

RAPPORT DE STAGE DE FIN D'ETUDES

En vue de l'obtention du diplôme d'Ingénieur d'Etat :

Ingénierie en Réseaux et Systèmes d'Information (IRISI)

Développement d'un Framework iOS pour les applications bancaires

Effectué à :



Adria Business & Technology

Du 20 février au 30 juin 2023

Réalisé par :

Ahmed EL RHAOUTI

Encadré par :

Pr. Saïd RAKRAK

M. Zakaria FEKRANE

Soutenu le 24 juin 2023 devant le jury :

Professeurs

FST Marrakech

Membres de l'Institution d'accueil

Sigle de l'Institution

...

...

Année universitaire 2022/2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dédicace

Je dédie ce travail

À ma mère Saïda est ma raison d'être, elle éclaire mon chemin d'amour et de bienveillance.

Grâce à elle, je suis illuminé par l'affection et l'amour qui m'ont permis d'avancer dans la vie.

À mon père Rahal mérite toute mon affection et ma gratitude pour les nombreux sacrifices et soutiens qu'il m'a accordés.

En signe de mon amour fraternel, je dédie ce travail à mon cher frère et ma sœur, Mariem, Imad. Je vous souhaite une vie remplie de bonheur et de succès, et que le Tout-Puissant Allah vous protège et vous guide dans vos parcours respectifs.

À tous ceux qui me sont chers, qu'ils soient de ma famille, de mes amis ou de mes proches,

Je tiens à leur dédier ce travail, en formulant mes vœux les plus sincères pour un avenir prometteur et épanouissant.

Remerciements

Je tiens à exprimer ma gratitude envers toutes les personnes qui m'ont soutenu et aidé tout au long de mon projet de stage. Sans leur contribution, ce travail n'aurait pas pu être accompli.

Je tiens tout d'abord à exprimer ma gratitude envers **M. BEKKAR RACHID**, le directeur d'Adria Business & Technology, pour avoir accepté ma candidature en tant que stagiaire au sein de l'entreprise.

Je tiens particulièrement à remercier **M. SAID RAKRAK**, mon encadrant pédagogique à la FSTG, pour son excellent encadrement et ses conseils avisés, qui ont grandement contribué à la réussite de mon stage.

Je suis reconnaissant envers **M. ZAKARIA FEKRANE**, mon encadrant chez Adria, pour son accueil chaleureux, sa disponibilité et son partage d'expertise. Sa confiance en moi m'a permis de m'épanouir pleinement dans mes missions.

Je suis reconnaissant envers toute l'équipe d'**Adria B&T** pour leur accueil chaleureux, leur disponibilité et leur soutien tout au long de mon stage.

Enfin, je tiens à remercier l'ensemble du corps administratif de la Faculté des Sciences et Techniques de Marrakech, pour leurs engagements à fournir une formation de qualité, en particulier **M. AZIZ DAROUICHI**, pour leur soutien constant à notre filière. Merci à tous pour votre contribution à mon développement professionnel.

Liste des abréviations

Acronyme	Signification
SDK	Software Development Kit
API	Application Programming Interface
REST	Representational State Transfer
iOS	iPhone Operating System
JSON	JavaScript Object Notation
UML	Unified Modeling Language
VIPER	View Interactor Presenter Entity Router
VIP	View Interactor Presenter

Liste des figures

Figure 1 : LOGO D'entreprise ADRIA	15
Figure 2 : Produits et services ADRIA	16
Figure 3 : Les références d'Adria	17
Figure 4 : Adria B&T présence dans le monde	17
Figure 5 : Les processus du méthode agile SCRUM	19
Figure 6 : Diagramme de Gantt des taches	21
Figure 7 : Logo ALAMOFIRE package	25
Figure 8 : Architecture d'une application se base sur ALAMOFIRE	25
Figure 9 : Architecture d'une application se base sur SDK	27
Figure 10 : Logo UML	31
Figure 11 : Diagramme des séquences de package	32
Figure 12 : Diagramme de séquences de recherche des mouvements	34
Figure 13 : Diagramme de séquences de récupération des messages envoyées	35
Figure 14 : Diagramme de séquences de récupération des messages reçus	35
Figure 15 : Diagramme de séquences de récupération des sujets de messagerie	36
Figure 16 : Diagramme de séquences de récupération des destinataires	36
Figure 17 : Diagramme de séquences d'enregistrement des messages	37
Figure 18 : Diagramme de séquences de récupération des documents	38
Figure 19 : Diagramme de séquences de récupération d'un document	38
Figure 20 : Diagramme de séquences de téléchargement d'un document	39
Figure 21 : Diagramme de cas d'utilisation de l'application iOS	40
Figure 22 : Architecture VIPER	43
Figure 23 : Architecture de package	45
Figure 24 : Architecture de package détaillée	45
Figure 25 : Architecture d'application	47
Figure 26 : Architecture d'application détaillée	47
Figure 27 : Logo de Swift	48
Figure 28 : Logo de Wormholy	48
Figure 29 : Les interface offerte par Wormholy	49
Figure 30 : Logo de Xcode	50
Figure 31 : Logo de Git	51
Figure 32 : Logo de Bitbucket	51
Figure 33 : Logo de Postman	52
Figure 34 : Logo de XCTest	53
Figure 35 : Page d'authenfication	56
Figure 36 : Page d'accueil	57
Figure 37 : Page des mouvements	58
Figure 38 : Page des messages envoyée et reçus	59
Figure 39 : Page de liste des destinataires et sujets de messages	59
Figure 40 : Page d'envoi d'un message	60
Figure 41 : Page de liste des documents et page de contenu du document	61
Figure 42 : Page de contenu de document avec le bouton de téléchargement	61

Table des matières

Dédicace	3
Remerciements.....	4
Liste des abréviations	5
Liste des figures	6
Résumé	10
Abstract	11
ملخص	12
Introduction générale.....	13
Chapitre 1 : Contexte General de projet	14
1. Introduction :.....	15
2. Présentation de l'organisme d'accueil :	15
2.1. Présentation de l'organisme :.....	15
2.2. Produits et services :.....	15
2.3. Principales références :	16
2.4. Présence à l'international :.....	17
3. Objectifs du projet de stage :	18
4. Conduite de projet :.....	18
4.1. Méthode de travail :.....	18
4.2. Planification de projet :	20
5. Conclusion :	22
Chapitre 2 : Analyse et spécification des besoins	23
1. Introduction :.....	24
2. Présentation de projet :.....	24
3. Étude et critique de l'existant :	24
3.1. Description de l'existant :.....	25
3.2. Critique de l'existant :.....	26
3.3. Solution envisager :	26
4. Spécification des besoins :.....	27
4.1. Les besoins fonctionnels :.....	28
4.2. Les besoins non fonctionnels :	29
5. Conclusion :	29

Liste des figures

Chapitre 3 : Conception et modélisation du projet.....	30
1. Introduction.....	31
2. Présentation de l’UML.....	31
3. Conception générale du projet.....	31
4. Conception des tâches effectuées.....	33
4.1. Récupération des mouvements d’un client :.....	33
4.2. Récupération des messages envoyé et reçus d’un client :.....	34
4.3. Récupération des destinataires et les sujets de messagerie d’un client :.....	35
4.4. Enregistrement d’un nouveau message d’un client :.....	36
4.5. Récupération et téléchargement des documents d’un client :.....	37
5. Conception d’application de teste	39
6. Conclusion	41
Chapitre 4 : Etude technique du projet.....	42
1. Introduction.....	43
2. Architecture de projet	43
2.1. Présentation de L’architecture VIPER :.....	43
2.2. Implémentation de l’architecture VIPER dans le package :	44
2.3. Implémentation de l’architecture VIPER dans l’application de teste :.....	46
3. Langage et technologies de développement	47
3.1 la langage Swift :.....	48
3.2 la bibliothèque Wormholly :.....	48
4. Outils et environnement de travail	49
4.1. Xcode :	50
4.2. Git :	51
4.3. BitBucket :.....	51
5. Les tests unitaires	52
5.1. Objectif des tests unitaires :	52
5.2. Avantages des tests unitaires :	52
5.3. XCTest framework :	53
6. Conclusion	54
Chapitre 5 : Mis en œuvre du projet	55
1. introduction :.....	56
2. Rapport de l’ensemble des réalisations dans le projet :	56

2.1. Authentification :	56
2.2. Récupération des mouvements :	57
2.3. Récupération des messages envoyées et reçus :	58
2.4. Récupération des destinataires et sujets de messages :	59
2.5. Enregistrement d'un nouveau message :	60
2.5. Récupération de la liste des documents d'un client :	60
2.6. Téléchargement d'un document :	61
4. conclusion :	62
Conclusion générale :	63
Webographie	64
1. Site web officiel de l'entreprise Adria B&T :	64
2. Documentation technique du projet ou de la technologie utilisée :	64

Résumé

L'objectif du projet de fin d'étude chez Adria Business & Technology est de mettre en place un package (SDK, Framework) visant à améliorer et accélérer le processus de développement des applications iOS dans le domaine de la banque digitale. Cette solution permettra une conception et un déploiement plus rapides et plus fiables des applications, contribuant ainsi à renforcer la compétitivité d'ADRIA sur le marché de la banque digitale. De plus, le package joue le rôle d'un intermédiaire entre l'application qui l'utilise et le serveur qui offre les données via une API.

Le package fournit une structure de base pour le développement d'applications iOS, ce qui permet de gagner du temps et de réduire les erreurs de codage. Enfin, le package peut également fournir des outils et des fonctionnalités supplémentaires pour améliorer les performances et la sécurité de l'application.

Pour mettre en place ce logiciel, nous avons commencé par rassembler les différentes fonctionnalités qui communiquent directement avec les API offertes par les serveurs dans un seul fichier. Ce fichier représente l'ensemble des tâches à implémenter dans le projet de fin d'étude. Ensuite, nous avons effectué l'analyse du projet, suivi de la conception et de la mise en place de l'architecture du projet, avant de commencer la phase de réalisation.

Afin d'assurer de bonnes conditions de travail lors de la réalisation d'un projet, nous avons choisi d'utiliser la méthode agile "Scrum". Cette méthode découpe le projet en plusieurs sprints. À la fin de chaque sprint, nous présentons l'avancement effectué et essayons de nous améliorer pour la prochaine étape.

Ce rapport a pour objectif de décrire de manière détaillée les différentes étapes qui ont permis la réalisation de ce projet.

Abstract

The objective of the end-of-study project at Adria Business & Technology is to set up a package (SDK, Framework) aimed at improving and accelerating the process of developing iOS applications in the field of digital banking. This solution will enable faster and more reliable design and deployment of applications, thus helping to strengthen ADRIA's competitiveness in the digital banking market. Moreover, the package acts as an intermediary between the application that uses it and the server that offers the data via an API.

The package provides a basic structure for iOS app development, which saves time and reduces coding errors. Finally, the package may also provide additional tools and features to improve application performance and security.

To set up this software, we started by gathering the different functionalities that communicate directly with the APIs offered by the servers in a single file. This file represents all the tasks to be implemented in the graduation project. Then, we carried out the analysis of the project, followed by the design and implementation of the project architecture, before starting the implementation phase.

In order to ensure good working conditions when carrying out a project, we have chosen to use the agile "Scrum" method. This method divides the project into several sprints. At the end of each sprint, we present the progress made and try to improve for the next stage.

The purpose of this report is to describe in detail the different stages that have made it possible to carry out this project.

ملخص

الهدف من مشروع نهاية الدراسة في Technology & Adria Business هو إعداد مكتبة (إطار عمل) تهدف إلى تحسين وتسريع عملية تطوير تطبيقات iOS في مجال الخدمات المصرفية الرقمية. سيتمكن هذا الحل من تصميم ونشر التطبيقات بشكل أسرع وأكثر موثوقية ، مما يساعد على تعزيز القدرة التنافسية لـ ADRIA في سوق الخدمات المصرفية الرقمية. علاوة على ذلك ، تعمل المكتبة كوسيط بين التطبيق الذي يستخدمها والخدم الذي يوفر البيانات عبر واجهة برمجية.

توفر المكتبة بنية أساسية لتطوير تطبيقات iOS ، مما يوفر الوقت ويقلل من أخطاء البرمجة. أخيرًا ، قد توفر المكتبة أيضًا أدوات وميزات إضافية لتحسين أداء التطبيق وأمانه.

لإعداد هذا البرنامج ، بدأنا بجمع الوظائف المختلفة التي تتصل مباشرة بواجهات البرمجية API التي تقدمها الخوادم في ملف واحد. يمثل هذا الملف جميع المهام التي سيتم تنفيذها في مشروع التخرج. ثم قمنا بتحليل المشروع تلاه تصميم وتنفيذ هيكل المشروع قبل البدء في مرحلة التنفيذ.

من أجل ضمان ظروف عمل جيدة عند تنفيذ المشروع ، اخترنا استخدام المنهجية "SCRUM" المرنة. تقسم هذه الطريقة المشروع إلى عدة مراحل . في نهاية كل مرحلة ، نقدم التقدم المحرز ونحاول تحسين المرحلة التالية.

الغرض من هذا التقرير هو الوصف التفصيلي للمراحل المختلفة التي جعلت من الممكن تنفيذ هذا المشروع.

Introduction générale

Le projet de fin d'études chez Adria Business & Technology a donné naissance à un package visant à améliorer et accélérer le processus de développement des applications iOS dans le domaine de la banque digitale. Ce package permet, par exemple, de récupérer les mouvements, d'envoyer des messages ou de télécharger des documents. Il joue le rôle d'intermédiaire entre l'application et le serveur via une API, permettant ainsi une conception et un déploiement plus rapides et plus fiables des applications. Il fournit également des outils et des fonctionnalités supplémentaires pour améliorer les performances et la sécurité de l'application.

