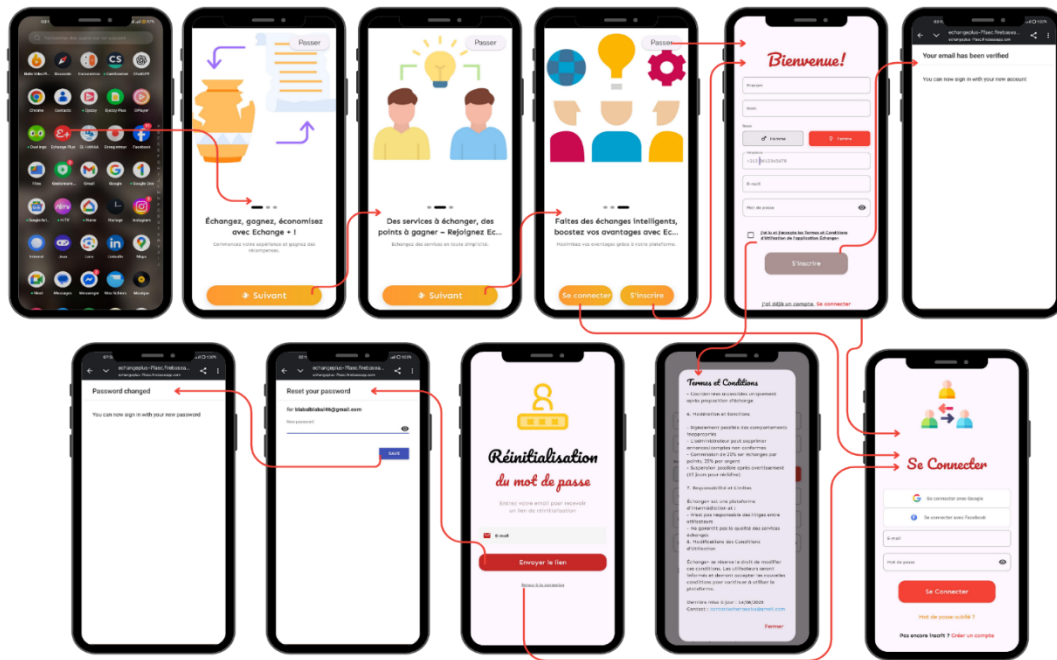


Figure 31: Interfaces de connexion et d'inscription "Echange+".

Cette figure 31 illustre le parcours utilisateur initial d'application « Echange + », couvrant les écrans d'introduction présentant les avantages de l'application, les processus d'inscription et de connexion, ainsi que le flux de réinitialisation de mot de passe. Elle inclut également l'écran des termes et conditions, assurant une prise en main complète pour tout nouvel utilisateur.

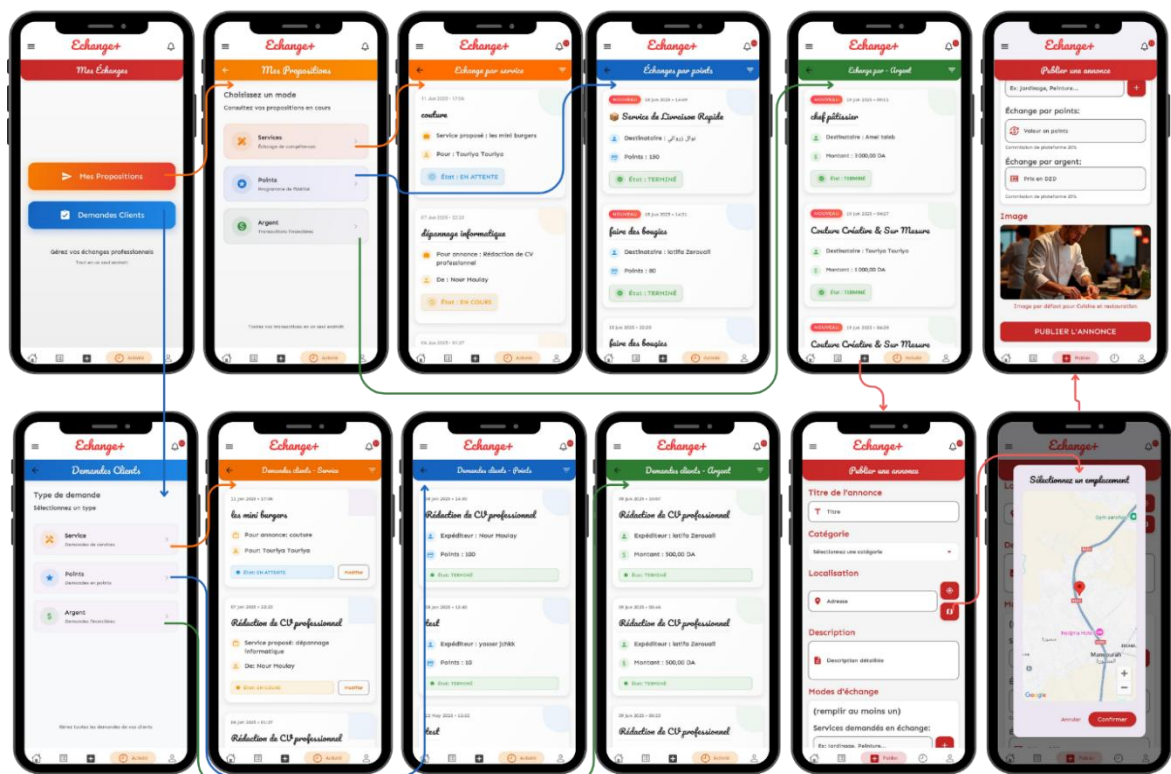


Figure 32: Interfaces de gestion des propositions et demandes utilisateur.

Chapitre III Réalisation et implémentation de l'application « Echange+ »

Cette figure 32 présente les interfaces de **gestion des annonces**. Elle couvre le **tableau de bord utilisateur**, le **processus de création/modification d'annonces** (par service, points, argent, avec image, titre, etc.), la **consultation des "Demandes Clients"** (filtrage par type), et la **sélection de localisation**. En somme, elle illustre la gestion des offres et des demandes sur la plateforme.

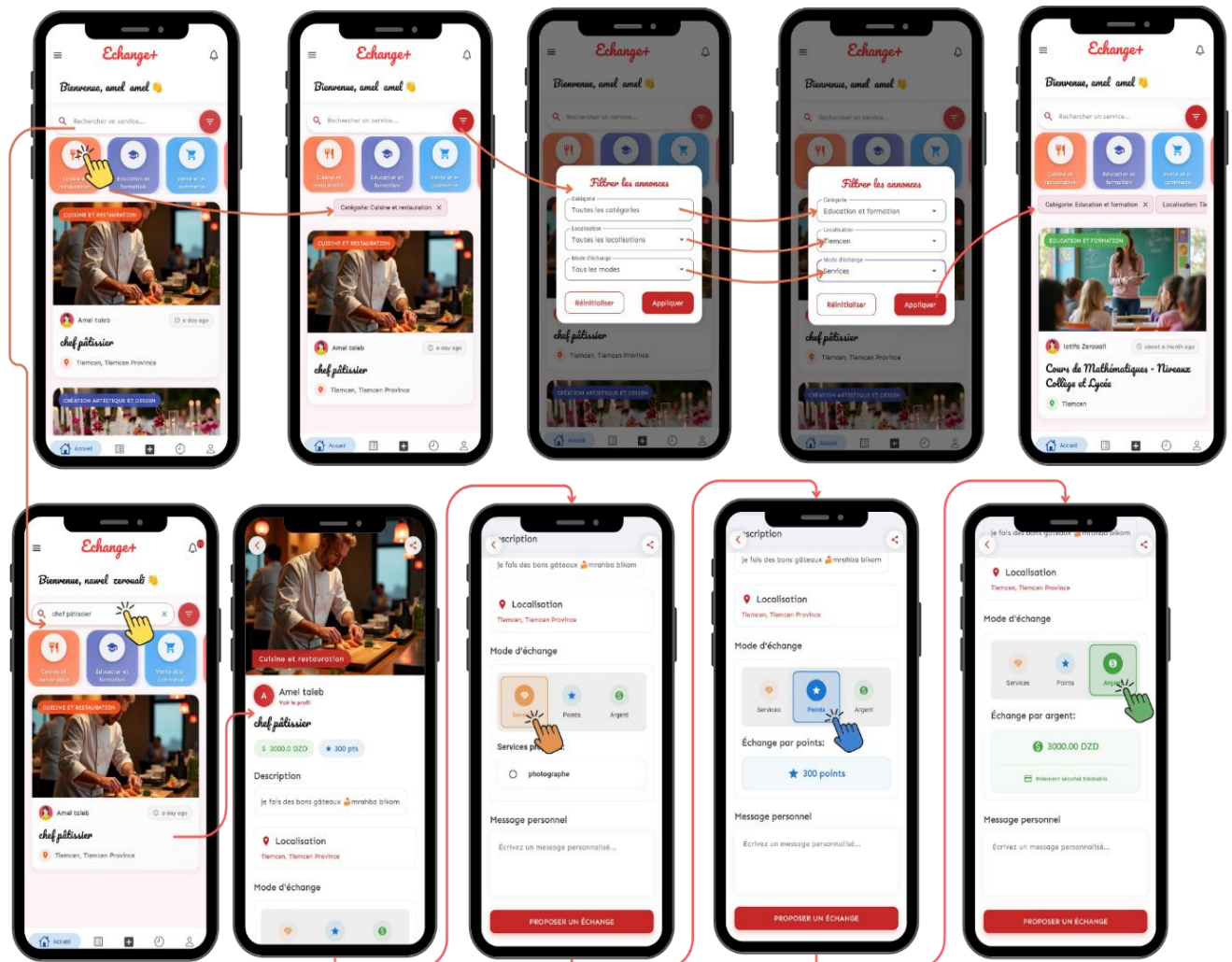


Figure 33: Interfaces de Recherche et proposition d'Échange .

Cette figure 33 présente le **flux utilisateur de recherche, consultation et proposition d'échanges** dans "Echange+". Elle inclut l'**écran d'accueil avec annonces**, la **fonctionnalité de filtrage** (catégorie, type, localisation, prix), l'**écran de détail d'une annonce** (description, modes d'échange), et les **options de proposition d'échange** (services, points, argent), soulignant la flexibilité des interactions.

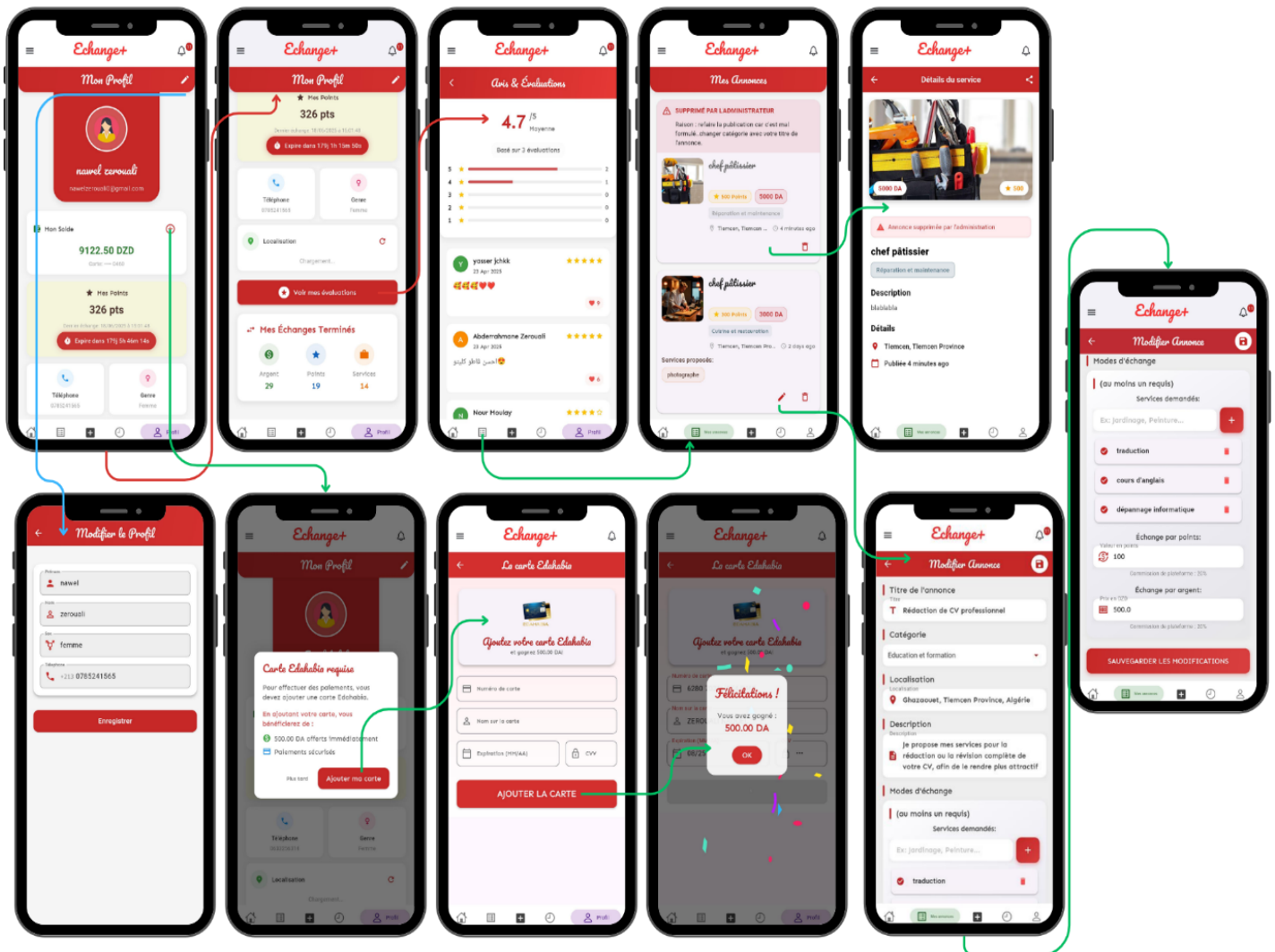


Figure 34: Interfaces de profil utilisateur et gestion des annonces.