Le projet a été réalisé en utilisant la méthode agile Scrum, découpant le projet en plusieurs sprints pour assurer de bonnes conditions de travail et permettre une présentation de l'avancement à chaque étape. Dans ce rapport, nous décrivons en détail les différentes étapes qui ont permis la réalisation de ce projet, depuis la collecte des différentes fonctionnalités jusqu'à la phase de réalisation, en passant par l'analyse et la conception de l'architecture du projet. Nous présentons également les choix technologiques effectués pour la mise en place de ce package, ainsi que les tests et les résultats obtenus.

Dans le premier chapitre, nous décrivons le contexte général du projet, en commençant par la présentation détaillée de l'entreprise d'accueil, son domaine d'activité et ses enjeux. Nous expliquons ensuite la méthode de travail adoptée pour mener à bien ce projet.

Le deuxième chapitre est dédié à l'étude des besoins et à la justification de l'existence de ce projet, en mettant en évidence son importance par rapport aux solutions existantes et en présentant les différentes fonctionnalités qu'il offre.

Dans le troisième chapitre, nous détaillons la partie conception et modélisation qui a été mise en place pour réussir la réalisation de ce projet.

Le quatrième chapitre présente en détail les différents outils utilisés afin de réaliser le projet, en précisant leur rôle et leur utilité.

Enfin, le dernier chapitre montre les différentes tâches réalisées durant la période de stage pour mener à bien le projet. Nous y présentons les résultats obtenus

Chapitre 1 : Contexte General de projet

- 1- Introduction
- 2- Présentation de l'organisme d'accueil
- 3- Objectifs du projet de stage
- 4- Conduite de projet
- 5- Conclusion

1. Introduction :

Le premier chapitre de ce rapport expose le contexte général du projet. Tout d'abord, il présente l'organisme d'accueil, ADRIA BUSINESS & TECHNOLOGY. Ensuite, il rappelle le sujet de mon projet de stage ainsi que les objectifs à atteindre et la planification de ce projet pour la durée du stage.

2. Présentation de l'organisme d'accueil :

2.1. Présentation de l'organisme :



Figure 1 : LOGO D'entreprise ADRIA

ADRIA BUSINESS & TECHNOLOGY est une entreprise présente dans plus de 20 pays, qui se spécialise dans l'édition et l'intégration de logiciels destinés aux banques et institutions financières.

Sa mission est d'accompagner les organisations financières dans leurs projets de Digital Banking en leur offrant des solutions performantes et des services à haute valeur ajoutée. Avec une profonde compréhension du secteur bancaire, Adria crée des produits bien conçus pour aider les banques à établir leur stratégie numérique au sein d'une structure de coûts définie et adaptée à chaque cas d'utilisation.

Cette entreprise est engagée et impliquée dans la transformation digitale de ses clients, en leur proposant des solutions personnalisables à la pointe de la technologie et en les accompagnant depuis le cadrage de leur projet jusqu'à la mise en service de leurs solutions.

En somme, Adria Business & Technology est bien plus qu'un prestataire de services, c'est une équipe fiable et innovante tournée vers l'avenir et le succès de ses clients.

2.2. Produits et services :

L'objectif premier d'Adria Business & Technology est de garantir la qualité et la pertinence de ses produits et services afin de répondre efficacement aux besoins actuels et futurs de ses

clients. La société propose une gamme diversifiée de produits et de services qui couvrent tous les canaux de communication, tels que le web et le mobile :

- **Adria banque directe cross canal :**

Ce produit modulaire est conçu pour couvrir l'ensemble du parcours client en ligne, en proposant des fonctionnalités de marketing, de vente en ligne, de services transactionnels et de communication avec le client.

- **Adria mobile branch banking :**

Elle propose une solution complète pour les agences mobiles et les agents nomades, couvrant toutes les fonctionnalités d'entrée en relation avec les clients, de souscription de produits, de gestion de comptes et de services transactionnels.

- **Adria analytics :**

Le produit d'Adria permet de recueillir, d'analyser et de présenter de manière visuelle toutes les interactions entre les clients et la banque, peu importe le canal utilisé.

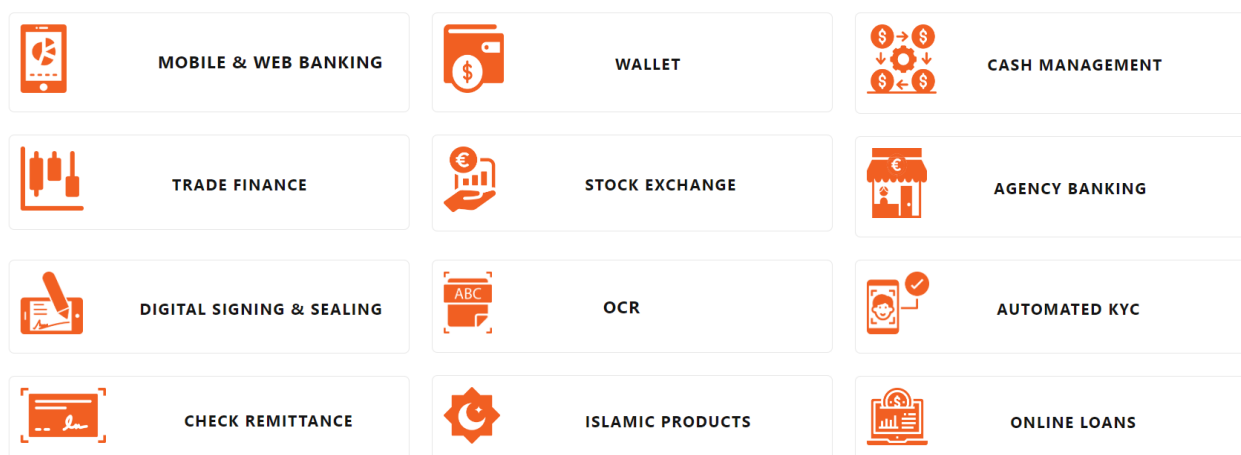


Figure 2 : Produits et services ADRIA

2.3. Principales références :

Adria B&T se positionne comme un acteur incontournable dans l'édition et l'intégration de logiciels destinés aux banques et institutions financières. La figure ci-dessous

présente les références majeures de l'expertise et du savoir-faire dans le domaine du Digital Banking d'Adria B&T.



Figure 3 : Les références d'Adria

2.4. Présence à l'international :

Grâce à la qualité de ses services, Adria a su satisfaire les attentes de grandes sociétés bancaires, tant au Maroc qu'à l'étranger, et a ainsi pu se constituer une base clientèle solide. Forte de ce potentiel, Adria a su élargir ses horizons, passant d'une entreprise locale à une entreprise internationale, et gagnant la confiance de clients en Afrique et en Europe.



Figure 4 : Adria B&T présence dans le monde

3. Objectifs du projet de stage :

Le projet de fin d'études consiste à la mise en place d'un SDK en utilisant le langage Swift. Ce SDK sera utilisable par toutes les applications iOS des produits e-Banking de l'entreprise ADRIA. Il permettra ainsi de faciliter la communication entre les services Web (API RESTful) et l'application qui implémente ce programme. Cette solution jouera le rôle d'un intermédiaire entre l'application et l'API du serveur, et elle présente plusieurs avantages, notamment :

- Une meilleure gestion des erreurs et des exceptions lors de la communication avec l'API.
- Une réduction du temps de développement et de maintenance des applications, car le SDK fournit des fonctionnalités communes pour toutes les applications.
- Une amélioration de la sécurité, car le SDK utilise des techniques de cryptage pour protéger les données sensibles transmises entre l'application et l'API.
- Une augmentation de la performance, car le SDK utilise des techniques d'optimisation pour accélérer les communications entre l'application et l'API.
- Facilite les tests unitaires des applications qui l'utilisent.
- Simplifier la communication entre l'application et l'api.
- Simple a utilisé.

Afin d'assurer une bonne utilisation de ce SDK, il est nécessaire de réaliser une documentation complète avec des exemples de code. De plus, la réalisation du SDK doit se faire sans aucune bibliothèque externe.

4. Conduite de projet :

La conduite de projet est une démarche visant à organiser et à optimiser la réalisation d'un projet en vue d'atteindre ses objectifs de manière efficace. Dans cette optique, notre entreprise privilégie l'utilisation des "Méthodes Agiles" pour le développement de ses projets, afin de favoriser la flexibilité, la collaboration et la qualité du travail réalisé.

4.1. Méthode de travail :

La méthode agile est une approche de gestion de projet qui se concentre sur la flexibilité, l'adaptation au changement et la collaboration entre les membres de l'équipe. Elle vise à

fournir des résultats rapides et de haute qualité en divisant le projet en étapes plus courtes appelées « sprints ».

Pour assurer de bonnes conditions de travail lors de la réalisation d'un projet, nous essayons d'organiser notre travail en utilisant la méthode agile « SCRUM ». Le diagramme suivant montre les processus du méthode SCRUM :

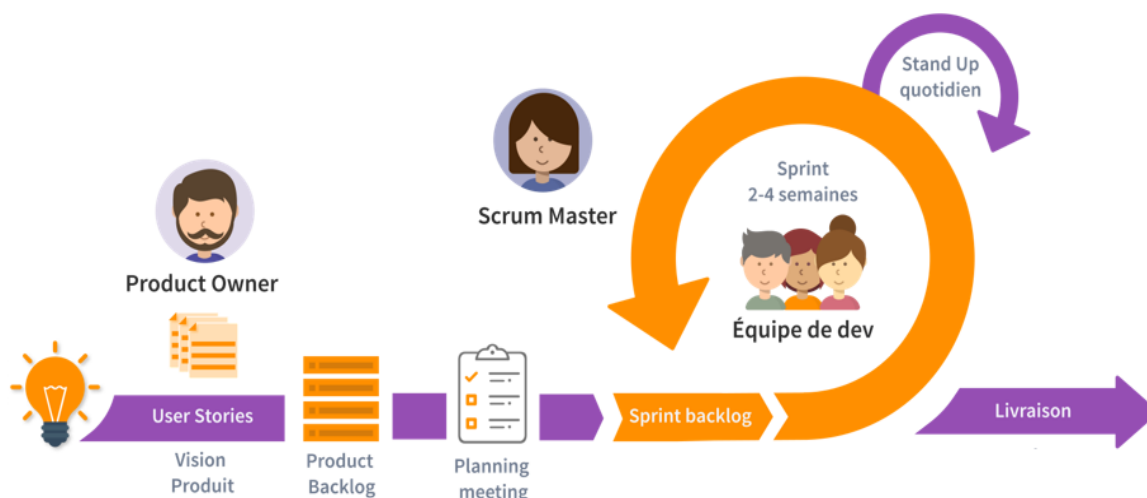


Figure 5 : Les processus du méthode agile SCRUM

Cela permet de mieux gérer les tâches à effectuer et de s'adapter aux changements qui peuvent survenir en cours de route. La méthode SCRUM implique des réunions régulières pour faire le point sur l'avancement du projet, identifier les obstacles et les résoudre rapidement. Elle encourage également une communication transparente et une collaboration étroite entre tous les membres de l'équipe.

La méthode SCRUM implique également des réunions régulières pour assurer une communication transparente et une collaboration étroite entre tous les membres de l'équipe. Les réunions principales sont les suivantes :

- **La réunion de planification de sprint** : c'est une réunion qui se tient au début de chaque sprint pour définir les objectifs et les tâches à réaliser durant le sprint.
- **Le daily Scrum** : c'est une réunion quotidienne qui se tient chaque matin pour faire le point sur l'avancement du sprint et identifier les obstacles éventuels.

- **La revue de sprint** : c'est une réunion qui se tient à la fin de chaque sprint pour présenter les résultats et les avancées de l'équipe.
- **La rétrospective de sprint** : c'est une réunion qui se tient également à la fin de chaque sprint pour évaluer le travail réalisé, identifier les points positifs et négatifs et suggérer des améliorations pour les sprints suivants.

En plus de ces réunions, la méthode SCRUM implique plusieurs rôles clés pour mieux gérer les tâches à effectuer et s'adapter aux changements qui peuvent survenir en cours de route. Les rôles principaux sont les suivants :

- **Le Scrum Master** : c'est le garant de la méthode Scrum et de son bon déroulement. Il s'assure que les membres de l'équipe respectent les règles, les rituels et les valeurs de Scrum. Il aide également l'équipe à surmonter les obstacles et à progresser de manière efficace et efficiente.
- **Le Product Owner** : c'est le représentant des clients et des utilisateurs. Il est responsable de la définition et de la priorisation des éléments du backlog produit, c'est-à-dire la liste de toutes les fonctionnalités et les tâches à réaliser pour le produit. Il est également responsable de l'acceptation des livrables de l'équipe.
- **L'équipe de développement** : c'est l'équipe qui réalise les tâches et les fonctionnalités du backlog produit. Elle est autoorganisée, pluridisciplinaire et autonome. Elle est composée de tous les professionnels nécessaires à la réalisation du produit (développeurs, testeurs, designers, etc.). Elle est responsable de la qualité du travail qu'elle réalise.

4.2. Planification de projet :

La réalisation de ce projet se fait par l'intermédiaire de sprints réguliers, avec une nouvelle planification des tâches à réaliser pour chaque sprint. Dans le diagramme de Gantt suivant, la planification des tâches qui ont été réalisées durant cette période de stage est représentée :

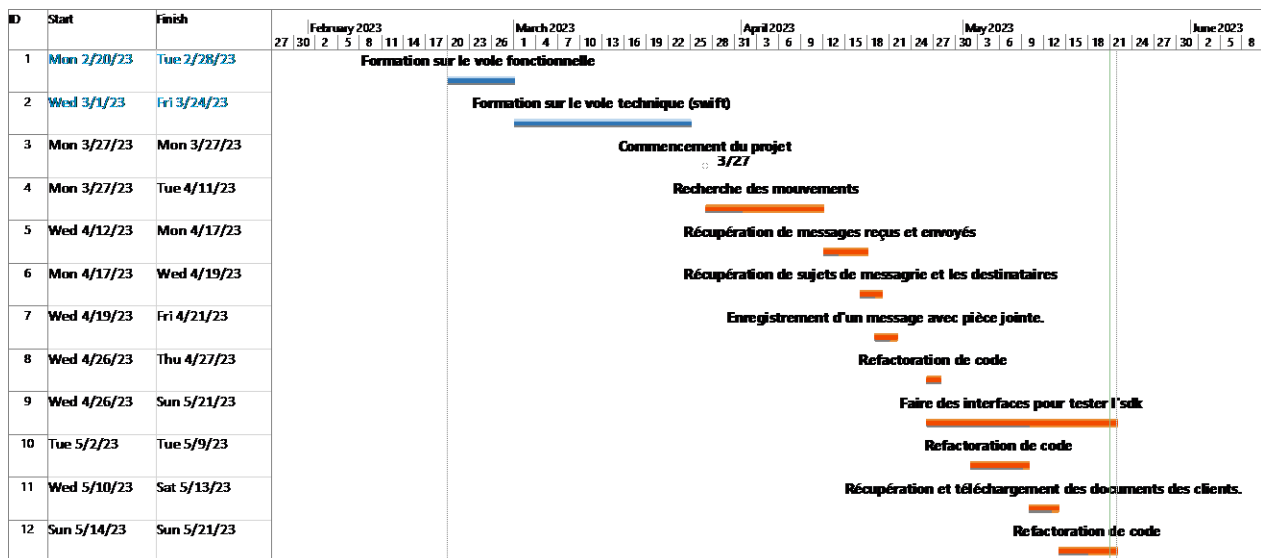


Figure 6 : Diagramme de Gantt des tâches

Dans ce diagramme, chaque tâche est composée de trois parties essentielles permettant de la décrire clairement. Ces parties sont les suivantes :

- **Durée réelle de réalisation de la tâche (en orange clair) :** il s'agit du temps effectivement nécessaire pour terminer la tâche, mesuré en jours.
- **Durée provisionnelle (estimée) de la tâche lors de l'affectation de la tâche (en gris) :** il s'agit de la durée estimée de la tâche lorsqu'elle a été initialement affectée. Cette durée peut être modifiée en fonction de l'avancement réel de la tâche.
- **La progression de la tâche qui représente le pourcentage qui a été complété de la tâche (en orange foncé) :** il s'agit de la mesure de l'avancement de la tâche, exprimée en pourcentage du travail total qui a été effectué jusqu'à présent. Cela permet de savoir si la tâche est en avance, en retard ou dans les délais par rapport au plan initial.

5. Conclusion :

Le but de ce chapitre était d'introduire le projet en présentant l'entreprise ADRIA B&T, ses activités et son organisation, ainsi que les différentes étapes et phases du projet. Le prochain chapitre sera consacré à l'analyse et à la spécification des besoins.

Chapitre 2 : Analyse et spécification des besoins

- 1- Introduction
- 2- Présentation de Projet
- 3- Étude d'existant
- 4- Spécification des besoins
- 5- Conclusion

1. Introduction :

Dans ce chapitre, nous examinerons de près les spécifications des besoins du système. Pour ce faire, nous allons passer en revue les exigences fonctionnelles et non fonctionnelles du projet, dans le but de mieux comprendre les fonctionnalités attendues et les performances souhaitées.

2. Présentation de projet :

Notre projet vise à développer un package destiner aux développeurs iOS internes de l'entreprise Adria, qui simplifiera considérablement les tâches de communication avec un serveur cible. En raison de la complexité croissante des applications mobiles et du nombre croissant de projets à finaliser, la communication avec les serveurs est devenue une tâche cruciale et répétitive pour les développeurs iOS. Cette tâche peut s'avérer fastidieuse et prendre beaucoup de temps et d'efforts, en particulier lorsqu'elle doit être répétée à plusieurs reprises dans différents projets.

Le projet s'inscrit dans le cadre de la préparation du stage de fin d'études en vue de l'obtention du diplôme d'Ingénieur d'Etat en Réseaux et Systèmes d'Information (IRSI) à la Faculté des Sciences et Techniques de Marrakech. Il a été effectué au sein de la société Adria Business & Technology pour une période de quatre mois. Nous nous engageons à fournir une solution efficace pour faciliter la communication entre les applications mobiles et les serveurs, tout en veillant à ce que le package soit adapté aux besoins spécifiques d'Adria.

Le package sera conçu de manière à être facilement utilisable, bien documenté et sécurisé, offrant des performances optimales pour répondre aux besoins de l'entreprise. Nous mettrons également l'accent sur la convivialité de l'interface utilisateur, pour faciliter la prise en main du package par les développeurs iOS d'Adria. Les fonctionnalités du package développer doivent être soumises à des tests unitaires pour s'assurer de leur fonctionnement correct et efficace.

En résumé, notre objectif est de développer un package efficace et facile à utiliser, qui permettra à l'entreprise de gagner du temps et de l'énergie, tout en améliorant la qualité des applications mobiles développées.

3. Étude et critique de l'existant :

L'étude de l'existant est essentielle pour éviter la duplication d'efforts et identifier les solutions déjà existantes. Je présente ici l'étude d'existant de notre projet.

3.1. Description de l'existant :

ALAMOFIRE

Elegant Networking in Swift

Figure 7 : Logo ALAMOFIRE package

Alamofire est une bibliothèque open-source très utilisée pour la programmation réseau en Swift. Elle simplifie la communication avec des API RESTful et autres services web en offrant des fonctionnalités avancées telles que la gestion des tâches de fond, l'authentification, la gestion des erreurs, ainsi que le traitement de données JSON. Cette bibliothèque est largement utilisée dans le développement d'applications iOS, en particulier celles qui interagissent avec des services web pour récupérer ou envoyer des données. La société Adria Business & Technology utilise également Alamofire pour développer des applications iOS répondant aux besoins de ses clients. L'image suivante décrit l'architecture des applications iOS développées par Adria en se basant sur ce package :

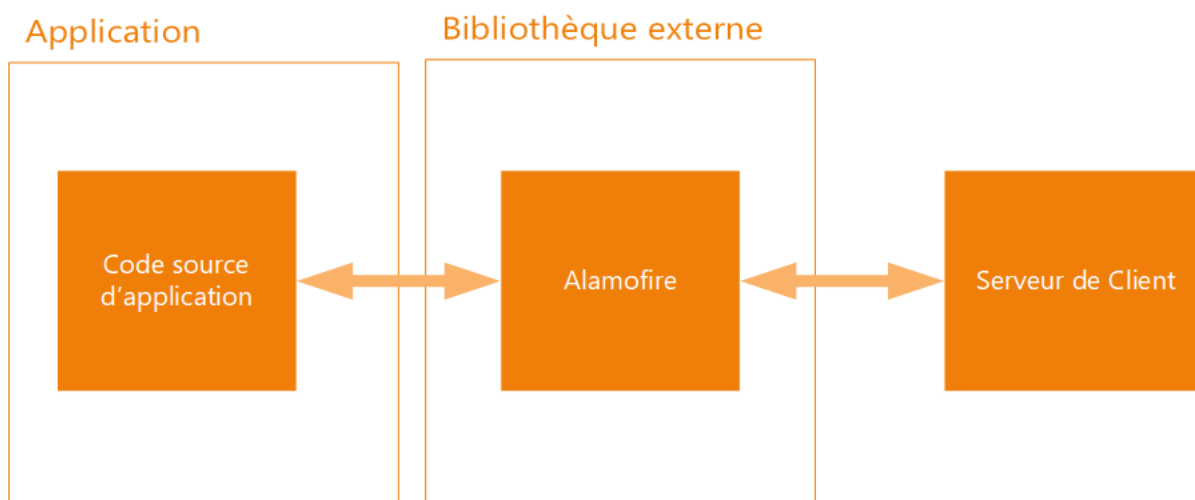


Figure 8 : Architecture d'une application se base sur ALAMOFIRE

3.2. Critique de l'existant :

La solution présentée dans la section précédente présente plusieurs avantages, tels que sa compatibilité avec tous les projets iOS, sa flexibilité et son optimisation. Cependant, le problème réside dans le fait qu'elle est une bibliothèque externe, ce qui pose un inconvénient majeur lors de son utilisation dans le domaine du développement d'applications iOS. En général, l'utilisation d'une bibliothèque externe entraîne des difficultés lors de l'exportation des applications. En outre, Elle peut présenter des inconvénients pour un utilisateur particulier qui développe plusieurs projets ayant presque les mêmes fonctionnalités, ce qui peut entraîner une lenteur dans la production des applications, car cette solution est considérée comme native, ce qui signifie que les développeurs doivent l'intégrer dans chaque application et la configurer en fonction des besoins spécifiques de chaque projet.

Cela peut prendre du temps et poser des problèmes lors de l'extraction de l'application finale. De plus, la mise à jour de la bibliothèque peut nécessiter des efforts supplémentaires pour assurer la compatibilité avec les applications qui l'utilisent.

3.3. Solution envisager :

Notre objectif principal dans ce projet est de développer un Framework complet répondant aux besoins des développeurs iOS d'Adria dans la réalisation de leurs projets. Contrairement aux solutions précédentes, notre Framework offrira plusieurs avantages, tels que :

- **Interfaces adaptées :** Notre package offre des interfaces adaptées à tous les projets iOS développés par Adria, ce qui garantit une expérience utilisateur plus optimale et cohérente.
- **Temps de développement plus rapide :** Notre Framework iOS pour Adria permet un développement plus rapide grâce à des fonctionnalités prédéfinies spécifiques aux projets iOS d'Adria et une interface unifiée simplifiant les tâches de développement.
- **Réduction des erreurs et des incohérences :** grâce à l'utilisation d'un Framework iOS, les développeurs peuvent suivre des normes et des conventions de codage établies, ce qui réduit les risques d'erreurs et d'incohérences dans le code. Cela

peut également faciliter la maintenance et l'évolution de l'application à long terme.

- **Sécurité améliorée** : Notre Framework offre une solution de gestion d'authentification hautement personnalisée et sécurisée pour garantir la confidentialité et la sécurité des données utilisateur.
- **Réduire le nombre de unit tests** : un autre avantage de notre Framework. Grâce à son architecture unifiée, les développeurs peuvent se concentrer sur les tests les plus critiques et essentiels, tout en minimisant la duplication des tests. Cela permet de réduire le temps et les efforts nécessaires pour effectuer des tests et facilite la maintenance du code.

De plus, ce package permet de contourner le problème d'Alamofire. Cela signifie que les applications développées en utilisant cette solution n'auront plus de problème lors de l'exportation des résultats finaux.

L'image suivante décrit l'architecture des applications iOS développées par Adria en se basant sur notre solution :

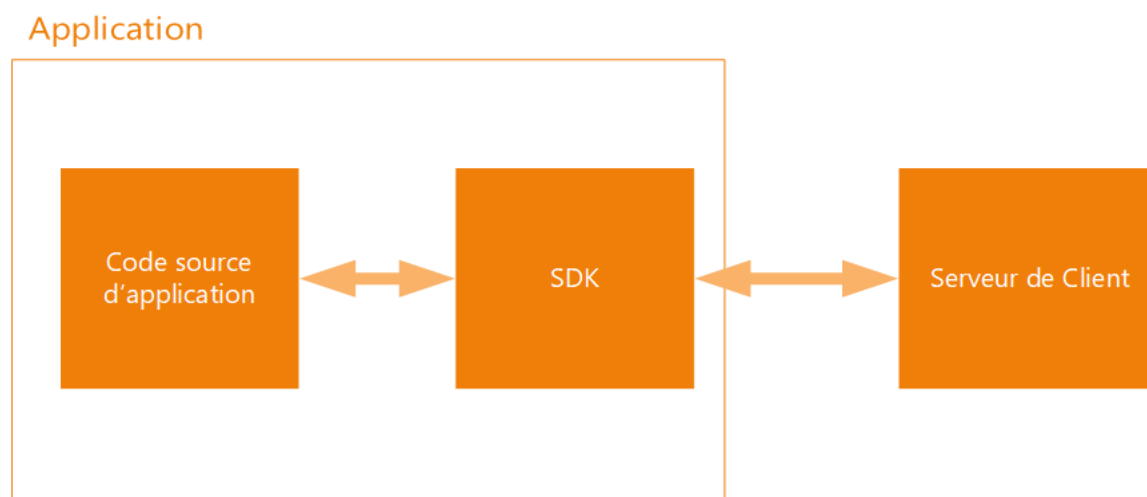


Figure 9 : Architecture d'une application se base sur SDK

4. Spécification des besoins :

Cette partie discutera des besoins fonctionnels et non fonctionnels du Framework que nous développerons pour répondre aux exigences de l'utilisateur :

4.1. Les besoins fonctionnels :

Dans cette section, nous allons explorer les besoins fonctionnels de notre projet, qui sont essentiels pour offrir une expérience utilisateur fluide et efficace. Parmi ces besoins, nous pouvons citer :

Gestion de la communication entre l'application et le serveur de manière sécurisée et personnalisée.