Cette figure 34 présente les interfaces de **gestion de profil utilisateur** dans "Echange+", incluant l'affichage des informations, le **système d'évaluation (notes et avis)**, la **consultation et modification des annonces**, la **mise à jour du profil personnel**, et la **gestion d'une carte de récompense**. L'ensemble vise à fluidifier les interactions et renforcer la confiance.

Chapitre III Réalisation et implémentation de l'application « Echange+ »

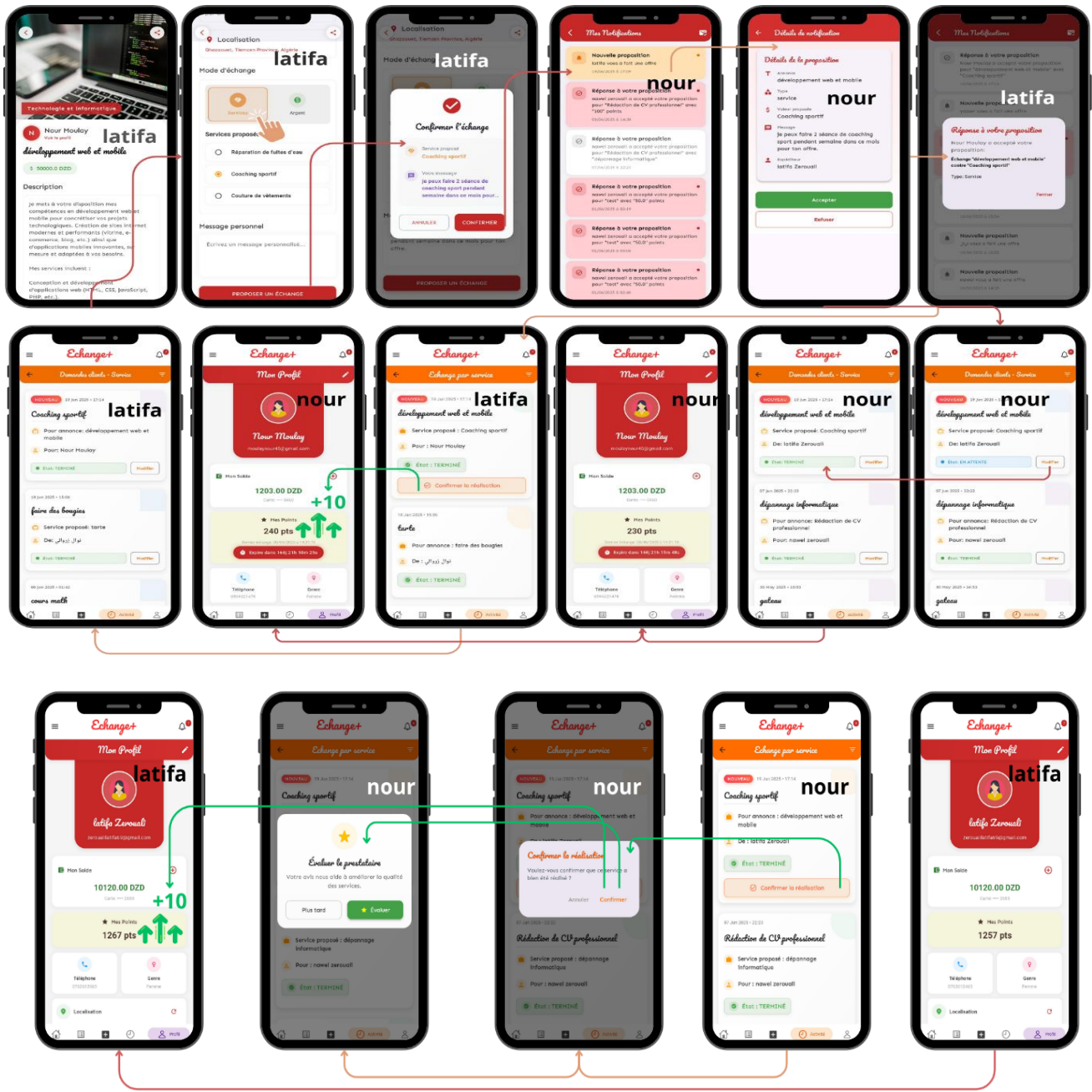


Figure 35: Interfaces d'échange service par service .

Cette figure 35 illustre le cycle complet d'un échange par service sur "Echange+". Elle montre la proposition et confirmation d'échanges via notifications, le suivi des échanges, la mise à jour du profil avec les points gagnés (bonus de 10 points par échange), et le système d'évaluation final, fluidifiant ainsi les interactions utilisateurs.

Chapitre III Réalisation et implémentation de l'application « Echange+ »

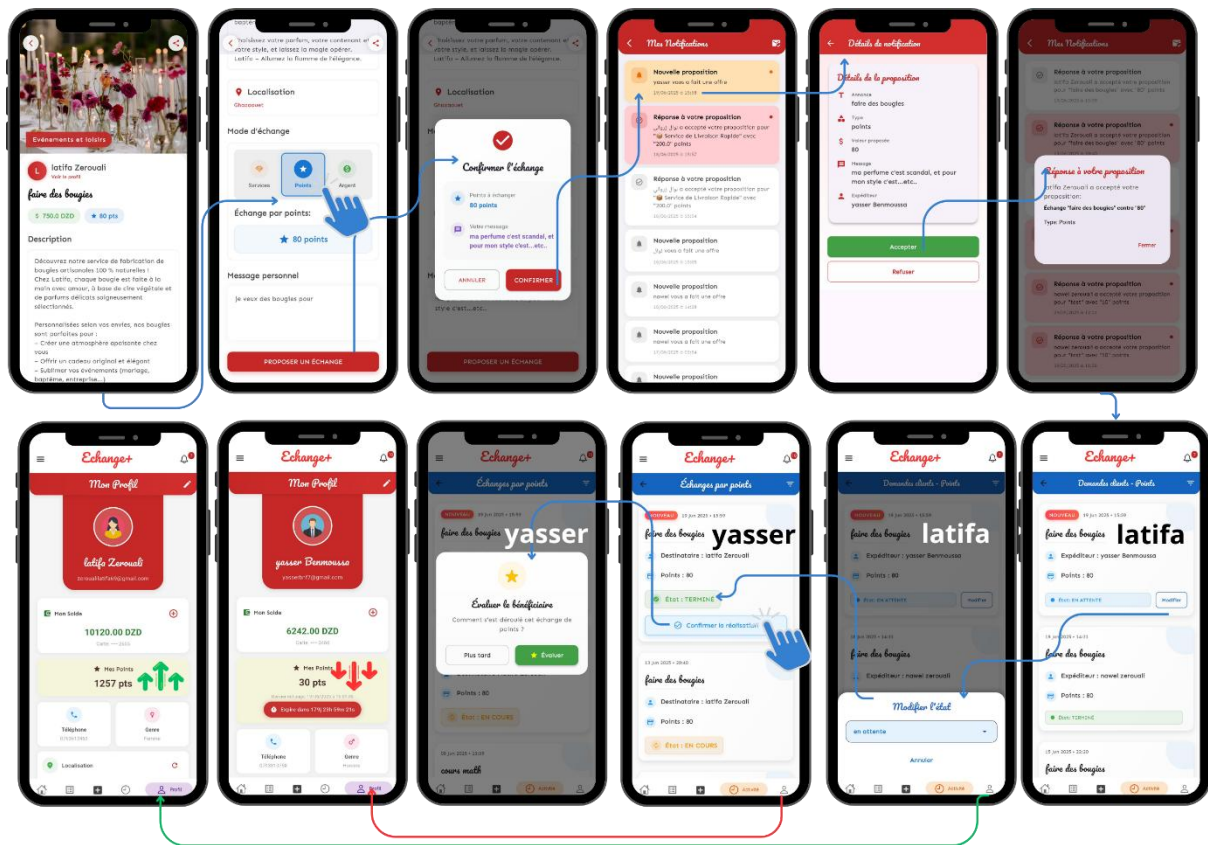


Figure 36: Interfaces d'échange service par points.

Cette figure 36 illustre le flux d'échanges basés sur les points dans "Echange+". Elle couvre la proposition et confirmation d'échanges via notifications, le suivi des transactions, et la gestion des points sur le profil utilisateur, offrant transparence et contrôle sur la plateforme.

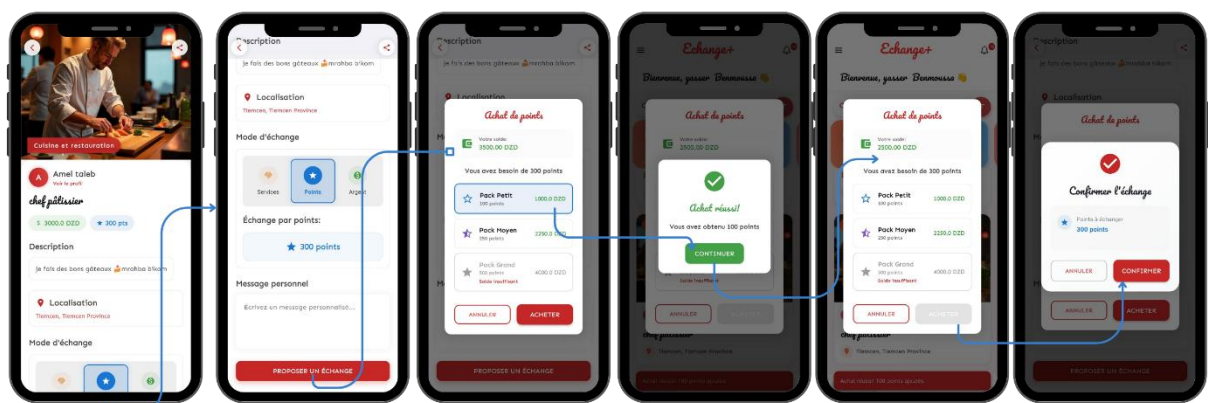


Figure 37: Interfaces d'achat de points par l'argent

Chapitre III Réalisation et implémentation de l'application « Echange+ »

Cette figure 37 montre l'écran d'achat de points dans l'application "Echange+", présentant des packs de points disponibles à l'acquisition avec leur coût associé, permettant aux utilisateurs d'acheter de la monnaie virtuelle pour les échanges.