- Interface utilisateur adaptée aux projets iOS d'Adria pour faciliter la tâche des développeurs.
- Gestion de l'authentification de manière sécurisée pour garantir la confidentialité des données utilisateur.
- Facilitation de l'intégration sans dépendance à des bibliothèques externes pour éviter les problèmes de version et de configuration.
- Fourniture d'une documentation complète et facile à comprendre pour permettre aux développeurs de comprendre rapidement l'utilisation du Framework.
- Capacité à gérer efficacement la gestion de la mémoire pour éviter les problèmes de performance et de stabilité.
- Offrir une interface unifiée pour simplifier les tâches de développement et accélérer la production d'applications.
- Fournir des fonctionnalités prédéfinies pour réduire le temps de développement et éviter la réinvention de la roue.
- Support pour les dernières versions d'iOS et les fonctionnalités les plus récentes pour garantir que les applications sont à jour et utilisent les meilleures pratiques.
- Capacité à s'adapter à différents projets et besoins pour offrir une solution flexible et personnalisée.

4.2. Les besoins non fonctionnels :

Dans cette section, nous allons explorer les besoins non fonctionnels de notre projet, qui sont tout aussi essentiels que les besoins fonctionnels pour offrir une expérience utilisateur de qualité. Les besoins non fonctionnels concernent généralement les caractéristiques du système telles que la sécurité, la performance, la fiabilité, l'évolutivité, la facilité d'utilisation et la convivialité. Parmi ces besoins, nous pouvons citer :

- La performance : le package doit être optimisé pour offrir des performances rapides et fiables lors de la communication avec les serveurs cibles, afin de réduire le temps de développement des applications mobiles.
- La fiabilité : le package doit être conçu pour fonctionner sans interruption et sans erreurs, afin de minimiser les temps d'arrêt et les erreurs potentielles qui pourraient affecter le développement des applications mobiles.
- La sécurité : le package doit être sécurisé pour garantir que les données des utilisateurs et les communications entre les applications mobiles et les serveurs sont protégées contre les attaques externes.
- L'évolutivité : le package doit être conçu pour être facilement extensible, afin de permettre l'ajout de nouvelles fonctionnalités à l'avenir, en fonction des besoins changeants de l'entreprise Adria.
- La facilité d'utilisation : le package doit être conçu pour être facile à utiliser, avec une documentation claire et concise, pour faciliter la prise en main par les développeurs iOS d'Adria qui n'ont pas nécessairement une grande expertise en communication avec les serveurs.
- La convivialité : l'interface utilisateur du package doit être conviviale et facile à utiliser, avec des messages d'erreur clairs et des explications détaillées pour minimiser les erreurs de communication avec les serveurs.

5. Conclusion :

En conclusion, l'analyse approfondie des spécifications des besoins du système est indispensable pour garantir la conception efficace et pertinente du projet de développement. Cela permet de mieux comprendre les exigences fonctionnelles et non fonctionnelles du projet et d'identifier clairement les fonctionnalités attendues et les performances souhaitées, pour répondre aux attentes des utilisateurs finaux et de l'entreprise.

Chapitre 3 : Conception et modélisation du projet

- 1- Introduction
- 2- Présentation de l'UML
- 3- Conception générale du projet
- 4- Conception des tâches effectuées
- 5- Conception d'application de teste
- 6- Conclusion

1. Introduction

Ce chapitre présente les différentes conceptions effectuées dans le cadre de ce projet. Il est divisé en quatre sections : la première expose une présentation générale de l'UML, la deuxième présente la conception globale du projet, la troisième se concentre sur la conception des tâches et des

2. Présentation de l'UML



Figure 10 : Logo UML

L'UML (Unified Modeling Language) est un langage de modélisation graphique standardisé créé au début des années 1990. Il permet de représenter les différentes parties d'un système logiciel, telles que les fonctionnalités, les classes, les relations entre les classes, les flux de données, etc. L'UML se compose de différents types de diagrammes, tels que les diagrammes de cas d'utilisation, les diagrammes de classes, les diagrammes de séquence, etc. Il est largement utilisé dans l'ingénierie logicielle, les systèmes d'information, la modélisation de processus métier, l'architecture d'entreprise, etc.

L'utilisation de l'UML permet de faciliter la communication et la collaboration entre les différents acteurs impliqués dans le développement d'un système logiciel. En utilisant un langage commun et des représentations graphiques, l'UML permet aux développeurs, architectes, chefs de projet et clients de comprendre rapidement et précisément les différentes parties du système logiciel en cours de développement. L'UML est également largement utilisé dans les méthodes de développement agiles, où il permet de représenter les différentes fonctionnalités à développer sous forme de "stories" ou de "scenarios".

3. Conception générale du projet

Pour réaliser ce package, nous nous sommes principalement basés sur le diagramme de séquence ci-dessous qui présente de manière globale les séquences que notre projet doit suivre

pour effectuer n'importe quel service demandé. Ce diagramme de séquence nous a permis de mieux comprendre l'interaction entre les différents acteurs du système et d'identifier les différentes étapes nécessaires à la réalisation des services demandés.

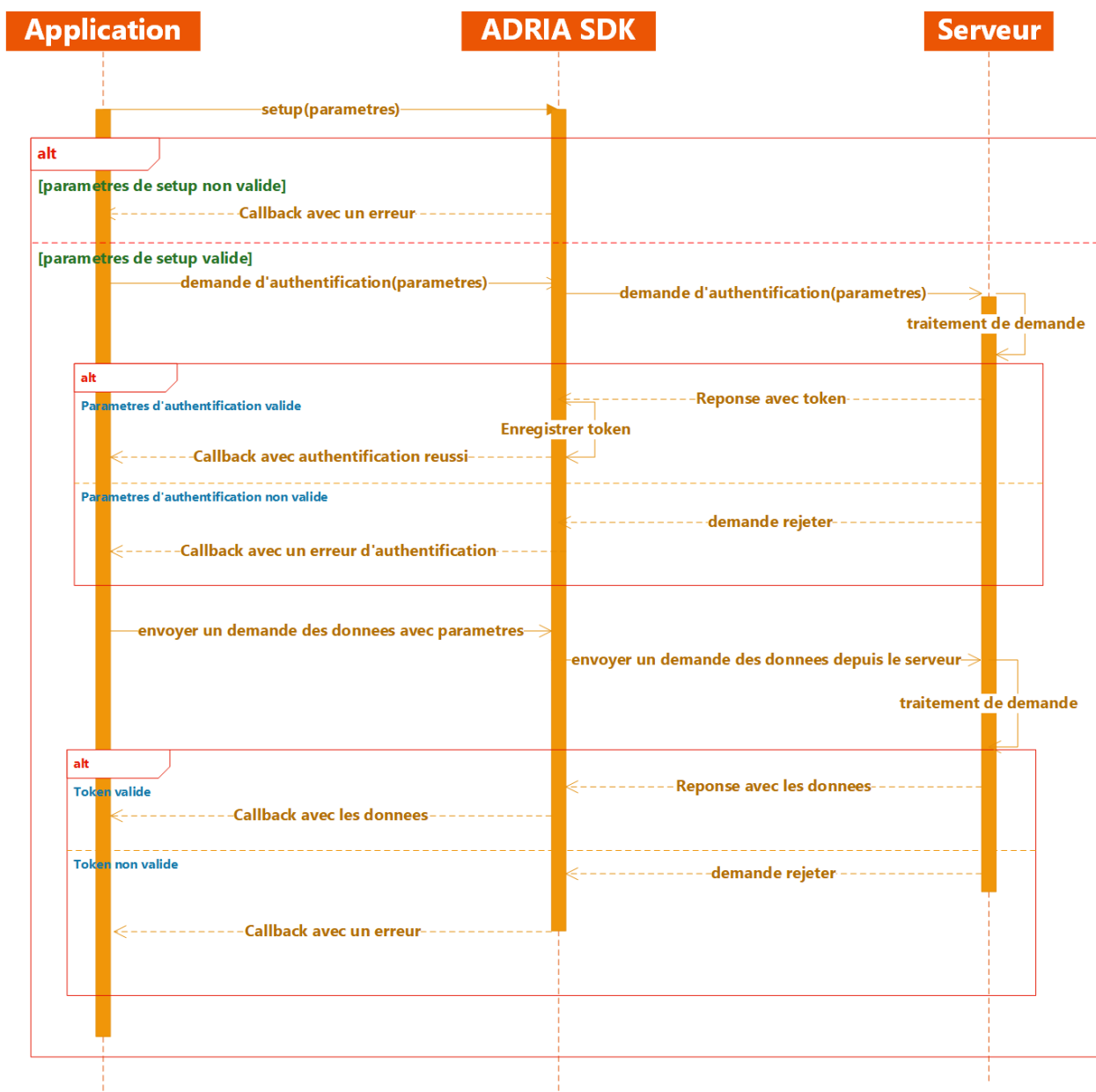


Figure 11 : Diagramme des séquences de package

Dans ce diagramme, on peut distinguer trois acteurs différents décrits comme suit :

- **L'application** : c'est l'application dans laquelle ce package sera intégré et qui va exploiter ses fonctionnalités. Elle doit le configurer lors de l'initialisation en fournissant les paramètres nécessaires.
- **Le package (Adria SDK)** : il représente notre projet que nous allons réaliser et qui permet la communication entre l'application et le serveur qui offre les webservice (API).
- **Le serveur** : dans le contexte de notre projet, il représente le serveur de client d'Adria qui se charge du traitement des demandes des applications clientes et de la gestion des données. Le serveur fournit les webservice (API) auxquels notre package va se connecter pour exécuter les différentes fonctionnalités demandées.

Pour que l'application puisse utiliser les services offerts par le package, elle doit suivre les étapes suivantes :

- 1- Configurer le package avec les paramètres nécessaires, notamment le lien de base.
- 2- Envoyer une demande d'authentification avec le login et le mot de passe à travers le package pour récupérer le Token d'authentification depuis le serveur cible, permettant ainsi au package de récupérer les données sans problème.
- 3- Exécuter les services offerts par le package pour communiquer avec le serveur une fois les étapes précédentes effectuées.

4. Conception des tâches effectuées

Cette partie présente les fonctionnalités que j'ai réalisées durant cette période de stage. Bien sûr, ces fonctionnalités nécessitent une authentification préalable entre le package et le serveur, comme il a été mentionné dans la partie précédente.

4.1. Récupération des mouvements d'un client :

Cette tâche implique la mise en place de la fonctionnalité de recherche de mouvements, qui offre aux utilisateurs du package la possibilité de rechercher des mouvements en utilisant des critères de recherche simples et intuitifs. Le diagramme ci-dessous illustre le processus de recherche en question :

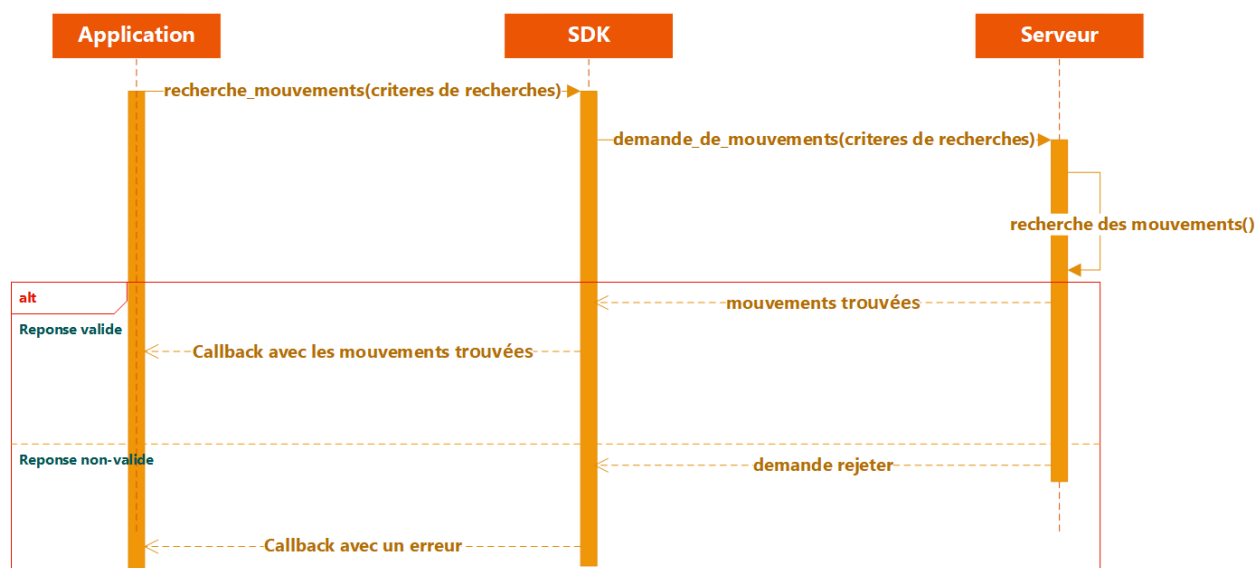


Figure 12 : Diagramme de séquences de recherche des mouvements

4.2. Récupération des messages envoyé et reçus d'un client :

Cette fonctionnalité permet aux utilisateurs de le package de récupérer tous les messages envoyés et reçus, en respectant les paramètres envoyés avec la requête au serveur de communication, de manière facile et simple en utilisant des modèles prédéfinis. Les deux images suivantes présentent le diagramme de séquence des opérations de récupération des messages envoyés et reçus respectivement :

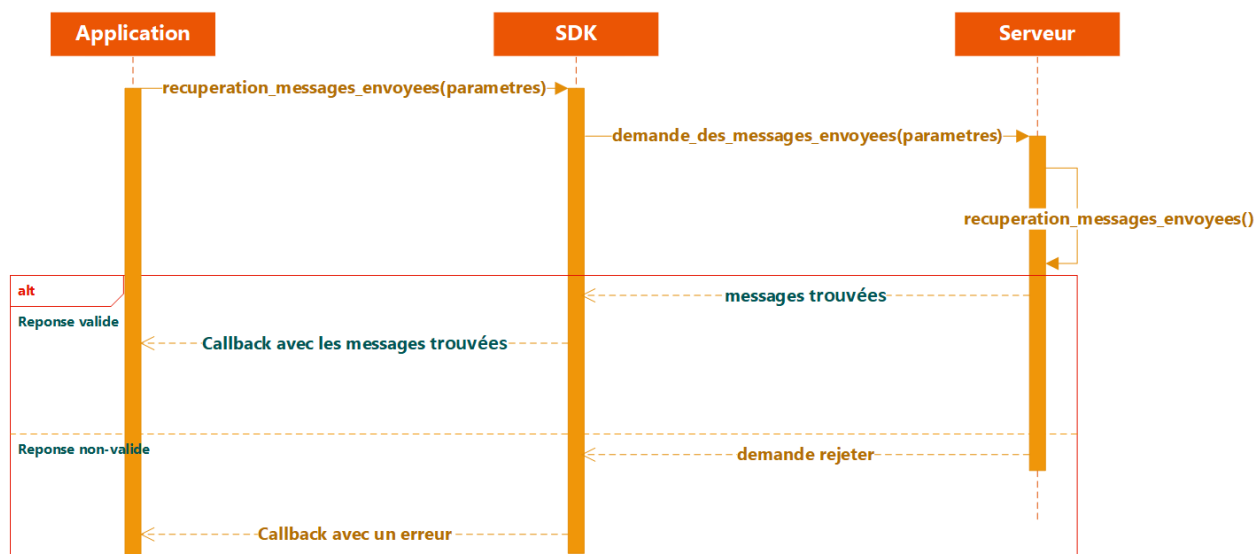


Figure 13 : Diagramme de séquences de récupération des messages envoyés

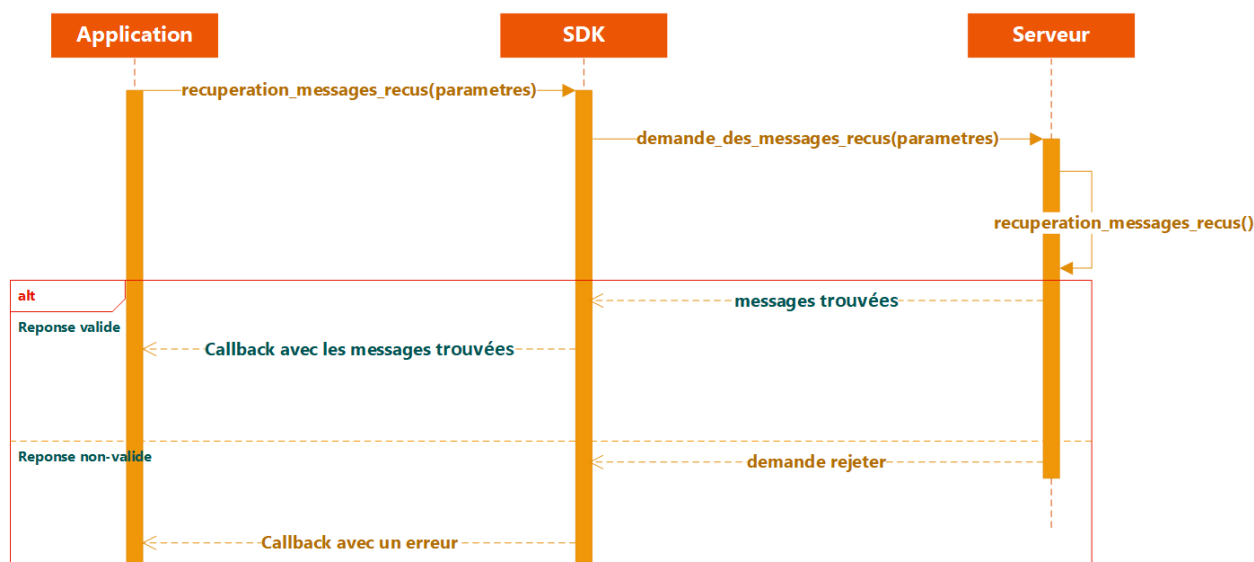


Figure 14 : Diagramme de séquences de récupération des messages reçus

4.3. Récupération des destinataires et les sujets de messagerie d'un client :

Cette partie permet de récupérer les informations des sujets de messagerie ainsi que la liste des destinataires, en respectant les paramètres envoyés avec la requête au serveur de communication. Les deux images suivantes présentent le diagramme de séquence des opérations de récupération des sujets de messagerie et des destinataires respectivement :

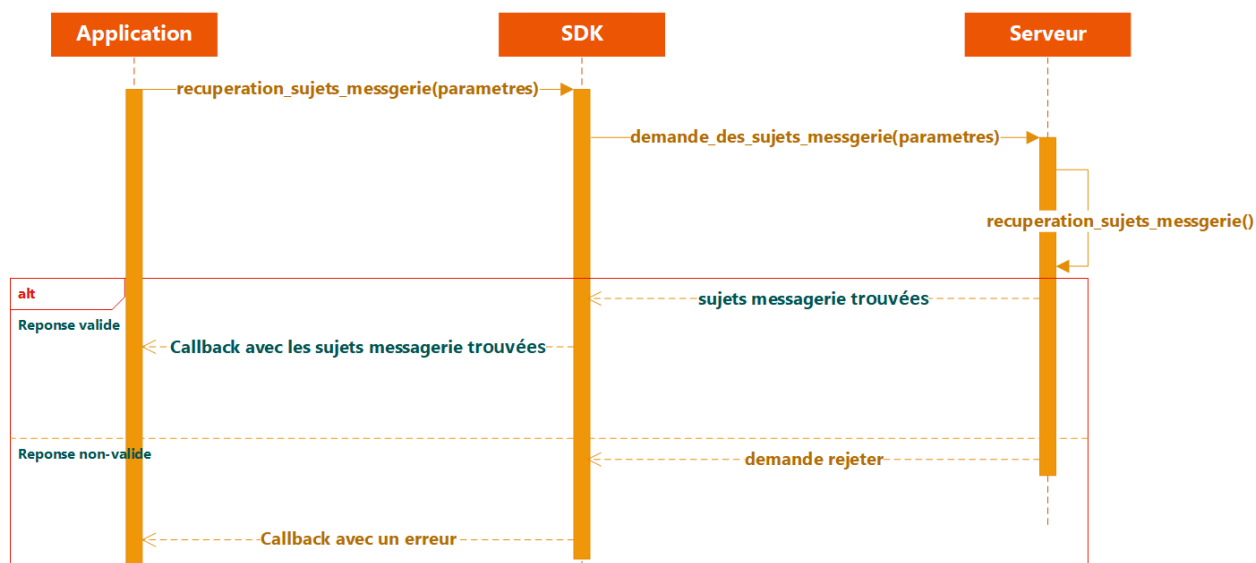


Figure 15 : Diagramme de séquences de récupération des sujets de messagerie

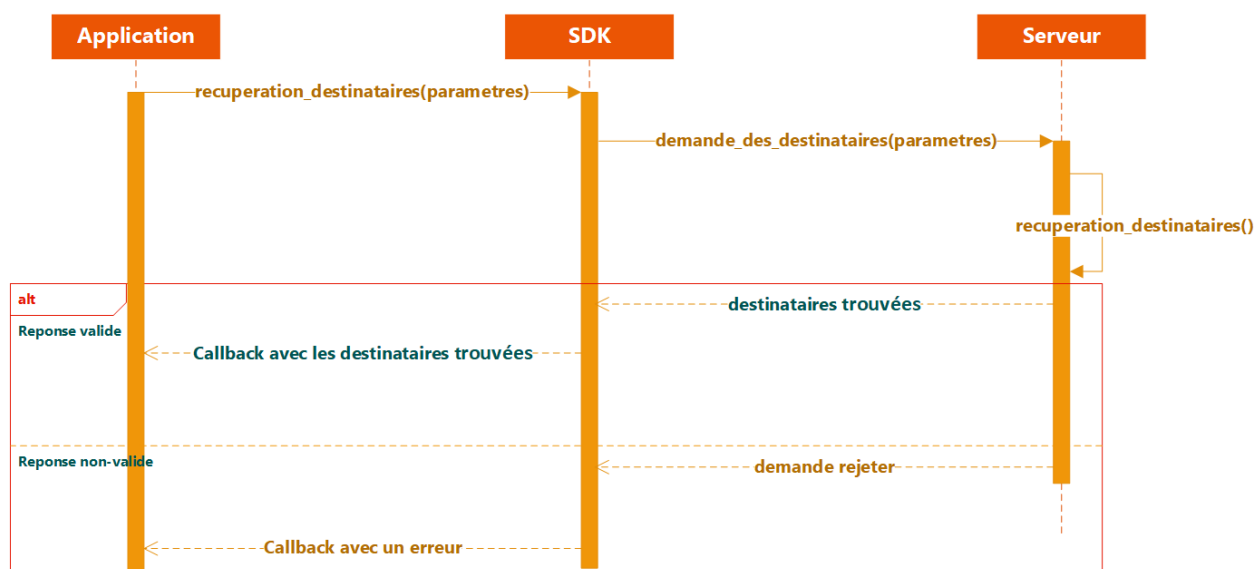


Figure 16 : Diagramme de séquences de récupération des destinataires

4.4. Enregistrement d'un nouveau message d'un client :

Cette fonctionnalité permet aux utilisateurs d'enregistrer des messages sur le serveur ainsi que les fichiers associés à ces messages. Le diagramme suivant présente le fonctionnement de cette action dans notre SDK :

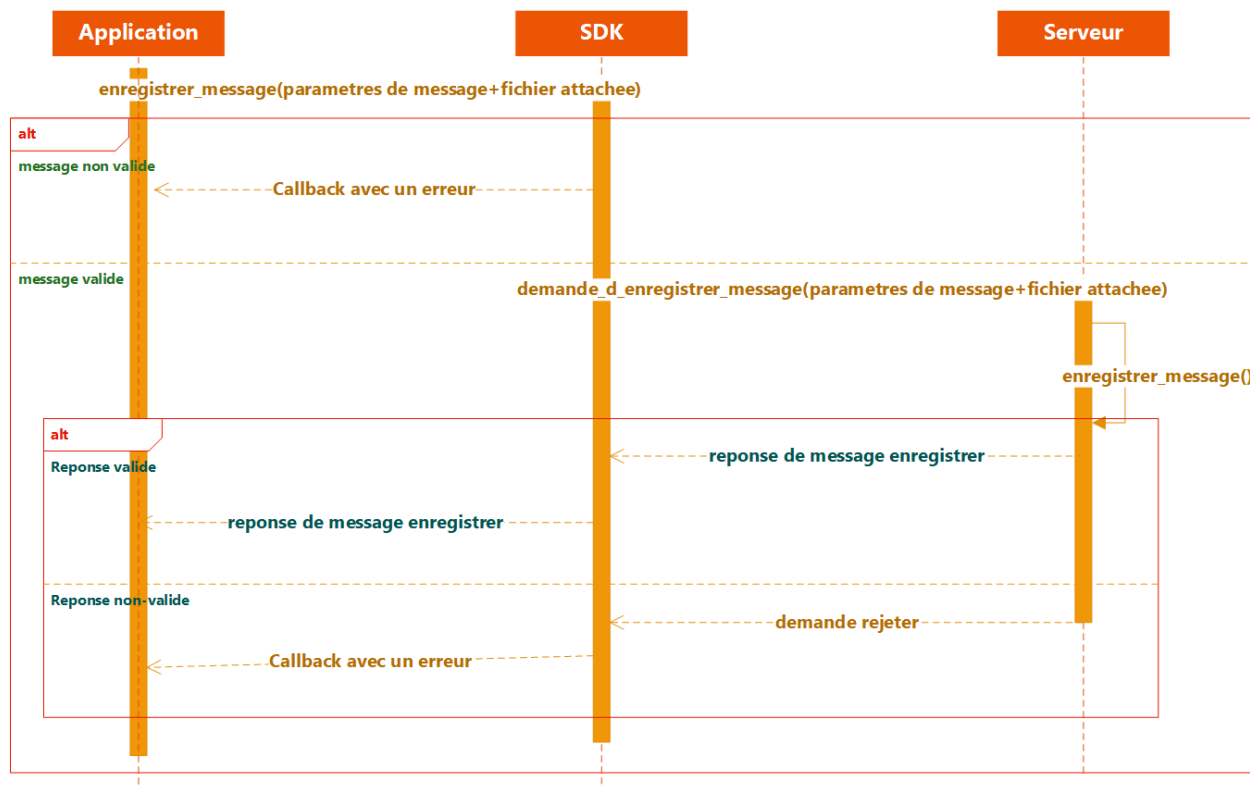


Figure 17 : Diagramme de séquences d'enregistrement des messages

4.5. Récupération et téléchargement des documents d'un client :

Ce service permet aux utilisateurs du package de récupérer la liste des documents d'un client donné et d'accéder au contenu d'un document spécifique en utilisant son identifiant. En outre, le package offre aux utilisateurs une fonctionnalité de téléchargement des documents sur leur appareil lorsqu'ils en font la demande. Les diagrammes de séquence suivants présentent le fonctionnement de ces fonctionnalités :