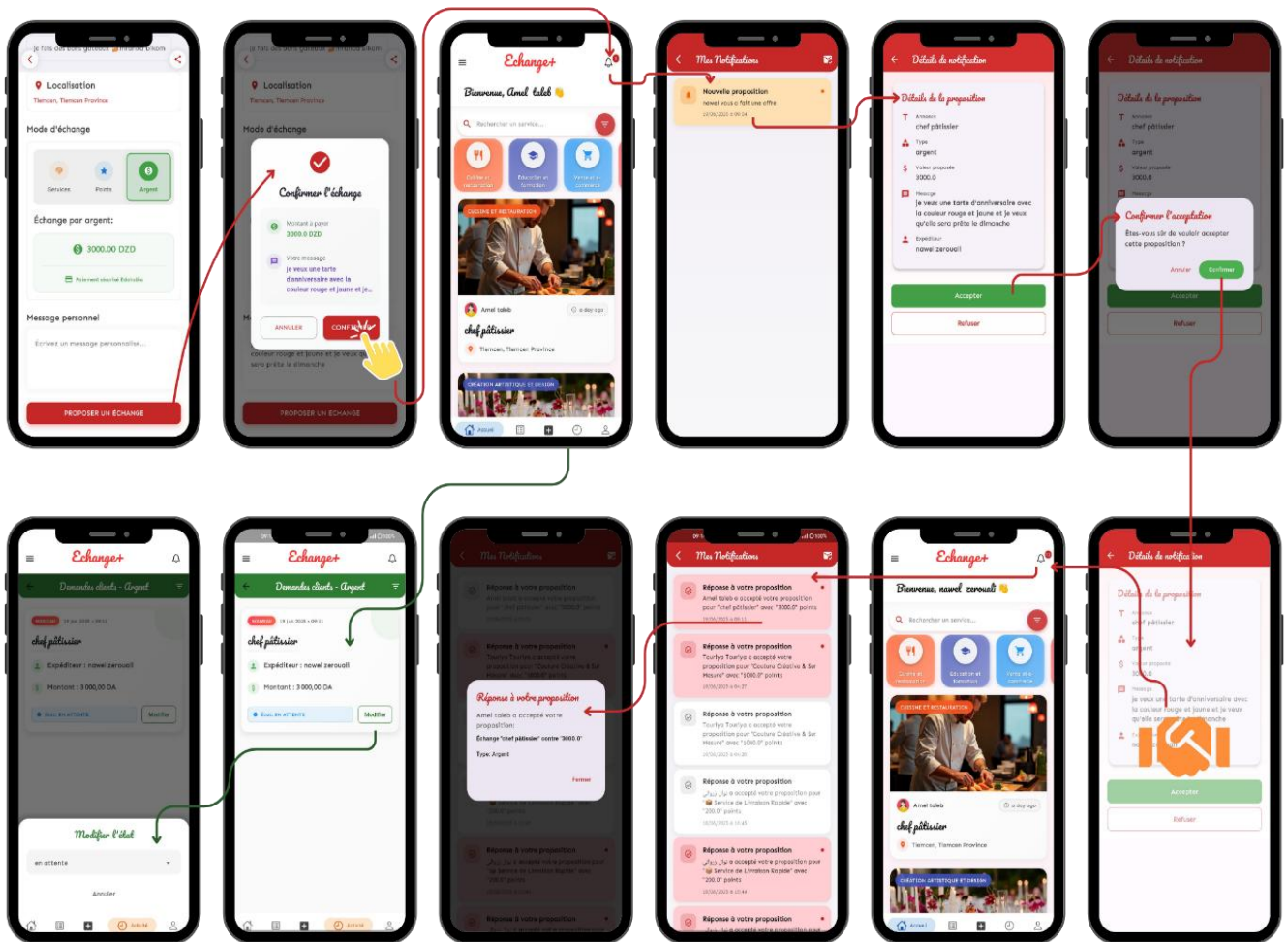


Figure 38: Interfaces de Proposition, Notification et Suivi d'Échanges Monétaires.

Cette figure 38 illustre le flux complet des échanges basés sur l'argent dans l'application "Echange+". Elle détaille la proposition d'échange monétaire pour une annonce, le processus de confirmation, et la gestion des notifications en temps réel (nouvelles propositions, acceptations/refus). Elle montre également le suivi des "Demandes clients" et des échanges, ainsi que l'affichage des mises à jour des soldes sur le profil utilisateur suite aux transactions. L'accent est mis sur la communication et la traçabilité des échanges financiers.

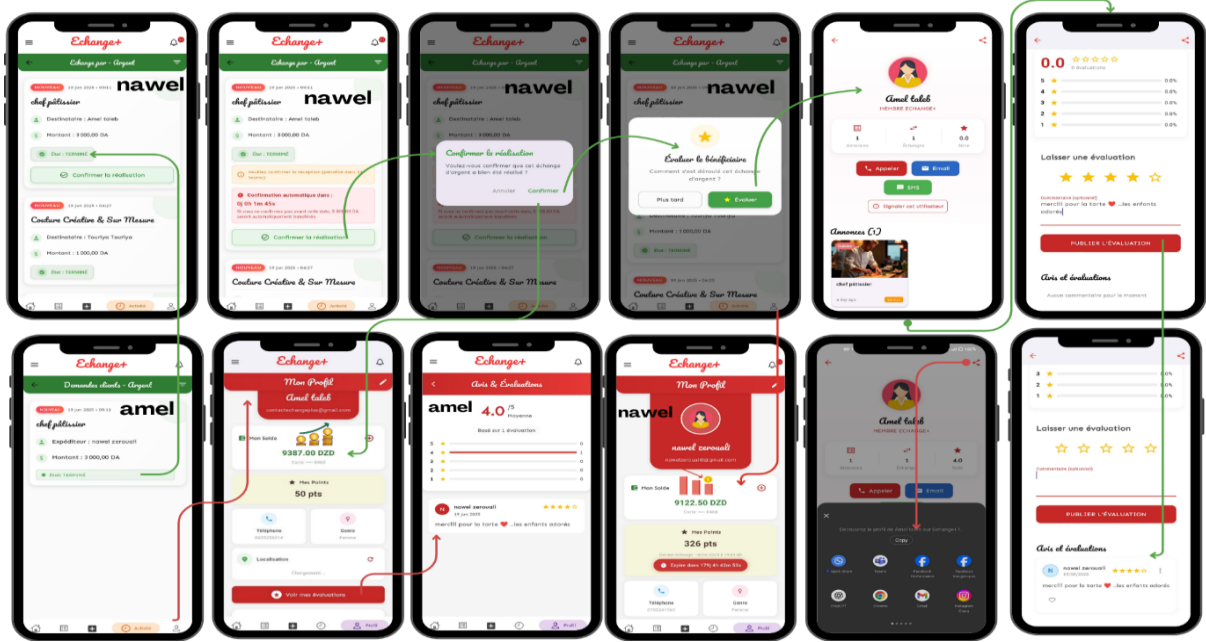


Figure 39: Interfaces de finalisation d'échange service par argent et d'évaluation Utilisateur

Cette figure 39 illustre le processus de **finalisation des échanges** et l'intégration du système d'évaluation dans "Echange+". Elle montre la **confirmation des transactions**, la **validation du bénéficiaire**, la **notation des utilisateurs par étoiles et commentaires**, ainsi que la **mise à jour des profils** avec les nouvelles notes et les avis reçus. Ce flux est essentiel pour bâtir la confiance et la réputation des membres de la plateforme.

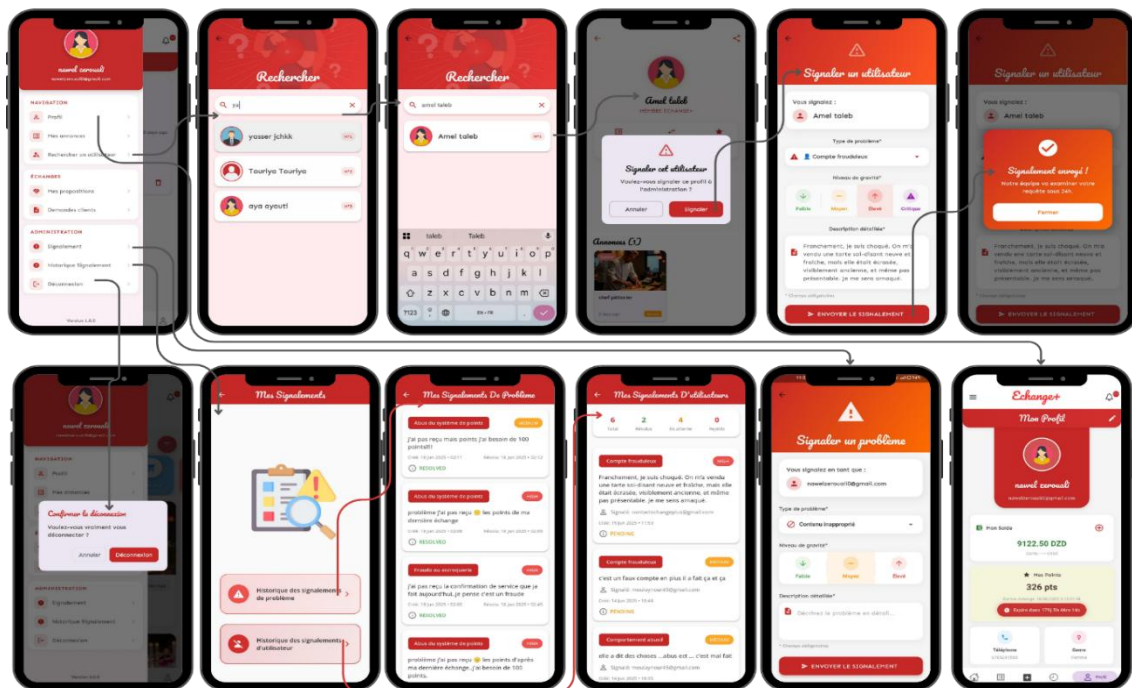


Figure 40: Interfaces de recherche et de signalement utilisateur

Chapitre III Réalisation et implémentation de l'application « Echange+ »

Cette figure 40 présente les fonctionnalités de **recherche d'utilisateurs** et de **signalement** dans "Echange+". Elle illustre le processus de **recherche par nom**, le **signalement d'un utilisateur** (avec choix du type de problème et confirmation d'envoi), et la **gestion des "Mes Signalements"** (consultation de rapports spécifiques et signalement de problèmes généraux). L'ensemble met en lumière la contribution des utilisateurs à la **sécurité et modération** de la plateforme.

III.8.2.3. Coté Admin

Cette section est dédiée à la présentation des **interfaces et fonctionnalités du panneau d'administration de "Echange+"**. Elle détaillera comment les administrateurs peuvent **superviser, gérer et modérer** les utilisateurs, les annonces, les échanges et les signalements pour assurer le bon fonctionnement et la sécurité de la plateforme.

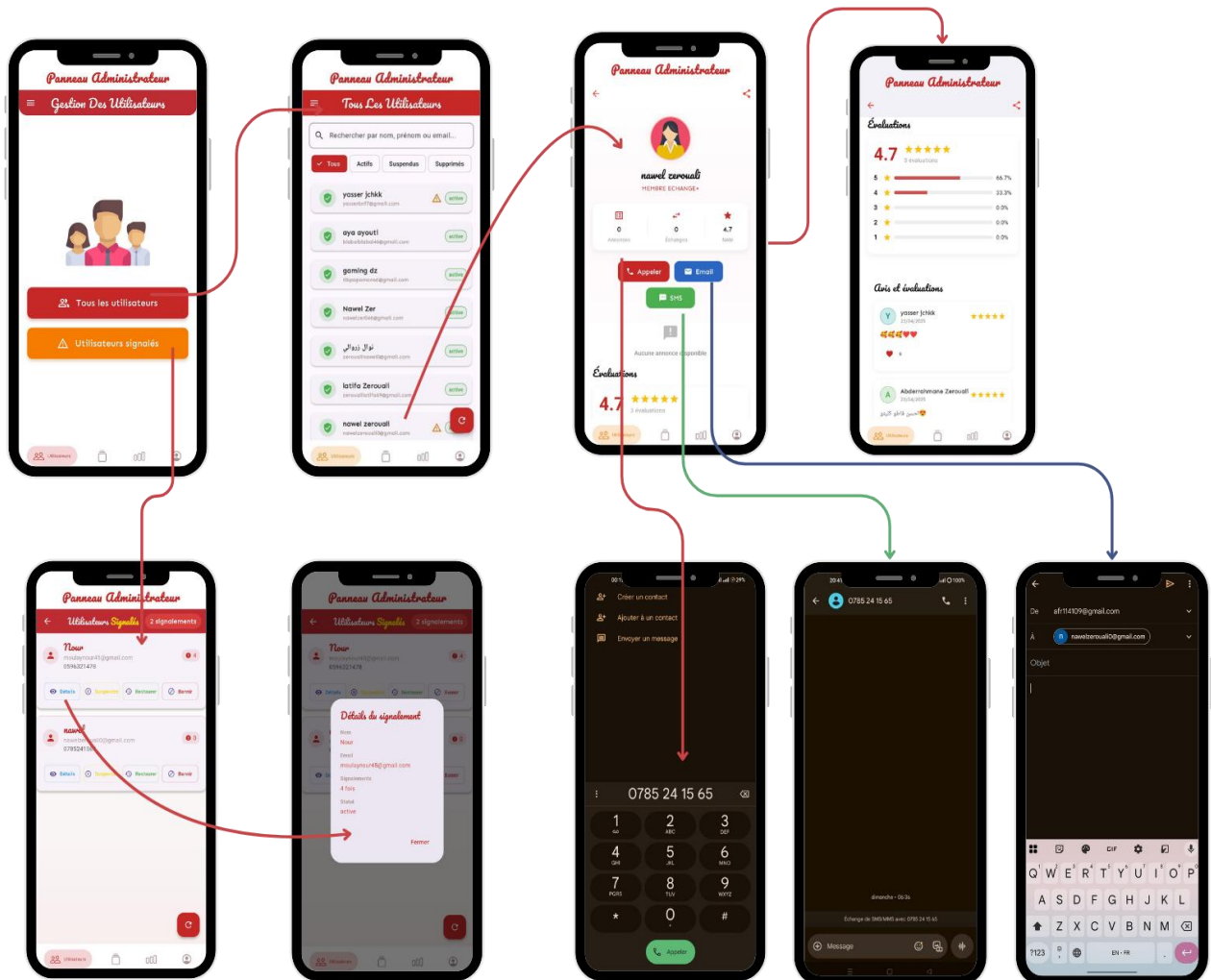


Figure 41: Gestion des utilisateurs et signalements (Panneau Admin)