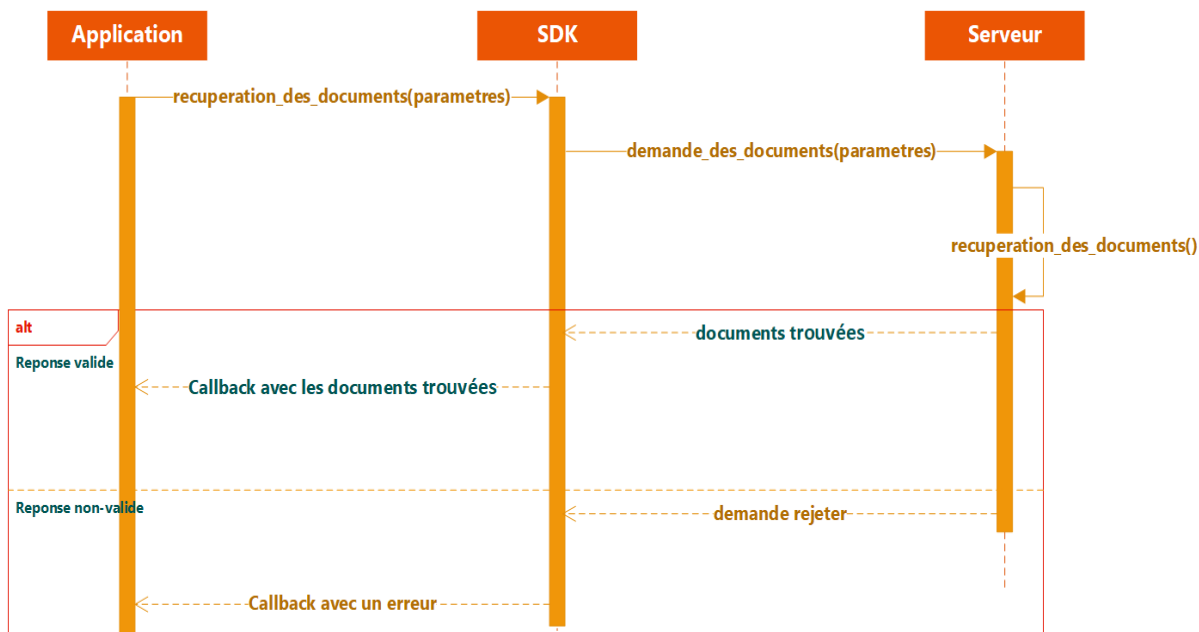


Figure 18 : Diagramme de séquences de récupération des documents

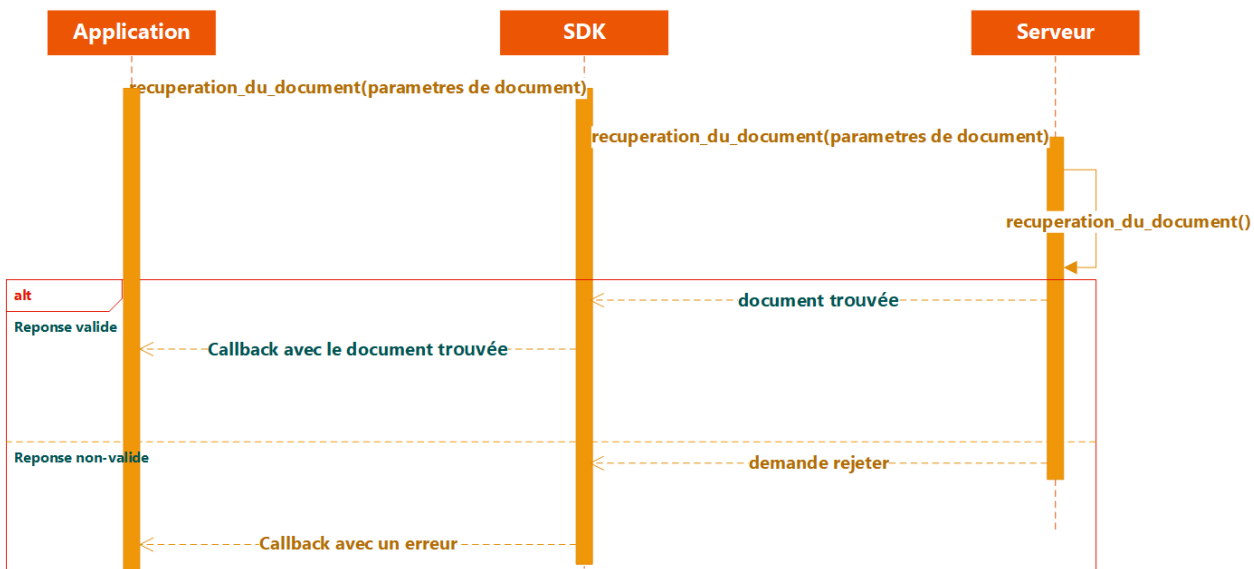


Figure 19 : Diagramme de séquences de récupération d'un document

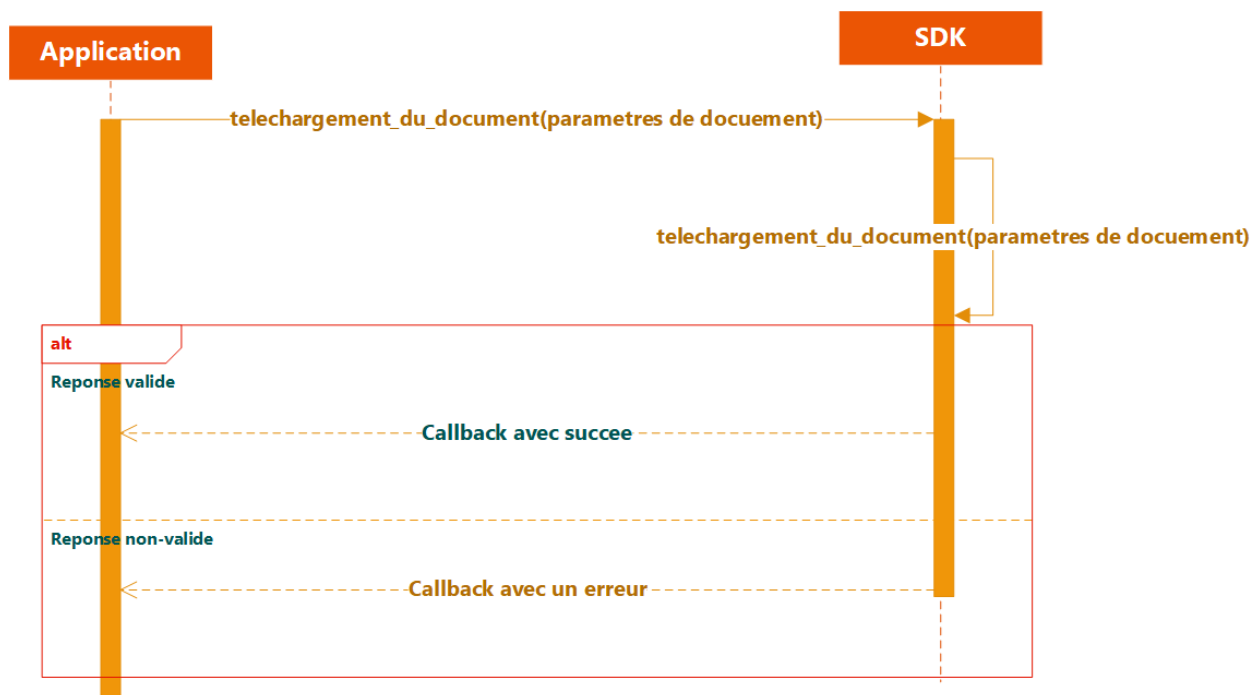


Figure 20 : Diagramme de séquences de téléchargement d'un document

5. Conception d'application de teste

Cette partie présente le diagramme des cas d'utilisation d'une petite application iOS qui a été développée pour deux raisons : la première est de pratiquer le design des applications iOS, et la seconde est de tester les fonctionnalités implémentées dans la section précédente. L'objectif est de s'assurer du bon fonctionnement de ces fonctionnalités et d'augmenter l'intégrité du package que nous sommes en train de développer.

Le diagramme des cas d'utilisation présenté dans cette partie illustre les différentes actions que peut effectuer l'utilisateur final de notre petite application iOS :

Adria app

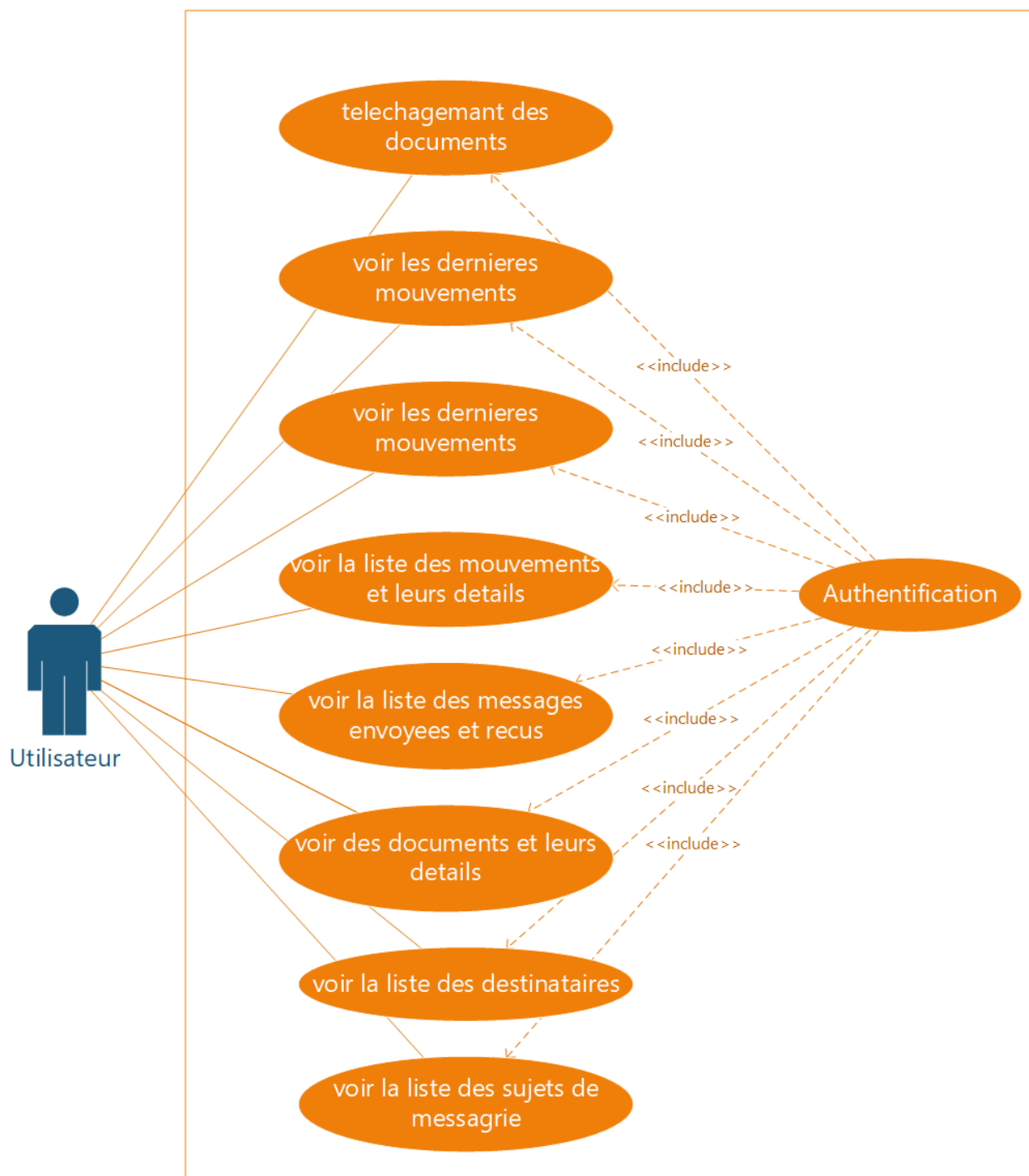


Figure 21 : Diagramme de cas d'utilisation de l'application iOS

6. Conclusion

En somme, la conception détaillée du projet est une étape essentielle qui contribue grandement à la réussite du projet. Elle permet de définir clairement les objectifs, de coordonner les tâches et d'assurer une qualité optimale du produit final.

Chapitre 4 : Etude technique du projet

1. Introduction
2. Architecture de projet
3. Langage et technologies de développement
4. Les tests unitaires
5. Outils et environnement de travail
6. Conclusion

1. Introduction

L'objectif de ce chapitre est de fournir une présentation détaillée des outils et technologies utilisés pour accomplir les tâches du projet. Nous commencerons par expliquer l'architecture mise en œuvre pour la création du package et ses interfaces de test. Ensuite, nous aborderons les outils et techniques de développement ainsi que l'environnement de travail utilisé. Enfin, nous discuterons des tests unitaires.

2. Architecture de projet

L'architecture revêt une importance cruciale, car elle détermine à la fois l'emplacement où l'application exécute sa fonctionnalité principale et la manière dont cette fonctionnalité interagit avec des éléments clés tels que la base de données et l'interface utilisateur. Dans ce chapitre, nous allons approfondir l'architecture "VIPER" ou "VIP", reconnue comme étant la plus performante dans le domaine du développement d'applications iOS. Nous examinerons en détail ses principes et son fonctionnement, mettant en évidence ses avantages et son utilisation dans notre projet.

2.1. Présentation de L'architecture VIPER :

VIPER est une architecture logicielle utilisée pour développer des applications iOS. Elle est conçue pour favoriser la séparation des responsabilités et améliorer la maintenabilité, la testabilité et la réutilisabilité du code.

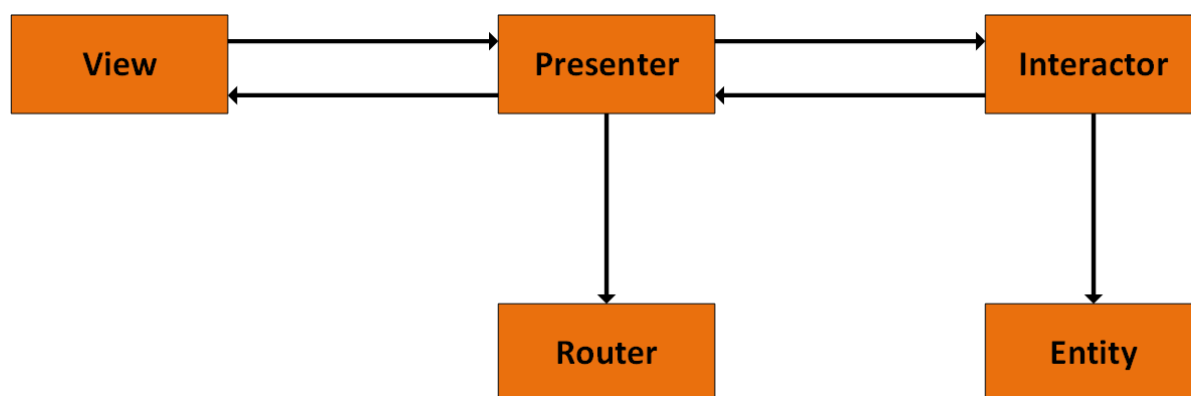


Figure 22 : Architecture VIPER

L'acronyme VIPER représente les différentes composantes de l'architecture :

- **View** : La couche de présentation qui affiche l'interface utilisateur. Elle gère les interactions de l'utilisateur et transmet les actions à la couche du présentateur (Presenter).
- **Interactor** : La couche métier qui contient la logique de l'application. Elle effectue les opérations, récupère les données et les traite.
- **Presenter** : La couche de présentation intermédiaire entre la vue (View) et l'interacteur (Interactor). Elle reçoit les événements de la vue et communique avec l'interacteur pour obtenir les données nécessaires à l'affichage.
- **Entity** : La couche modèle qui représente les données de l'application. Elle peut inclure des objets, des structures ou des classes qui encapsulent les données et les fonctionnalités.
- **Router** : La couche de navigation responsable de la gestion de la transition entre les différentes vues.

L'architecture VIPER favorise la dépendance unidirectionnelle entre les différentes couches, ce qui facilite la compréhension et la modification du code. Chaque couche a une responsabilité claire et les communications entre les couches se font par des interfaces bien définies.

De plus, VIPER facilite les tests unitaires en permettant l'isolation des différentes couches pour effectuer des tests individuels. Cela contribue à une meilleure qualité et fiabilité du code.

En résumé, l'architecture VIPER est une approche organisée pour le développement d'applications iOS, qui encourage la séparation des responsabilités et la modularité du code. Elle offre des avantages tels que la maintenabilité, la testabilité et la réutilisabilité, ce qui en fait un choix populaire parmi les développeurs iOS.

2.2. Implémentation de l'architecture VIPER dans le package :

Dans notre projet, nous utilisons l'architecture VIPER de manière légèrement différente pour mieux organiser le travail réalisé par toute l'équipe. Les images ci-dessous montrent l'architecture de packages utilisée.

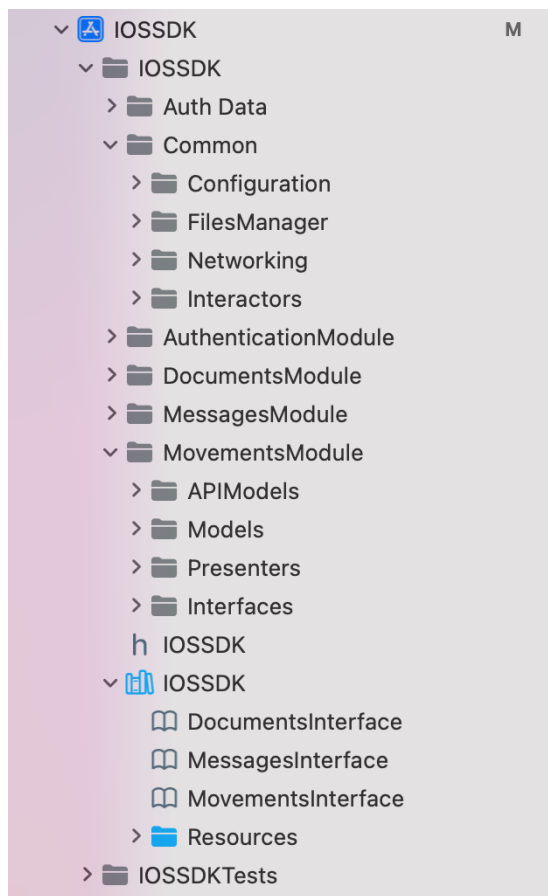


Figure 23 : Architecture de package

Application

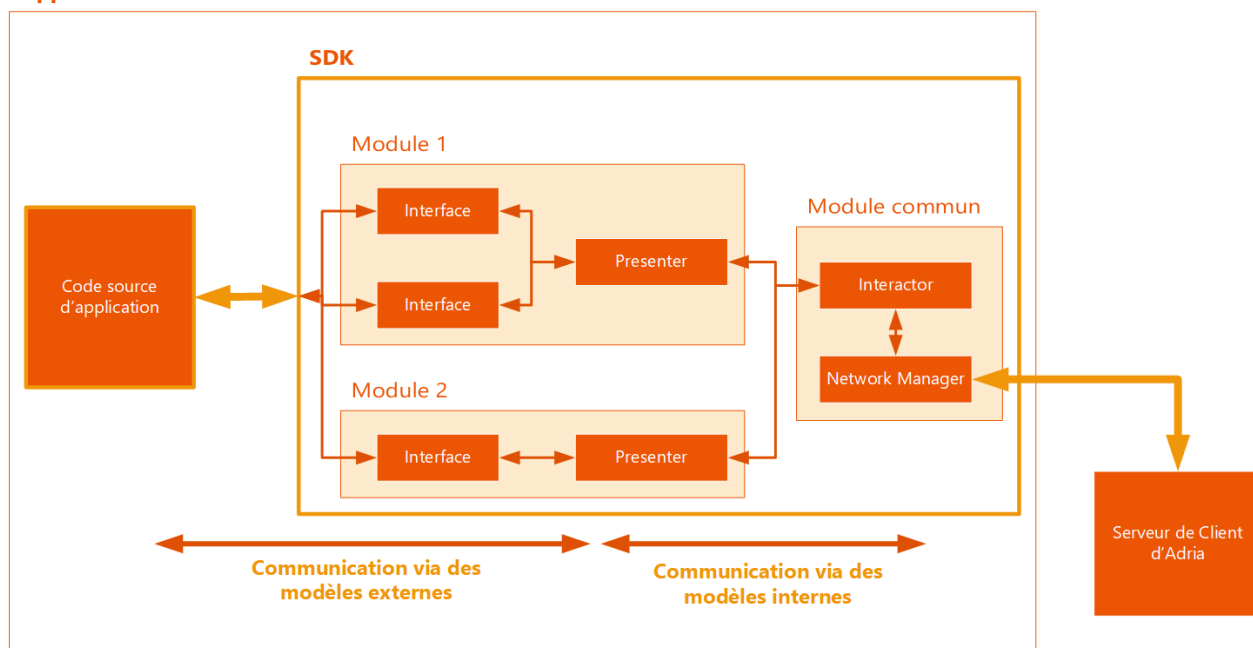


Figure 24 : Architecture de package détaillée

Pour faciliter le travail en équipe, nous avons adopté une architecture modulaire pour notre package. Chaque membre de l'équipe travaille sur un module spécifique, qui est décrit comme suit :

Chaque module comprend une interface qui représente la "View" dans l'architecture VIPER. Cette interface offre différentes fonctionnalités aux utilisateurs et communique avec le "Présentater" pour effectuer les traitements nécessaires. Chaque "Présentater" du module interagit avec le Module commun pour récupérer les données dont il a besoin. Le Module commun se compose de deux éléments principaux : l'Interactor, qui est responsable de la gestion des différentes demandes des "Présentaters", et le "NetworkManager", qui permet de lancer les requêtes HTTP et de communiquer avec le serveur.

Cette organisation modulaire offre plusieurs avantages. Elle permet une séparation claire des responsabilités entre les modules, favorise une meilleure collaboration au sein de l'équipe et facilite le développement parallèle. Chaque module peut être développé indépendamment, ce qui améliore l'efficacité du développement et permet de se concentrer sur des fonctionnalités spécifiques. Le Module commun assure la cohérence et la centralisation des fonctionnalités partagées.

2.3. Implémentation de l'architecture VIPER dans l'application de teste :

Pour le cas de cette application, nous avons implémenté l'architecture VIP, qui est une version simplifiée de l'architecture VIPER sans utiliser les composants "Entity" et "Router".

L'architecture VIP est généralement utilisée dans des projets plus petits ou lorsque l'architecture plus simple est suffisante. Elle sépare les composants en trois parties principales :

- **View** : qui gère l'interface utilisateur et les interactions utilisateur.
- **Interactor** : qui contient la logique métier et interagit avec les sources de données.
- **Presenter** : qui fait office de médiateur entre View et Interactor.

Nous avons opté pour cette architecture, car cette application est spécifiquement conçue pour servir de projet principal dans la création d'un package. L'architecture adoptée est organisée de la manière présentée dans les images suivantes :

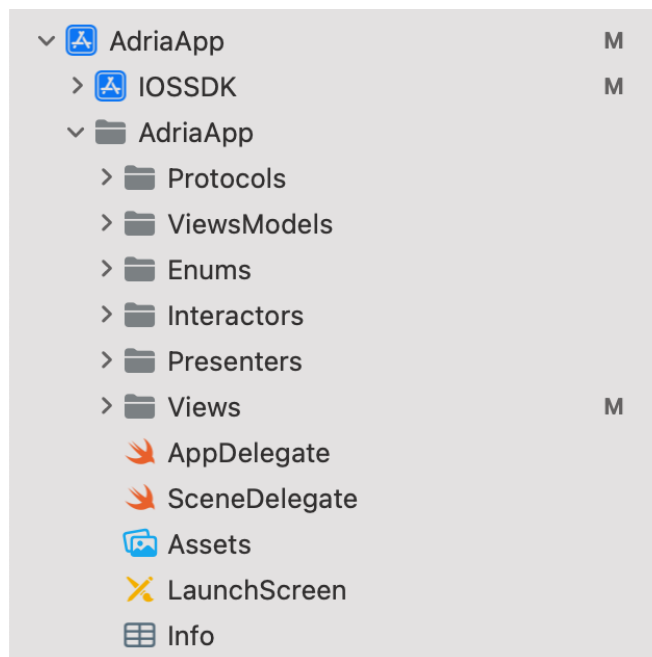


Figure 25 : Architecture d'application

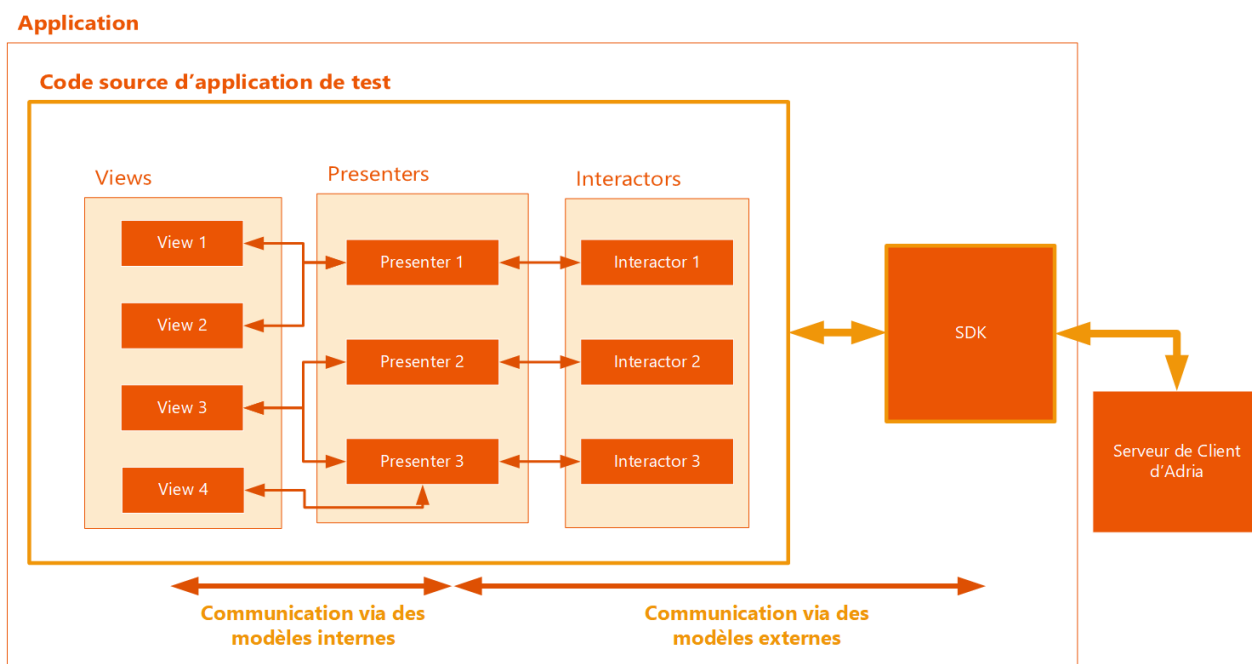


Figure 26 : Architecture d'application détaillée

3. Langage et technologies de développement

Dans cette section, je vais vous présenter le langage de programmation et les différentes technologies que nous avons utilisés pour mener à bien notre projet tout au long de mon stage.