Chapitre III Réalisation et implémentation de l'application « Echange+ »

Cette figure 41 présente le **Panneau Administrateur "Echange+"**, axé sur la **gestion des utilisateurs et des signalements**. Il permet à l'administrateur de **visualiser et filtrer tous les utilisateurs**, d'accéder à leurs **profils détaillés et évaluations**, et de **traiter les "Utilisateurs signalés"** en consultant les rapports et en interagissant directement. L'objectif est de **superviser et modérer** efficacement la plateforme pour sa sécurité et son bon fonctionnement.

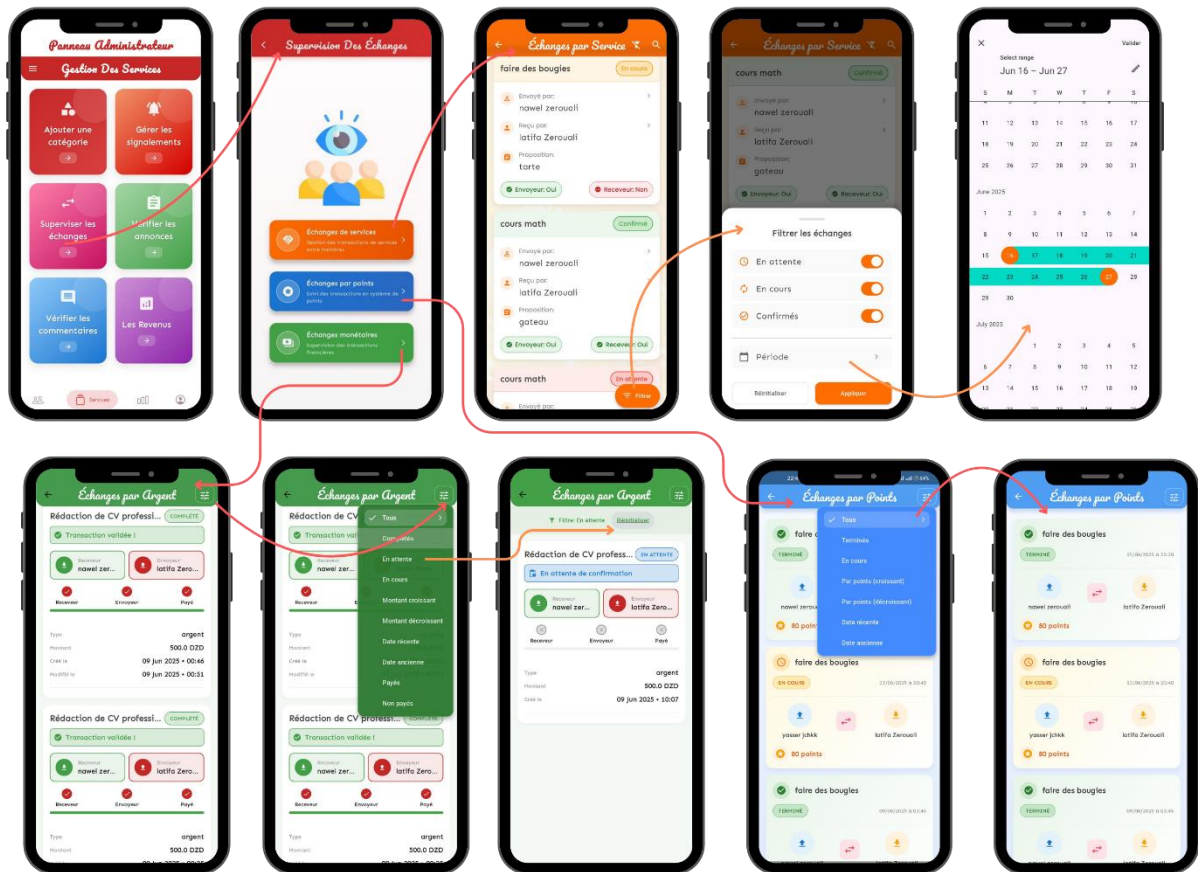


Figure 42: Panneau administrateur - Supervision des échanges et calendrier

Cette figure 42 présente le **panneau administrateur de "Echange+" dédié à la gestion et la supervision des services et échanges**. Elle montre les listes d'échanges (par service, argent, ou points), les options de filtrage par statut, et un calendrier, permettant à l'administrateur de **modérer et suivre l'activité transactionnelle** de la plateforme.

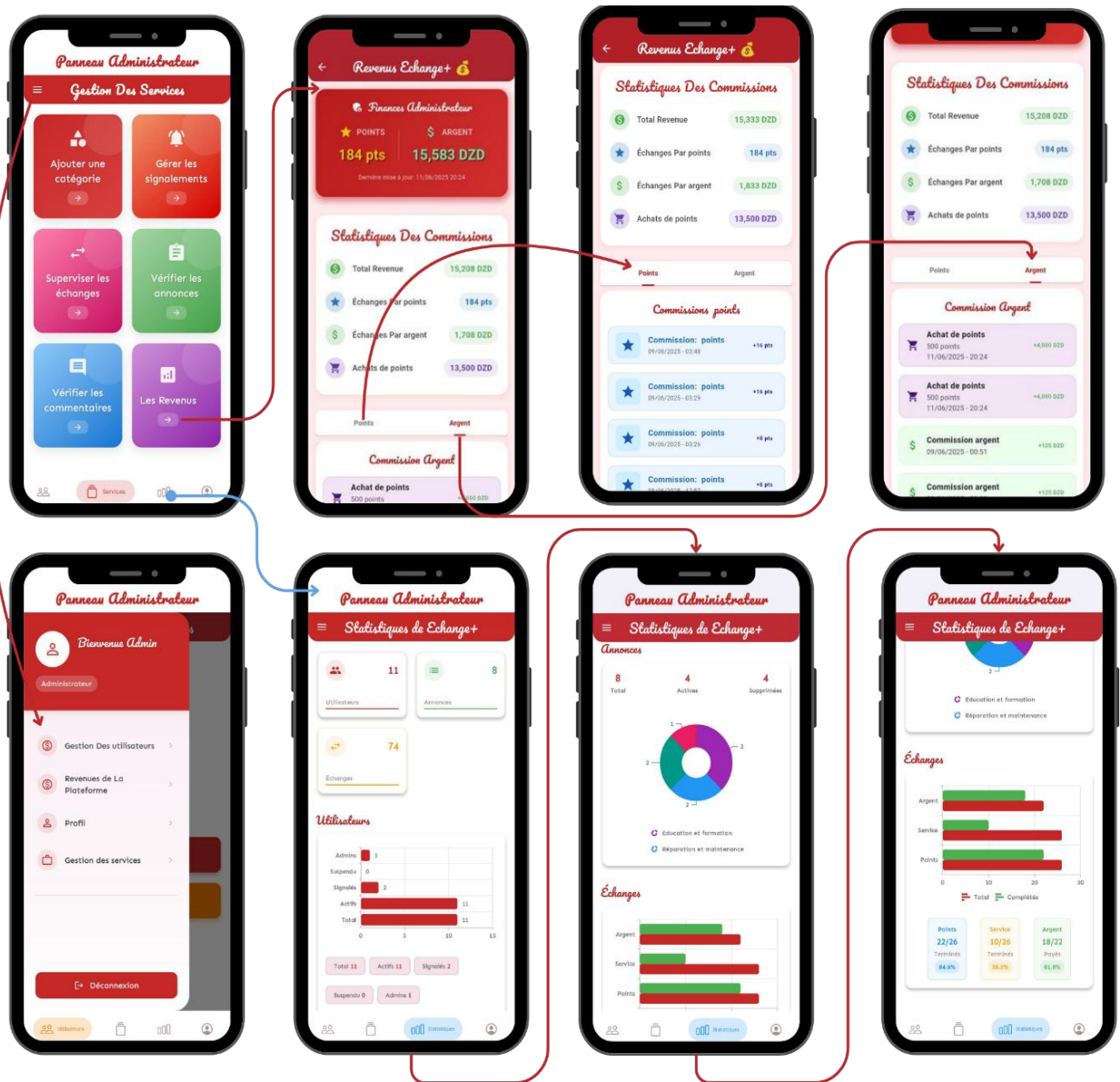


Figure 43: Panneau administrateur - Statistiques, revenus et services

Cette figure 43 présente le **panneau administrateur** de "Echange+" axé sur les **statistiques et les revenus**. Elle affiche les **revenus totaux**, les **commissions (points et argent)**, les **statistiques d'échanges** (utilisateurs, annonces, catégories, graphiques d'échanges), et les **achats de points**, offrant une vue d'ensemble de la performance financière et opérationnelle de l'application.

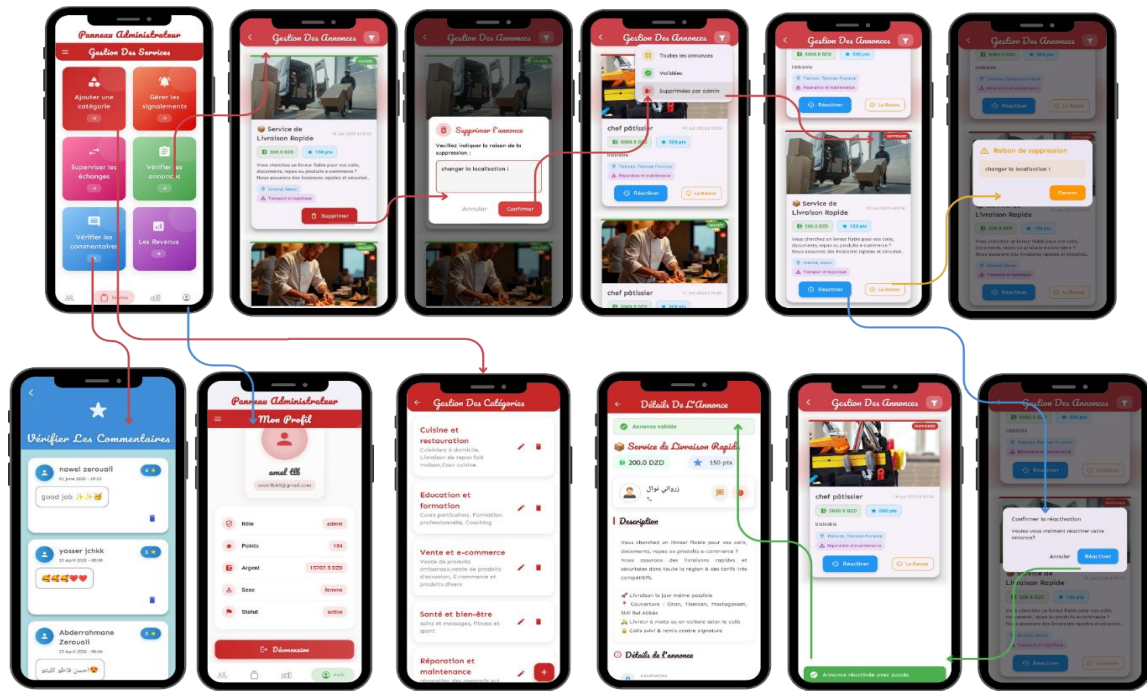


Figure 44: Panneau administrateur - Gestion des annonces et commentaires

Cette figure 44 présente le **panneau administrateur de "Echange+"** pour la gestion des services et des annonces. Elle montre la **modération des commentaires**, la **visualisation et modification des annonces**, ainsi que la **gestion des catégories d'annonces**, permettant à l'administrateur de contrôler le contenu et l'organisation de la plateforme.

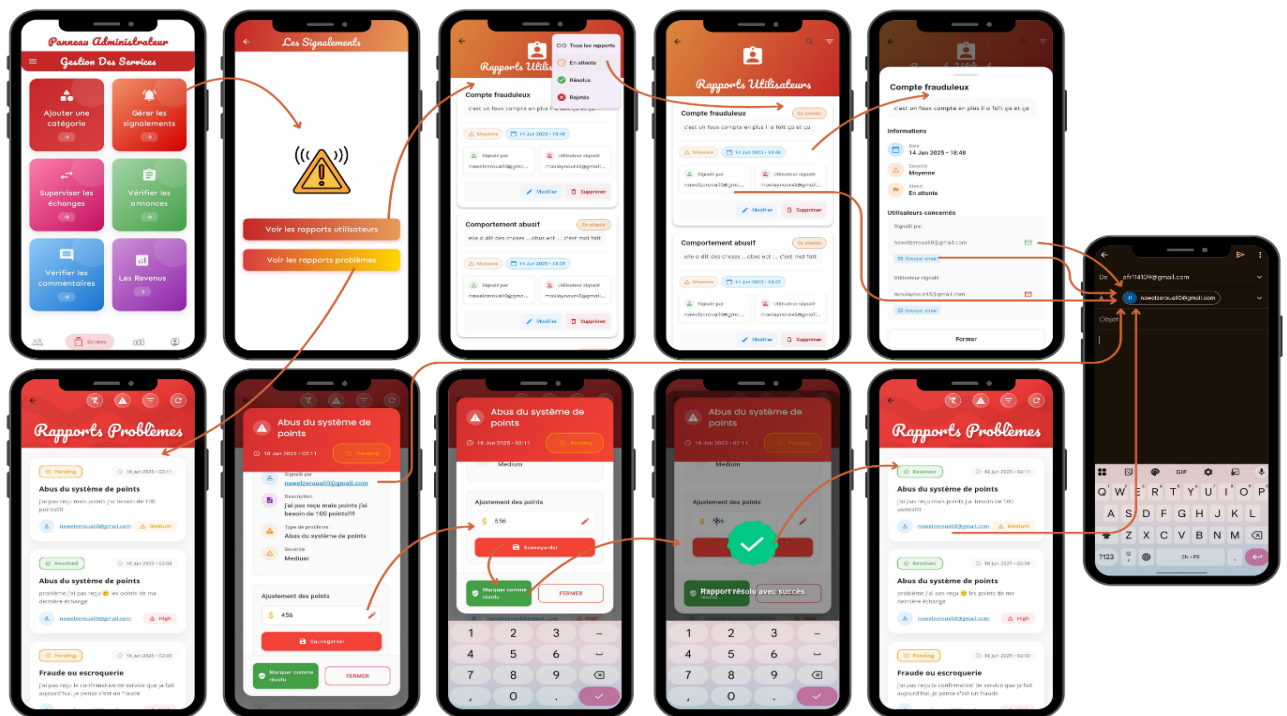


Figure 45: Panneau administrateur - Gestion des signalements et problèmes

Cette figure 45 présente le **panneau administrateur de "Echange+" pour la gestion des services et le traitement des signalements**. Elle montre la **vue des signalements d'utilisateurs et de problèmes (fraude, abus de points)**, incluant les détails des rapports et les actions de résolution, permettant à l'administrateur de maintenir la sécurité et l'équité de la plateforme.

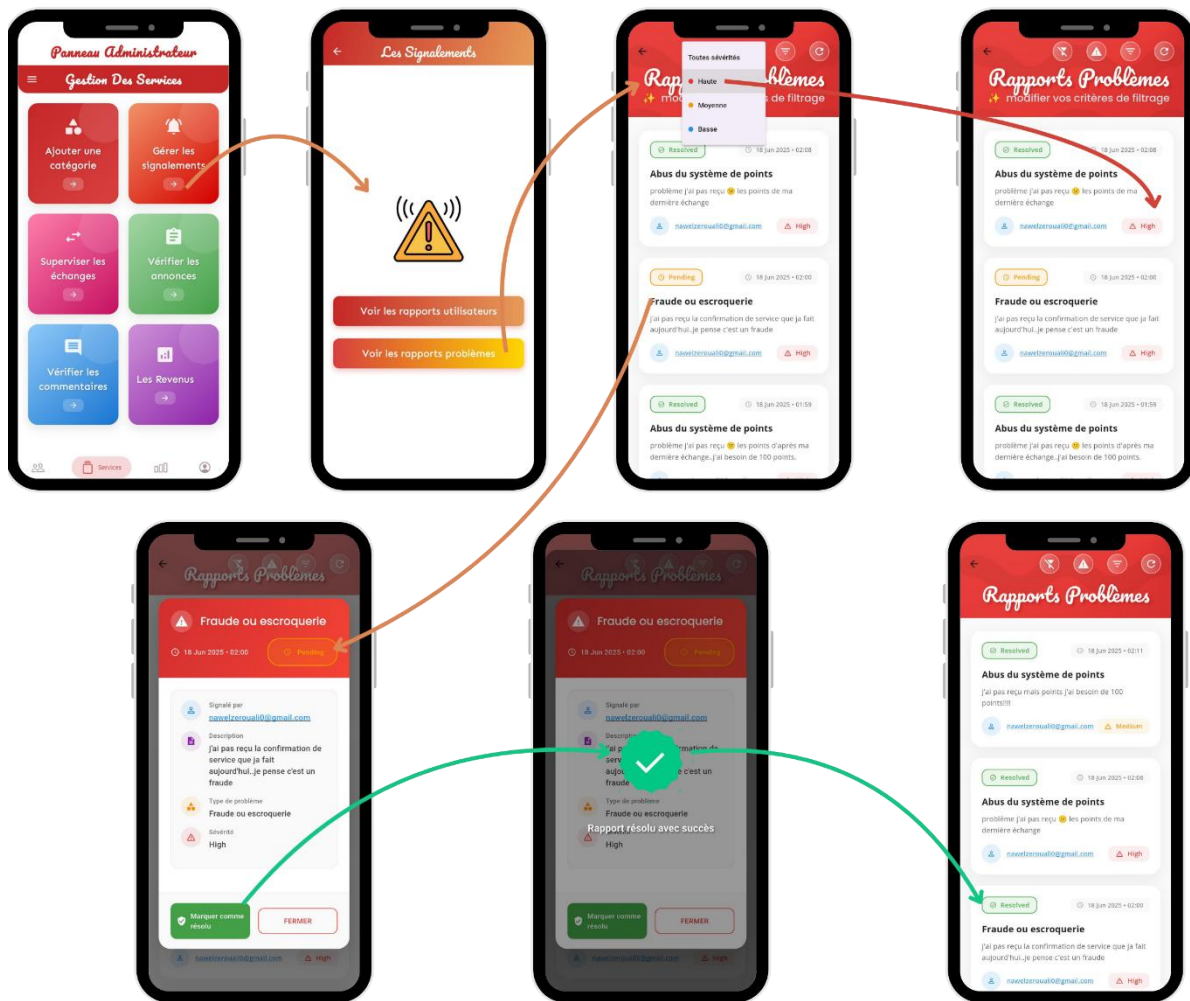


Figure 46: Panneau administrateur - Gestion complète

Cette figure 46 présente le **panneau administrateur de "Echange+" pour la gestion des signalements**. Elle montre la **vue d'ensemble des "Rapports Problèmes" (fraude, abus de points)**, incluant les détails des incidents et les options pour marquer un rapport comme résolu, permettant une gestion efficace des irrégularités sur la plateforme.

III.9. Conclusion

Ce chapitre a retracé l'ensemble des étapes techniques liées au développement de notre application mobile Echange+. En adoptant l'architecture MVVM, nous avons assuré une séparation claire des responsabilités entre les couches de notre système. Grâce à la méthode SCRUM, nous avons planifié notre travail de manière itérative, tout en maintenant une vision structurée du projet. L'utilisation du diagramme de Gantt nous a permis de suivre efficacement l'avancement de chaque tâche. Les outils choisis, tels que Flutter, Firebase et Azure, ont renforcé la robustesse et la modernité de notre solution.

Un travail particulier a été accordé à la conception et à la réalisation des interfaces utilisateur et administrateur, garantissant une expérience fluide, intuitive et visuellement cohérente pour tous les utilisateurs, ainsi qu'une gestion efficace pour les administrateurs.

Nous avons également su faire face à plusieurs défis techniques et organisationnels, que nous avons relevés avec rigueur et créativité. Chaque difficulté rencontrée nous a permis d'approfondir nos compétences et d'améliorer notre approche. Les fonctionnalités développées, **intégrées au sein de ces interfaces soignées**, traduisent notre volonté de proposer une application fluide, utile et bien pensée. Ce chapitre témoigne ainsi de notre implication, de notre capacité d'adaptation et de notre maîtrise des technologies utilisées. Il constitue la preuve concrète du travail accompli de la conception à la mise en œuvre.

Chapitre IV : Business Model Canvas

IV.1. Introduction

Dans ce chapitre, nous allons découvrir l'importance du Business Model Canvas (BMC) pour notre projet. Cet outil nous aide à mieux comprendre, construire et expliquer le modèle économique de notre entreprise. En l'utilisant, nous pourrions vérifier si notre projet est viable, repérer des opportunités de développement et définir des stratégies adaptées.

IV. 2. Les piliers du Business Model Canvas :

IV. 2.1. Propositions de Valeur (Value Propositions)

a. Quels problèmes résolvons-nous pour nos clients ?

ما هي المشاكل التي نحلها لعملائنا ؟

- Les clients rencontrent des difficultés à trouver rapidement des prestataires de services fiables selon leurs besoins et leur localisation.
- Il existe un manque de sécurité et de confiance dans les paiements en ligne.
- La recherche de services sur les réseaux sociaux est mal organisée et prend du temps.
- Certains clients ne peuvent pas toujours payer en argent et manquent d'autres options de paiement
- Le chômage touche une partie des jeunes actifs qui ont des compétences, mais n'ont pas de canal structuré pour proposer leurs services.

b. Quels besoins de nos clients satisfont nos produits ou services ?

ما هي الاحتياجات التي يلبئها منتجاتنا أو خدماتنا لعملائنا؟

- Une plateforme spécialisée qui permet de trouver rapidement des services fiables, avec des offres innovantes, et la possibilité d'échanger ou de payer via plusieurs modes, pas uniquement en argent.
- Système sécurisé et performant avec une gestion de signalement sur certains problèmes surtout les paiements en ligne.
- Une plateforme donne la possibilité d'avoir une vision profonde sur tous les services disponibles dans le marché.
- Un système qui permet aux utilisateurs de suivre en temps réel leurs propositions et demandes grâce aux notifications.
- Une plateforme démocratique qui permet aux utilisateurs de commenter et d'évaluer les prestataires après un échange complet, sous la supervision de l'administrateur.

- Plateforme respect le droit public et adapter avec la mentalité algérienne.
 - Création d'une nouvelle opportunité pour les jeunes de s'épanouir en partageant leurs compétences, tout en s'amusant, gagnant en reconnaissance et en affirmant leur place dans la société et ainsi contribuer à réduire le chômage.
- c. **En quoi notre offre est-elle différente de celle de nos concurrents ?**

في ماذا تختلف عروضنا عن تلك التي يقدمها منافسوننا؟

D'après le tableau comparatif et notre étude, Notre offre se distingue par une combinaison unique de fonctionnalités absentes ou incomplètes chez les concurrents :

- Elle permet l'échange de services sans paiement en argent, contrairement à Fiverr et Youchoz.
 - Elle intègre une gestion de points, permettant d'accumuler des crédits pour accéder à d'autres services.
 - Elle offre un niveau de sécurité élevé grâce au paiement sécurisé, qui garantit la confidentialité des informations personnelles et financières des utilisateurs, ainsi qu'à la géolocalisation, au système de signalement des utilisateurs et des problèmes, et à une gestion fiable des évaluations.
 - Contrairement à Simbi ou Youchoz, Échange+ propose toutes les fonctionnalités clés réunies dans une seule plateforme, incluant aussi une interface d'administration complète.
- d. **Quelles est notre proposition unique de valeur ?**

ما هو العرض الفريد للقيمة لدينا؟

- Nous permettons aux utilisateurs d'échanger des services sans payer en argent, tout en garantissant sécurité, traçabilité et accessibilité, grâce à un système de points et des outils intégrés fiables (géolocalisation, messagerie, évaluation, etc.).

IV. 2.2. Segments de clients (Customer Segments)

a. **Quels sont nos clients principaux ?**

من هم العملاء او الزبائن الرئيسيون؟

- Nos clients principaux sont les personnes ayant un revenu limité, telles que les étudiants, les chômeurs, et les personnes aux compétences sous-exploitées cherchant à proposer leurs services ou à en bénéficier à moindre coût.

b. **Quels sont les différents segments de clients que nous visons ?**

ما هي الفئات المختلفة من العملاء التي تستهدفها؟

Nous visons principalement quatre segments :

- Les personnes en situation financière difficile (personnes pauvres).
- Les chômeurs à la recherche d'une activité ou d'un revenu.
- Les étudiants souhaitant accéder à des services abordables ou proposer leurs compétences.
- Les personnes compétentes qui veulent offrir leurs services, élargir leur visibilité et échanger leurs compétences.

c. Quels sont les besoins spécifiques de chaque segment de clients ?

ما هي الاحتياجات الخاصة لكل فئة من العملاء؟

- Personnes pauvres : accéder à des services utiles à bas prix ou par échange.
- Chômeurs : valoriser leurs compétences, trouver des opportunités et générer un revenu.
- Étudiants : accéder à des services abordables et arrondir leurs fins de mois via des petits services.
- Personnes compétentes : élargir leur réseau de clients, proposer leurs services facilement, et bénéficier d'un système d'échange.

d. Comment pouvons-nous catégoriser nos clients en groupes distincts ?

كيف يمكن تصنيف عملائنا الى مجموعات مختلفة؟

- Nous pouvons les classer selon leur situation économique (faible revenu, sans emploi), leur statut (étudiants, actifs), ou encore leur rôle dans l'application (prestataires de services ou demandeurs de services).

IV. 2.3. Relations avec les clients (Customer Relationships)

a. Quel type de relation chaque segment de clients attend il de nous ?

اي نوع من العلاقة يتوقعه كل فئة من العملاء منا؟

Les clients attendent une relation :

- Basée sur le respect, la liberté d'expression et la confiance, où chacun peut échanger des services en toute simplicité et dignité.
- Encourageante, grâce à des récompenses dès l'inscription (500 DA et 50 points) et à chaque échange de service validé (10 points).
- Participative, avec la possibilité de publier des annonces à tout moment, d'évaluer librement les prestataires et de laisser des commentaires en toute transparence.
- Protectrice, grâce à un système de signalement efficace et à l'intervention rapide de l'admin en cas de problème.

b. Comment entretenons-nous actuellement les relations avec nos clients ?

كيف نحافظ حاليًا على العلاقات مع عملائنا؟

- Nous entretenons une relation solide et interactive en :
- Offrant un bonus d'inscription (argent + points) pour accueillir chaque nouvel utilisateur.
- Récompensant les utilisateurs actifs et fiables via un système de points cumulables.
- Permettant à chacun de donner son avis, commenter et évaluer les services à tout moment, sans restriction.
- Offrant la possibilité de signaler les problèmes ou comportements abusifs, avec un traitement rapide par l'administrateur.
- Garantissant une interface simple, ouverte et respectueuse de chaque membre.

c. Comment pouvons-nous améliorer ou personnaliser nos interactions avec nos clients ?

كيف يمكننا تحسين أو تخصيص تفاعلاتنا مع عملائنا؟

Nous pourrions aller plus loin en :

- Intégrant un profil utilisateur enrichi avec des badges de fiabilité, de réactivité ou d'ancienneté.
- Offrant un espace personnel où l'utilisateur suit ses points, ses évaluations et ses annonces.
- Mettant en place un service d'assistance personnalisé via chat ou message privé.
- Proposant des notifications intelligentes (nouveaux services, suggestions, rappels d'évaluation).
- Organisant des événements communautaires ou des campagnes de bonus pour valoriser l'implication.

IV. 2.4. Canaux (Channels)

a. Par quels canaux nos clients veulent-ils être atteints ?

من خلال أي قنوات يفضل عملاؤنا أن يتم التواصل معهم؟

Nos clients souhaitent être atteints par :

- L'application mobile, accessible à tout moment, pratique et simple d'utilisation.
- Les réseaux sociaux (Facebook, Instagram), pour suivre les nouveautés et recevoir des rappels ou annonces.
- Les notifications mobiles intégrées à l'application, pour être informés en temps réel.

- L'e-mail, pour la confirmation des échanges, les alertes de points ou les messages administratifs.

b. Quels canaux sont les plus efficaces pour atteindre chaque segment de u clients ?

ما هي القنوات الأكثر فعالية للوصول إلى كل فئة من العملاء؟

- Étudiants et jeunes actifs : principalement via l'application mobile et les réseaux sociaux (usage quotidien).
- Chômeurs et personnes en difficulté financière : via l'application mobile et éventuellement par SMS ou e-mail, pour assurer l'accessibilité même en cas de faible connexion.
- Prestataires compétents : via l'application, les notifications ciblées, et les emails personnalisés pour rester informés des opportunités ou demandes.

c. Comment pouvons-nous intégrer différents canaux pour améliorer l'expérience clients ?

كيف يمكننا دمج مختلف القنوات لتحسين تجربة العملاء؟

- Synchroniser les notifications de l'application, les emails et les réseaux sociaux pour que les clients reçoivent les informations essentielles par leur canal préféré.
- Offrir une expérience fluide entre le site web et l'application mobile, avec les mêmes fonctionnalités accessibles.
- Mettre en place un système de messagerie interne centralisée, accessible depuis tous les canaux.
- Utiliser les réseaux sociaux pour informer et interagir, tout en gardant les échanges sécurisés dans l'application.

IV. 2.5. Partenaires clés (Key Partners)

a. Qui sont nos partenaires clés ?

من هم شركاؤنا الرئيسيون؟

Nos partenaires clés sont :

- Des partenaires institutionnels, comme notre université et notre encadrant, qui nous accompagnent dans la conception et la validation du projet.
- Des prestataires de paiement locaux tels que BaridiMob, pour faciliter les transactions sécurisées.
- Des fournisseurs technologiques comme Google, Gmail, Google Cloud, Firebase et Facebook, utilisés pour l'authentification, la géolocalisation, le stockage et la communication.

- Nous-mêmes, en tant qu'équipe projet, jouons un rôle actif dans la conception, le développement et la gestion de la plateforme.
- b. **Quels sont les partenariats qui nous aident à réduire les coûts, à accéder à de nouvelles ressources ou à améliorer notre proposition de valeur ?**

ما هي الشراكات التي تساعدنا على خفض التكاليف أو الوصول إلى موارد جديدة أو تحسين قيمتنا المقترحة؟

- L'appui de notre université et de notre encadrant nous permet de réduire les coûts de lancement tout en assurant la crédibilité académique du projet.
- Les services gratuits ou à faible coût proposés par Google, Firebase et Facebook nous permettent d'intégrer des technologies avancées (géolocalisation, notifications, authentification, base de données) sans frais élevés.
- Les partenariats avec les prestataires de paiement locaux renforcent la confiance des utilisateurs et améliorent l'expérience de paiement en ligne.
- L'implication directe de l'équipe fondatrice assure une maîtrise des ressources humaines et techniques, ce qui contribue à la flexibilité et à l'optimisation des coûts.
- c. **Comment pouvons-nous aligner nos intérêts avec ceux de nos partenaires ?**

كيف يمكننا مزامنة مصالحنا مع تلك لشركائنا؟

- En développant des relations mutuellement bénéfiques, où chaque partenaire tire profit de la visibilité, de la fiabilité et de l'impact social de notre projet.
- En respectant les standards et conditions d'utilisation des technologies partenaires (ex. Google et Firebase) tout en valorisant leurs outils dans notre solution.
- En collaborant avec notre université et encadrant pour assurer un suivi pédagogique, des résultats concrets et une valorisation académique du projet.
- En établissant des accords clairs et transparents avec nos partenaires, basés sur la confiance, l'engagement et les objectifs communs.

IV. 2.6. Activités clés (Key Activities)

- a. **Quelles sont les actions principales que nous devons entreprendre pour livrer notre proposition de valeur ?**

ما هي الأنشطة الرئيسية التي يجب علينا القيام بها لتقديم قيمتنا المقترحة؟

- Permettre aux utilisateurs d'échanger des services selon trois modes : service contre service, service contre points, ou service contre argent.
- Offrir la possibilité de signaler un utilisateur ou un problème, avec un suivi rapide et efficace.

- Intégrer un système d'évaluation et de commentaires pour renforcer la confiance et la transparence.
- Donner la liberté aux utilisateurs de publier une ou plusieurs annonces à tout moment.
- Assurer une interface fluide et sécurisée, favorisant une expérience simple et accessible.

b. Quelles sont les opérations essentielles pour notre entreprise ?

ما هي العمليات الأساسية لشركتنا؟

- Suivi des propositions et des demandes d'échange entre utilisateurs.
- Gestion complète par l'administrateur, incluant :
 - La gestion des utilisateurs, des annonces, des revenus, des statistiques, des catégories, aussi la supervision des échanges, le traitement des signalements.
 - Mise à jour continue de la plateforme, pour garantir performance, sécurité et ajout de nouvelles fonctionnalités.
 - Support technique et assistance aux utilisateurs pour résoudre rapidement les problèmes rencontrés.

c. Quelles sont les activités qui créent le plus de valeur pour nos clients ?

ما هي الأنشطة التي تخلق أكبر قيمة لعملائنا؟

- La possibilité d'échanger des services sans argent, grâce à un système de points souple et accessible.
- L'inscription avec bonus (argent + points), qui motive et récompense dès le départ.
- L'expérience utilisateur simple, libre et sécurisée, avec géolocalisation, messagerie, signalement et évaluations.
- La liberté de poster des annonces à tout moment, et de commenter ou évaluer les autres utilisateurs librement.
- La réactivité de l'administration en cas de problème ou de litige.

IV. 2.7. Ressources clés (Key Resources)

a. Quels sont nos actifs matériels, immatériels et humains essentiels ?

ما هي الأصول المادية وغير المادية والبشرية الأساسية لدينا؟

- Ressources humaines : notre propre équipe projet, qui assure la conception, le développement, la gestion et le suivi de la plateforme.
- Ressources immatérielles : l'idée innovante de notre projet, notre savoir-faire, la maquette de l'application, les mécanismes d'échange (points, argent, service) et la structure du système d'administration.

- Ressource académique : l'accompagnement pédagogique et méthodologique de notre encadrante, Mme Benmansour Fassila Lamiya, qui nous guide tout au long du projet.
 - Nous ne disposons d'aucune ressource matérielle ou financière externe : tout est développé à partir de nos propres moyens personnels (ordinateurs, connexion Internet, logiciels libres ou gratuits).
- b. Quels sont les outils, les technologies ou les partenariats dont nous avons besoin pour réussir ?**

ما هي الأدوات والتكنولوجيا أو الشراكات التي نحتاجها لتحقيق النجاح؟

- Firebase : pour la base de données, l'authentification, les notifications et l'hébergement.
 - Google Cloud : pour le stockage, la performance et l'intégration avec d'autres services.
 - Google, Gmail et Facebook : pour l'authentification et l'intégration des comptes utilisateurs.
 - Outils de développement : Android Studio, Figma, GitHub...
 - Encadrement académique : pour orienter notre réflexion, structurer notre solution, valider les choix fonctionnels.
- c. Quels sont les principaux avantages concurrentiels de nos ressources ?**

ما هي المزايا التنافسية الرئيسية لمواردنا؟

- Une idée adaptée au contexte local (algérien) avec une vraie réponse aux besoins sociaux (échange sans argent).
- Une équipe motivée, autonome et polyvalente, capable de concevoir une solution complète sans ressources financières extérieures.
- L'usage de technologies modernes, puissantes et gratuites (Firebase, Google Cloud), qui nous permet de concurrencer des plateformes bien financées.
- Un accompagnement pédagogique de qualité, qui renforce la crédibilité et la rigueur du projet.

IV. 2.8. Charges et coûts (Cost Structure)

- a. Quels sont les coûts fixes et variables associés à notre modèle économique ?**

ما هي التكاليف الثابتة والمتغيرة المرتبطة بنموذجنا الاقتصادي؟

Dans notre cas, les coûts sont surtout liés à nos ressources personnelles :

- Coûts fixes : électricité, consommation internet, matériel personnel (ordinateurs), et environnement de travail.
- Coûts variables : temps passé à apprendre Flutter, Dart et les outils nécessaires depuis zéro, efforts de développement, erreurs à corriger, et tests à répéter.

- Nous n'avons pas pu héberger l'application, ni utiliser certaines fonctionnalités (comme le stockage de photos via Firebase) en raison des coûts de services externes.

b. Quels sont les coûts les plus importants pour notre entreprise ?

ما هي التكاليف الأكثر أهمية لشركتنا؟

- L'hébergement de l'application (Firebase Storage, bases de données) : les solutions gratuites sont limitées, et les versions payantes sont trop coûteuses pour nous.
- Intégration de services de paiement comme BaridiMob : inaccessible pour nous faute de registre de commerce, sachant que la gestion d'un tel statut nécessite environ 30 millions de centimes par mois, un coût impossible à couvrir en tant qu'étudiants.
- L'investissement personnel : nous avons tout appris et construit par nous-mêmes, à deux, sans encadrement technique, avec de longues heures de travail, de frustrations mais aussi de résilience.

c. Comment pouvons-nous réduire les coûts ou améliorer l'efficacité de nos opérations ?

كيف يمكننا خفض التكاليف أو تحسين كفاءة عملياتنا؟

- En utilisant des outils gratuits ou en version limitée comme Firebase, Google Cloud, GitHub, et d'autres technologies accessibles aux étudiants.
- En ayant tenté d'intégrer Azure Blob Storage via un package GitHub, mais l'accès étant limité par une clé privée, cela n'a pas été possible. Nous avons aussi testé Cloudinary, mais ses restrictions sur le nombre de fichiers et la capacité nous ont poussés à chercher une autre solution.
- Pour contourner ces limites, nous avons choisi d'utiliser des images statiques :
- Des avatars de profil génériques selon le genre de l'utilisateur.
- Des images fixes représentant les catégories d'annonces, sélectionnées automatiquement.
- Concernant le paiement en ligne (BaridiMob), son intégration nécessitant un registre de commerce et des frais élevés, nous avons opté pour une simulation réaliste du paiement dans le prototype, afin d'illustrer le fonctionnement prévu.
- Nous avons ainsi pu livrer un prototype fonctionnel, centré sur les fonctionnalités essentielles, grâce à une bonne organisation, de la créativité et une forte autonomie.

IV. 2.9. Revenus (Revenue)

a. Quels produits ou services nos clients sont-ils prêts à payer ?

ما هي المنتجات أو الخدمات التي يكون عملاؤنا على استعداد لدفع ثمنها؟

Actuellement, notre plateforme fonctionne sans véritable paiement en ligne. Toutefois, les utilisateurs :

- Peuvent acheter des points (en théorie), qui leur permettent d'obtenir des services via l'échange par points.
- Peuvent effectuer un échange par argent de manière simulée, car l'intégration d'un système de paiement réel (ex. BaridiMob) n'est pas encore en place.
- Acceptent, dès l'inscription, les conditions générales qui expliquent le mode de fonctionnement des revenus, les commissions, et les règles de sécurité.

À l'avenir, nous proposerons une formule d'abonnement premium, qui déblocuera des fonctionnalités illimitées après un certain niveau d'activité (par exemple, après 5 échanges).

b. Quels sont les différents moyens par lesquels nous pouvons générer des revenus ?

ما هي الطرق المختلفة التي يمكننا من خلالها تحقيق الدخل؟

- ★ Actuellement (en mode freemium et simulation)

Échange service par service :

-Actuellement gratuit.

Échange par points :

-Les utilisateurs gagnent 10 points à chaque échange validé.

-Ils peuvent acheter des points (fonction simulée pour l'instant).

-La plateforme prélève une commission de 20 % sur la valeur du service en points.

Échange par argent (simulation) :

-Le paiement est non intégré techniquement, mais le processus est représenté dans le prototype.

-Une commission de 25 % est prévue sur le montant payé au prestataire (en théorie).

Expiration des points :

Les points non utilisés 6 mois après le dernier échange expirent automatiquement.

Mesures disciplinaires :

En cas d'abus ou de comportement dangereux, un utilisateur peut être suspendu ou supprimé, ce qui entraîne la perte de ses points et gains simulés.

- ★ Futur (mode premium)

Abonnement obligatoire après un certain seuil d'usage (par exemple, après 5 échanges).

Tarifs envisagés :

- 1 mois : 3000 DA
- 3 mois : 8000 DA
- 1 an : 30 000 DA

Dans ce modèle, les commissions sur les échanges seront supprimées pour les abonnés, qui bénéficieront de services illimités et de fonctionnalités avancées.

c. Quel est notre modèle de tarification ?

ما هو نموذج التسعير لدينا؟

Notre modèle repose sur :

- Un accès gratuit de base avec bonus d'inscription (points + argent simulé).
- Un système d'échange entre utilisateurs (par service, points ou argent).
- Des commissions simulées (20 % sur les échanges en points, 25 % sur les paiements en argent).
- Une future offre premium sur abonnement qui remplacera les commissions et offrira des avantages exclusifs.
- Un cadre réglementaire clair, via les conditions générales d'utilisation acceptées dès l'inscription.

IV. 3. Business Model Canvas :

Partenaires clés	Activités clés	Proposition de valeur	Relation avec les clients	Segments de clients
<ul style="list-style-type: none"> Université et encadrant. Google, Firebase, Facebook (technologies). I2E. 	<ul style="list-style-type: none"> Gestion des échanges (service, points, argent) Système d'évaluation et signalement Interface sécurisée et simple Supervision et administration complète 	<ul style="list-style-type: none"> Échange de services avec ou sans argent. Système de points pour faciliter les échanges. Plateforme sécurisée avec modération. Géolocalisation des prestataires. Évaluations et avis entre utilisateurs. Plusieurs modes de paiement disponibles. Interface tout-en-un, simple et efficace. 	<ul style="list-style-type: none"> Respect, confiance, liberté d'expression. Essai gratuit. Récompenses à l'inscription et après échanges validés Signalement rapide des problèmes, modération active (commentaire, annonce). Interactions possibles via annonces, évaluations, commentaires... 	<ul style="list-style-type: none"> Personnes à faibles revenus (étudiants, chômeurs). Prestataires compétents cherchant visibilité et opportunités. Besoins : Services abordables, valorisation des compétences, accès facile aux échanges.
Ressources clés <ul style="list-style-type: none"> Infrastructure Cloud (Firebase, Google Cloud) authentification avec Facebook et Gmail. Équipe projet (développeurs, designers, gestionnaires) = nous-mêmes. Outils de géolocalisation et notifications Partenariats (à venir) : BaridiMob, université, I2E. 			Canaux de distribution <ul style="list-style-type: none"> Réseaux sociaux (Facebook, Instagram). Notifications intégrées et e-mails. SMS pour accès bas débit. vidéo publicitaire (scénario écrit). les panneaux publicitaires. 	
Charges et coûts <ul style="list-style-type: none"> Coûts fixes mensuels : <ul style="list-style-type: none"> Un développeur par contrat (occasionnelle) (freelancer) pour l'hébergement de l'application mobile : <ul style="list-style-type: none"> Google Play: 3400 DA/mois App store : 14000 DA /mois Sous-total fixes annuels = 208 800 DA/an Coûts variables mensuels : <ul style="list-style-type: none"> Services cloud (Firebase, stockage (plan blaze)) : -13500 DA/année. pour (à venir) : baridi mob (Comfort): 300 transactions/mois sa valeur : moyenne 1000 da chacune et total mensuel payé par utilisateurs (300*1000 DA = 300 000 DA), et leur commission (1,25% c'est 3750 DA/Mois). 3 750 DA/mois × 12 = 45 000 DA/an google Maps api : <ul style="list-style-type: none"> → 0 DA tant que tu restes sous 14 000 utilisateurs actifs par mois (quota gratuit de 200 \$ non dépassé). → Coût variable uniquement si dépassement (pay as you go). vidéo publicitaire de coût : 30 000 DA.(payé par nous pour le moment). publier par infulceur : 50 000 DA. <p>1. Sous-total variables annuels = 13 500 DA + 45 000 DA + 50 000 DA = 108 500 DA/an Total estimé : -208 800 DA (fixes) + 108 500 DA (variables) = 317 300 DA/an</p>			Revenus <ul style="list-style-type: none"> Échange par points : <ul style="list-style-type: none"> Achat : 100 points = 1000 DA / 250 points =2250 DA / 500 points = 4000 DA. On prend une commission de 20 % sur chaque échange en points. Comme ça, Encourage l'achat régulier de points, car les utilisateurs perdent un peu à chaque échange. Échange par argent (simulation) : <ul style="list-style-type: none"> Commission : 25 % sur chaque transaction monétaire (non intégrée actuellement) Expiration des points : <ul style="list-style-type: none"> Points inactifs depuis 6 mois sont supprimés ⇒ incite à l'achat régulier Abonnement Premium (à venir) :pression des commissions pour abonnés, accès illimité aux fonctionnalités <ul style="list-style-type: none"> 1 mois : 3 000 DA / 3 mois : 8 000 DA / 1 an : 30 000 DA. Estimation de revenus mensuels (à maturité) : <ul style="list-style-type: none"> Revenus points : 332 500 DA (si on vents : 100 paquets de 100 points, 50 paquets de 250 points ,30 paquets de 400 points). Commission échanges argent (25%) : 50 000 DA (exemple :100 transactions × 2 000 DA = 200 000 DA/mois ,donc : 25 % × 200 000 DA = 50 000 DA/mois). Abonnements Premium : 150 000 DA (50 abonnés mensuels). <p>→ Total estimé mensuel = 332 500 + 50 000 + 150 000 = 532 500 DA/mois → Total estimé : -532 500 DA/mois × 12 = 6 390 000 DA/an.</p>	

Figure 47: Le modèle économique et sociale de l'application "Echange+"

IV. 4. Conclusion :

Pour conclure ce chapitre sur l'utilisation du Business Model Canvas (BMC) dans notre projet, nous comprenons désormais l'importance de cet outil pour construire et analyser notre modèle économique. En détaillant les principaux éléments de notre application, le BMC nous a aidés à mieux comprendre notre fonctionnement et notre stratégie.

Conclusion générale

Conclusion Générale

Au terme de ce projet de fin d'études, nous avons eu l'opportunité de concevoir, développer et finaliser une application mobile innovante intitulée Echange+, qui vise à transformer la manière dont les utilisateurs interagissent pour échanger des services et des compétences. Ce projet s'inscrit dans un contexte sociétal où l'entraide, l'optimisation des ressources prennent une place croissante dans les modes de vie contemporains.

Echange+ repose sur un concept simple mais ambitieux : faciliter les échanges locaux via une plateforme numérique accessible, conviviale et sécurisée. Trois modes de transaction sont proposés pour s'adapter aux besoins et aux possibilités de chacun : l'échange direct sans contrepartie financière, l'utilisation de points convertibles, et le paiement électronique via des solutions locales telles que BaridiMob. Cette diversité d'options permet d'englober plusieurs profils d'utilisateurs et de promouvoir à la fois la solidarité et l'inclusion économique.

Le développement de l'application nous a amenés à combiner plusieurs compétences techniques, notamment en programmation mobile (Flutter), en architecture logicielle, en intégration de bases de données cloud (Firebase), et en déploiement de services tiers (authentification, notifications push, géolocalisation). L'ensemble du cycle de développement a été réalisé dans le respect des bonnes pratiques de conception logicielle : analyse des besoins, modélisation UML, définition d'une architecture scalable, puis mise en œuvre agile, avec des itérations successives permettant d'affiner et de stabiliser le produit.

Sur le plan fonctionnel, Echange+ propose un ensemble d'outils destinés à optimiser l'expérience utilisateur : publication rapide d'annonces, moteur de recherche avancé, cartographie des offres, notifications ciblées, et système d'évaluation communautaire pour garantir la fiabilité des échanges. L'ergonomie de l'interface a été pensée pour être intuitive, en vue de permettre à des utilisateurs de tous âges et de tous niveaux techniques d'accéder facilement aux fonctionnalités de l'application.

Au-delà de ses aspects techniques, Echange+ présente un potentiel économique et social majeur. L'application ouvre des perspectives de monétisation via des services premium (mise en avant des annonces, abonnements spéciaux), la création de partenariats avec des entreprises

Conclusion générale

locales souhaitant valoriser leurs produits ou services auprès d'une communauté active et engagée. Cette dynamique économique est soutenue par une approche responsable, visant à maintenir l'équilibre entre rentabilité et utilité sociale.

Les perspectives d'évolution de Echange+ sont nombreuses et ambitieuses :

- Intégration d'un système de paiement en ligne hautement sécurisé, pour simplifier les transactions financières et renforcer la confiance entre les utilisateurs.
- Mise en place d'un mécanisme de fidélité intelligent : chaque interaction, contribution ou transaction pourrait générer des points de fidélité, échangeables contre des avantages exclusifs ou des réductions, renforçant ainsi l'engagement et la rétention des utilisateurs.
- Segmentation plus fine des catégories de services : notamment des espaces dédiés aux étudiants (échange de livres, de matériel pédagogique), aux artisans, aux associations caritatives et aux jeunes entrepreneurs.
- Ajout d'un système d'assistance intelligente basé sur l'IA pour aider les utilisateurs à trouver plus rapidement ce qu'ils cherchent, proposer des offres pertinentes et optimiser la mise en relation.
- Intégration d'un système de messagerie instantanée permettant la communication entre les utilisateurs eux-mêmes ainsi qu'avec les administrateurs.

D'un point de vue stratégique, Echange+ s'inscrit pleinement dans les dynamiques actuelles de transformation numérique en Algérie. Le projet accompagne la montée en puissance de l'économie collaborative, la valorisation du digital dans la vie quotidienne, et la démocratisation de l'accès aux services numériques, y compris dans des régions éloignées ou défavorisées.

Sur le plan personnel et académique, la réalisation de ce projet a été particulièrement formatrice. Nous avons appris à allier rigueur technique, vision stratégique et écoute des besoins utilisateurs, compétences essentielles pour aborder avec confiance nos futures missions professionnelles. Nous avons également pris conscience des défis réels liés au déploiement d'une application mobile : sécurité des données, expérience utilisateur, évolutivité, compatibilité avec différentes plateformes et contraintes locales spécifiques.

Conclusion générale

En définitive, Echange+ ne se limite pas à une simple application : c'est une réponse innovante à des problématiques sociales concrètes, une contribution à l'économie locale et un levier pour encourager de nouvelles formes de solidarité numérique. Nous espérons que ce projet pourra inspirer d'autres initiatives du même type et ouvrir la voie à une culture d'échange, de collaboration et de confiance entre citoyens algériens.

Nous sommes fiers de ce que nous avons accompli au travers de ce projet et confiants que Echange+ continuera à évoluer pour mieux répondre aux attentes d'une communauté toujours plus connectée, plus solidaire et plus innovante.

Avec Echange+ : Échangez plus, gagnez plus !

Bibliographie

Bibliographie

- [1] « Définition d'une application mobile », *TechTarget - WhatIs.com*. Disponible sur : <https://www.techtarget.com/whatis/definition/mobile-app> .Consulté le : 07 avril 2025.
- [2] « Application mobile d'entreprise : 20 fonctionnalités essentielles ». Disponible sur : [LumApps, Application mobile d'entreprise : 20 fonctionnalités essentielles](#) . Consulté le : 08 avril 2025.
- [3] « 5 Caractéristiques d'une Application réussie », Bluumi. Disponible sur : <https://aws.amazon.com/fr/compare/the-difference-between-web-apps-native-apps-and-hybrid-apps/> . Consulté le : 08 avril 2025.
- [4] « Avantages et inconvénients des applications mobiles », Oppizi. Disponible sur : [Oppizi, Avantages et inconvénients des applications mobiles](#) .Consulté le : 08 avril 2025.
- [5] « Types des Applications », *Amazon Web Services (AWS)*. Disponible sur : [Amazon Web Services, Différences entre applications web, natives et hybrides](#) .Consulté le : 08 avril 2025.
- [6] « Application native », Disponible sur : <https://www.eurotechconseil.com/glossaire/application-native/> . Consulté le : 08 avril 2025.
- [7] « Avantages et inconvénients des applications natives », *AWS*. Disponible sur : <https://www.kreative.com/blog/application-mobile-native-avantages-inconvenients> . Consulté le : 08 avril 2025.
- [8] « Application web », *Mendix*. Disponible sur : <https://agence-scroll.com/blog/avantages-et-inconvenients-dune-application-web> . Consulté le : 08 avril 2025.
- [9] « Application mobile hybride ». Disponible sur : <https://www.appleute.de/fr/bibliotheque-de-developpeurs-dapplications/native-app-vs-hybrid-app/>. Consulté le : 08 avril 2025.
- [10] « Avantages et inconvénients d'une application hybride – Agence Scroll », *Vertuoz*. Disponible sur : <https://www2.stardust-testing.com/blog-fr/applications-hybrides-avantages-limitations-consequences-phases-de-test>..Consulté le : 08 avril 2025.
- [11] « Application multiplateforme ». Disponible sur : <https://www.eurotechconseil.com/glossaire/application-multiplateforme/> .Consulté le : 08 avril 2025.
- [12] « Avantages et inconvénients d'une application multiplateformes ». Disponible sur : <https://www.nbsolution.link/fr/developpement-dapplications-multiplateformes-avantages-et-inconvenients-pour-les-entreprises> .Consulté le : 08 avril 2025.

- [13] « L'Architecture d'une Application ». Disponible sur : <https://developer.android.com/topic/architecture?hl=fr> . Consulté le : 08 avril 2025.
- [14] « Modules d'application ». Disponible sur : <https://developer.android.com/topic/modularization/patterns?hl=fr> . Consulté le : 08 avril 2025.
- [15] « La plateforme Simbi ». Disponible sur : <https://simbi.com/> . Consulté le : 16/03/2025.
- [16] « La présence mondiale de Fiverr (Plus de 160 pays) ». Disponible sur : <https://www.fiverr.com/news/fall-2024-bti> . Consulté le : 16/03/2025.
- [17] « Les 3,4 millions d'acheteurs actifs dans Fiverr » . Disponible sur : <https://www.sec.gov/ix?doc=/Archives/edgar/data/0001762301/000117891321000690/fvrr20f1220.htm> . Consulté le : 16/03/2025.
- [18] « Application Fiverr ». Disponible sur : <https://www.fiverr.com/>
<https://play.google.com/store/apps/details?id=com.fiverr.fiverr> . Consulté le : 16/03/2025.
- [19] « Définition de plateforme Youchoz ». Disponible sur : <https://www.do4africa.org/projets/youchoz/> . Consulté le : 16/03/2025.
- [20] « Application Youchoz ». Disponible sur : <https://youchoz.com/>
<https://play.google.com/store/apps/details?id=com.youchoz.particulier&hl=ar> .
Consulté le : 16/03/2025.
- [21] « Application Youchoz Pro » Disponible sur : <https://play.google.com/store/apps/details?id=com.youchoz.professionnel> . Consulté le : 16/03/2025.
- [22] « Diagrammes de cas d'utilisation ». Disponible sur : <https://www.ibm.com/docs/fr/dmrt/9.5?topic=diagrams-use-case> . Consulté le : 19/03/2025.
- [23] « Diagramme de séquence ». Disponible sur : <https://www.ibm.com/docs/fr/rsm/7.5.0?topic=uml-sequence-diagrams> . Consulté le : 19/03/2025.
- [24] « Base de données NoSQL ». Disponible sur : <https://datascientest.com/nosql-tout-savoir> . Consulté le : 21/03/2025.
- [25] « Définition du modèle MVVM ». Disponible sur : <https://learn.microsoft.com/fr-fr/dotnet/architecture/maui/mvvm#the-mvvm-pattern> . Consulté le : 13 avril 2025.

- [26] « Définition de la structure du modèle MVVM ». Disponible sur : <https://medium.com/androidmood/comprendre-larchitecture-mvvm-sur-android-aa285e4fe9dd>. Consulté le : 13 avril 2025.
- [27] « combinaison du flutter et firebase». Disponible sur : <https://medium.com/@asajharland/developer-productivity-with-flutter-firebase-02471f267181>. Consulté le : 13 avril 2025.
- [28]« Définition de firebase authentification ». Disponible sur : <https://firebase.google.com/docs/auth?hl=fr> . Consulté le : 13 avril 2025.
- [29]« Définition de cloud firestore ». Disponible sur : <https://firebase.google.com/docs/firestore?hl=fr>. Consulté le : 13 avril 2025.
- [30]« Définition de la méthode SCRUM ». Disponible sur : <https://www.qrpinternational.fr/blog/glossaire/scrum-cest-quoi-definition-scrum/>. Consulté le : 13 avril 2025.
- [31] « Définition diagramme de Gantt » Disponible sur : [Explication des diagrammes de Gantt \[+ Comment en créer un\] | Atlassian](#) . Consulté le : 13 avril 2025.
- [32]« Définition de modelio ». Disponible sur : [Modelio - Créer un diagramme de classe](#). Consulté le : 13 avril 2025.
- [33] « Définition du Vs ». Disponible sur : [Visual Studio Code : l'éditeur de code gratuit et complet de Microsoft](#). Consulté le : 13 avril 2025.
- [34]« Définition du plateforme GitHub ». Disponible sur : <https://www.lemagit.fr/definition/GitHub>: . Consulté le : 13 avril 2025.
- [35]« Définition de Flutter ». Disponible sur : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/flutter-cest-quoi/>. Consulté le : 20 juin 2025.
- [36] « Définition de langue Dart ». Disponible sur : <https://dart.dev/overview>. Consulté le : 20 juin 2025.
- [37] « Définition de l'outil canva ». Disponible sur : https://www.canva.com/fr_fr/about/. Consulté le : 11 mai 2025.

Résumé :

Dans le cadre de ce mémoire, nous avons conçu une application mobile baptisée Échange+, destinée à faciliter les échanges de services entre utilisateurs en Algérie. Elle permet à chacun de publier des annonces, de rechercher des services selon la localisation et en filtrant par catégorie, de choisir un mode d'échange, et d'interagir à travers un système de notifications et d'évaluations. Pensé pour répondre à des besoins concrets du quotidien, cette application se veut simple, pratique et accessible à tous, tout en intégrant des fonctionnalités modernes comme la géolocalisation, l'authentification par réseaux sociaux, et un tableau de bord utilisateur. À travers ce projet, nous avons également envisagé un modèle économique durable basé sur des fonctionnalités payantes ou partenariats, permettant à l'application d'être source de revenus pour ses administrateurs.

Mots clés : Application mobile, Flutter, Dart, Firebase, NoSQL, Android, échange, service, géolocalisation, évaluation, gestion, plateforme.

Abstract :

As part of this thesis, we designed a mobile application called Échange+, intended to facilitate the exchange of services between users in Algeria. It allows users to post ads, search by location, choose an exchange method (free, points, or payment), and interact through a notification and rating system. Designed to meet concrete everyday needs, this application is simple, practical, and accessible to all, while integrating modern features such as geolocation, social network authentication, and a user dashboard. Through this project, we also considered a sustainable business model based on paid features or partnerships, allowing the application to generate revenue for its administrators.

Keywords: Mobile application, Flutter, Dart, Firebase, NoSQL, Android, exchange, service, article, geolocation, evaluation, management, platform.

ملخص:

في إطار هذه المذكرة، قمنا بتصميم تطبيق جوال يسمى Echange+، يهدف إلى تسهيل تبادل الخدمات بين المستخدمين في الجزائر. يتيح للجميع نشر الإعلانات، وإجراء عمليات البحث حسب الموقع، واختيار طريقة التبادل (مجاناً، أو نقاط، أو دفع)، والتفاعل من خلال نظام الإشعارات والتقييمات. تم تصميم هذا التطبيق لتلبية الاحتياجات اليومية الملموسة، ويهدف إلى أن يكون بسيطاً وعملياً ويمكن الوصول إليه من قبل الجميع، مع دمج الميزات الحديثة مثل تحديد الموقع الجغرافي والمصادقة عبر الشبكات الاجتماعية ولوحة معلومات المستخدم. ومن خلال هذا المشروع، أخذنا في الاعتبار أيضاً نموذجاً اقتصادياً مستداماً يعتمد على الميزات المدفوعة أو الشراكات، مما يسمح للتطبيق بأن يكون مصدر دخل لمشرفيه.

الكلمات المفتاحية: تطبيق جوال، Flutter، Dart، Firebase، NoSQL، Android، تبادل، خدمة، تحديد الموقع الجغرافي، تقييم، إدارة، منصة.