3.1 la langage Swift :



Figure 27 : Logo de Swift

Swift est un langage de programmation moderne développé par Apple. Introduit en 2014, il est devenu le langage principal pour le développement d'applications iOS, macOS, watchOS et tvOS. Avec une syntaxe concise et expressive, Swift facilite la lecture et l'écriture du code. Il prend en charge plusieurs paradigmes de programmation et offre des fonctionnalités avancées telles que les génériques, les "closures" et les tuples. La sécurité et la stabilité du code sont prioritaires grâce à des concepts tels que le typage fort et la gestion automatique de la mémoire. Swift s'intègre facilement avec les Framework et les bibliothèques d'Apple et est continuellement mis à jour avec de nouvelles fonctionnalités.

En conclusion, Swift est un langage puissant, convivial et performant pour le développement d'applications sur les plateformes Apple.

3.2 la bibliothèque Wormholy :

Afin d'améliorer la qualité et l'efficacité du développement, nous avons choisi d'intégrer la bibliothèque Wormholy comme un outil d'interception de réseau lors des tests du package dans notre application de test. Cette décision nous permet de capturer et d'analyser les appels réseau effectués par le package, ce qui nous aide à identifier les problèmes, à détecter les erreurs et à optimiser les performances.



Figure 28 : Logo de Wormholy

La bibliothèque Wormhole est un outil puissant et pratique pour le débogage et la surveillance des appels réseau dans les applications iOS. Elle permet aux développeurs de visualiser et d'analyser les requêtes HTTP/S effectuées par l'application en temps réel.

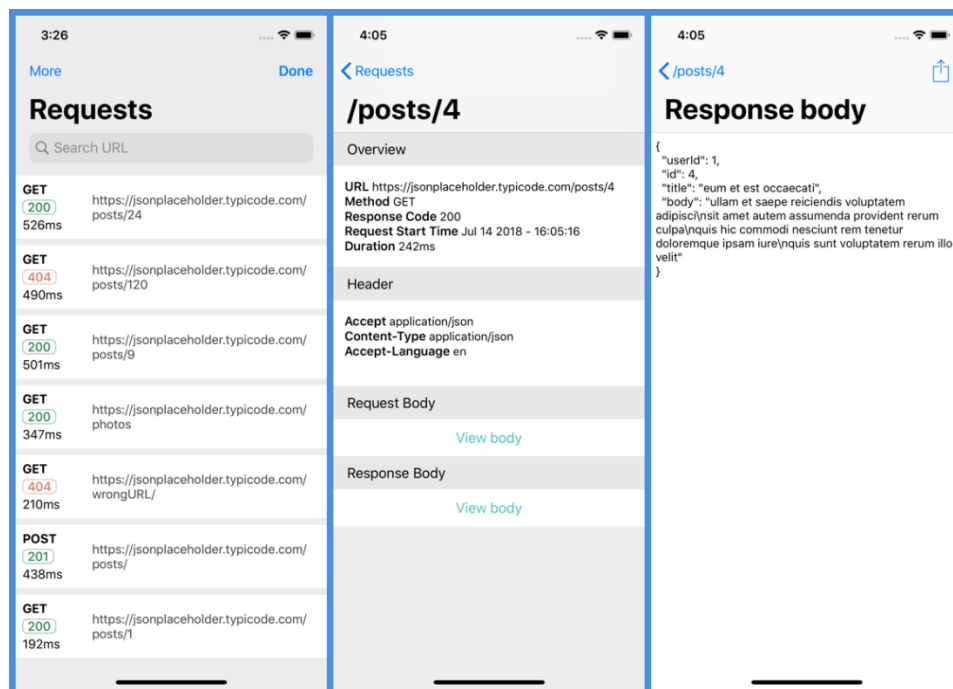


Figure 29 : Les interface offerte par Wormhole

Wormhole offre une interface conviviale qui affiche les détails des requêtes, y compris les en-têtes, les paramètres, les réponses et les temps de réponse. Elle permet également de visualiser les réponses au format JSON de manière organisée et facilement navigable.

Grâce à Wormhole, les développeurs peuvent inspecter les appels réseau, détecter les erreurs, surveiller les performances et déboguer les problèmes liés aux requêtes réseau. Cela peut être particulièrement utile lors du développement et du test d'applications qui dépendent fortement des services web.

4. Outils et environnement de travail

Cette section fournit une description détaillée des outils utilisés et de l'environnement de travail pendant la période de stage.

4.1. Xcode :



Figure 30 : Logo de Xcode

Xcode est l'environnement de développement intégré (IDE) d'Apple, spécialement conçu pour le développement d'applications pour les plateformes iOS, macOS, watchOS et tvOS. Il est largement utilisé par les développeurs pour créer des applications pour les appareils Apple.

Xcode offre une suite complète d'outils et de fonctionnalités qui facilitent le processus de développement. Il comprend un éditeur de code source puissant avec une coloration syntaxique, une complétion automatique et des fonctionnalités de refactorisation. Il propose également un débogueur intégré pour identifier et corriger les erreurs, ainsi qu'un gestionnaire de version pour suivre les modifications du code. Une autre caractéristique clé de Xcode est son interface de conception graphique, appelée Interface Builder, qui permet de créer des interfaces utilisateur visuellement attrayantes en utilisant des éléments graphiques prédéfinis et en les reliant à du code source. Il prend également en charge la compilation, le test et le déploiement d'applications sur les appareils Apple. Il offre des outils de profilage pour optimiser les performances de l'application et des outils de localisation pour adapter l'application à différentes langues et régions.

En résumé, Xcode est un environnement de développement puissant et complet pour créer des applications pour les appareils Apple. Il simplifie le processus de développement en fournissant des outils et des fonctionnalités essentiels, et il est largement utilisé par les développeurs pour sa compatibilité avec les plateformes Apple.

4.2. Git :

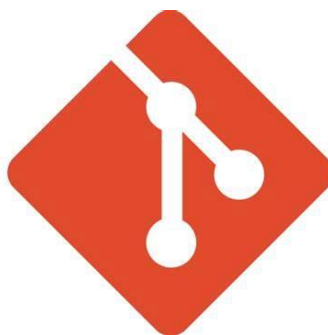


Figure 31 : Logo de Git

Git est un système de contrôle de version distribué largement utilisé dans le développement de logiciels. Il permet de suivre les modifications apportées aux fichiers et aux projets, de gérer les branches de développement, de collaborer efficacement et de revenir à des versions antérieures si nécessaire. Git offre une gestion flexible des versions, facilite le travail en équipe et permet une intégration continue fluide. Il est populaire pour sa rapidité, sa robustesse et sa capacité à gérer des projets de toutes tailles.

4.3. BitBucket :



Figure 32 : Logo de Bitbucket

Bitbucket est une plateforme de gestion de code source et de collaboration pour les projets de développement logiciel. Elle permet aux équipes de stocker, gérer et partager leurs dépôts Git ou Mercurial de manière sécurisée. Bitbucket offre des fonctionnalités telles que le contrôle de version, la gestion des branches, les demandes de fusion, les intégrations avec d'autres outils de développement et la possibilité de collaborer en temps réel avec les membres de l'équipe. Il permet également d'héberger des dépôts de code privés et publics, offrant ainsi une flexibilité pour les projets personnels et professionnels.

4.4. Postman :



Figure 33 : Logo de Postman

Postman est un outil populaire de développement d'API qui permet aux développeurs de tester, documenter et collaborer sur leurs API. Il offre une interface conviviale où les utilisateurs peuvent envoyer des requêtes HTTP vers des endpoints spécifiques, tester les réponses et analyser les résultats. Postman prend en charge diverses méthodes de requête, des paramètres, des en-têtes personnalisés, des tests automatisés, la génération de documentation API et bien plus encore. Il facilite également la collaboration entre les membres de l'équipe grâce à la possibilité de partager des collections d'API et de travailler ensemble sur des tests et des scénarios. Postman est utilisé par de nombreux développeurs et équipes pour simplifier et accélérer le processus de développement et de test des API.

5. Les tests unitaires

Les tests unitaires sont des outils puissants qui garantissent la qualité du code en vérifiant le bon fonctionnement de chaque unité de code individuelle. Leur utilisation joue un rôle crucial dans l'assurance de la robustesse et de la fiabilité de nos applications.

5.1. Objectif des tests unitaires :

Les tests unitaires ont pour objectif de valider le comportement précis des unités de code, telles que les fonctions, les méthodes ou les classes. Ils permettent de détecter les erreurs dès les premières étapes du développement, facilitant ainsi leur correction et réduisant les risques de bogues dans le produit final.

5.2. Avantages des tests unitaires :

- ✓ **Détection précoce des erreurs** : Les tests unitaires nous aident à identifier les erreurs dès leur apparition, permettant une correction rapide avant qu'elles ne se propagent dans l'ensemble du système.

- ✓ **Facilitation de la collaboration** : Les tests unitaires définissent des spécifications claires pour chaque unité de code, facilitant ainsi la collaboration entre les membres de l'équipe de développement.
- ✓ **Amélioration de la maintenabilité du code** : Les tests unitaires fournissent une base solide pour effectuer des modifications et des mises à jour ultérieures, garantissant que le code reste fonctionnel et évolutif.
- ✓ **Réduction des risques lors des modifications ultérieures** : Les tests unitaires offrent une confiance accrue lors de l'introduction de nouvelles fonctionnalités ou de modifications du code existant, en permettant de vérifier rapidement si les fonctionnalités existantes sont toujours opérationnelles.

5.3. XCTest framework :



Figure 34 : Logo de XCTest

Afin de valider les fonctionnalités de notre package, nous utilisons XCTest, un Framework de tests unitaires fourni par Apple dans l'environnement de développement Xcode. XCTest est spécifiquement conçu pour tester les applications iOS, macOS, tvOS et watchOS. Ce Framework offre des fonctionnalités complètes et des APIs permettant aux développeurs d'écrire et d'exécuter des tests unitaires de manière efficace. Parmi ces caractéristiques, on peut citer :

- ✓ **Création de cas de tests** : XCTest permet aux développeurs de créer des cas de tests pour chaque unité de code à tester, offrant une structure modulaire pour organiser les tests.
- ✓ **Assertions** : XCTest fournit des assertions prédéfinies pour valider les résultats attendus lors de l'exécution des tests, telles que les égalités, les inégalités, les conditions booléennes, etc.

- ✓ **Configuration des tests** : XCTest permet de configurer des opérations de configuration et de nettoyage avant et après l'exécution des tests, garantissant un environnement cohérent pour les tests.
- ✓ **Tests asynchrones** : XCTest prend en charge les tests asynchrones, permettant de tester des opérations qui s'exécutent de manière asynchrone, comme les appels réseau.
- ✓ **Mesures de performances** : XCTest permet de mesurer les performances des unités de code à tester, permettant ainsi d'évaluer les performances des fonctions ou des portions spécifiques de code.
- ✓ **Intégration à Xcode** : XCTest est intégré à l'environnement de développement Xcode, offrant une interface conviviale pour exécuter les tests, afficher les résultats et faciliter la gestion des tests unitaires.

En utilisant XCTest, nous pouvons effectuer des tests approfondis et fiables sur notre package, nous assurant ainsi de sa qualité et de son bon fonctionnement dans l'environnement iOS.

6. Conclusion

En conclusion de ce chapitre, nous avons passé en revue l'environnement de travail que nous avons utilisé pour mener à bien notre projet. Nous avons présenté en détail les différentes technologies que nous avons mises en place pour le développement du package. Ces choix technologiques ont été essentiels pour assurer une mise en œuvre efficace de notre projet.

Chapitre 5 : Mis en œuvre du projet

1. Introduction
2. Rapport de l'ensemble des réalisations dans le projet
3. Conclusion

1. introduction :

Ce chapitre met en évidence les réalisations réalisées dans notre projet, notamment le développement des différentes interfaces d'application de test. Nous avons réussi à réaliser plusieurs avancées significatives, en implémentant des fonctionnalités clés et en mettant en place des interfaces de test pour garantir le bon fonctionnement de notre package.

2. Rapport de l'ensemble des réalisations dans le projet :

Dans cette section, je présenterai les différentes interfaces que j'ai développées dans notre package. Chacune de ces interfaces représente une fonctionnalité spécifique implémentée dans notre projet. Les démonstrations qui suivront permettront d'illustrer comment les utilisateurs peuvent intégrer ces interfaces dans leurs propres projets. De plus, je fournirai également un exemple d'utilisation des interfaces dans le contexte de tests d'application.

2.1. Authentification :

Avant qu'un développeur puisse utiliser et bénéficier des fonctionnalités implémentées dans notre SDK, il doit d'abord créer une page d'authentification pour les utilisateurs finaux de leur application. Cette page permettra aux utilisateurs de se connecter en utilisant un nom d'utilisateur et un mot de passe. Dans notre application de test voici cette interface :



Figure 35 : Page d'authentification

Une fois l'authentification effectuée, l'utilisateur sera redirigé vers le tableau de bord de l'application, tel que présenté dans les deux images suivantes :



Figure 36 : Page d'accueil

2.2. Récupération des mouvements :

La fonctionnalité de Récupération des mouvements est un service qui permet aux développeurs d'obtenir tous les mouvements d'un client à partir d'un serveur bancaire spécifique. Elle offre un accès essentiel aux informations de transaction d'un client, pouvant être utilisées dans une application. L'image ci-dessous illustre la démonstration de la page affichant les mouvements récupérés via cette interface :

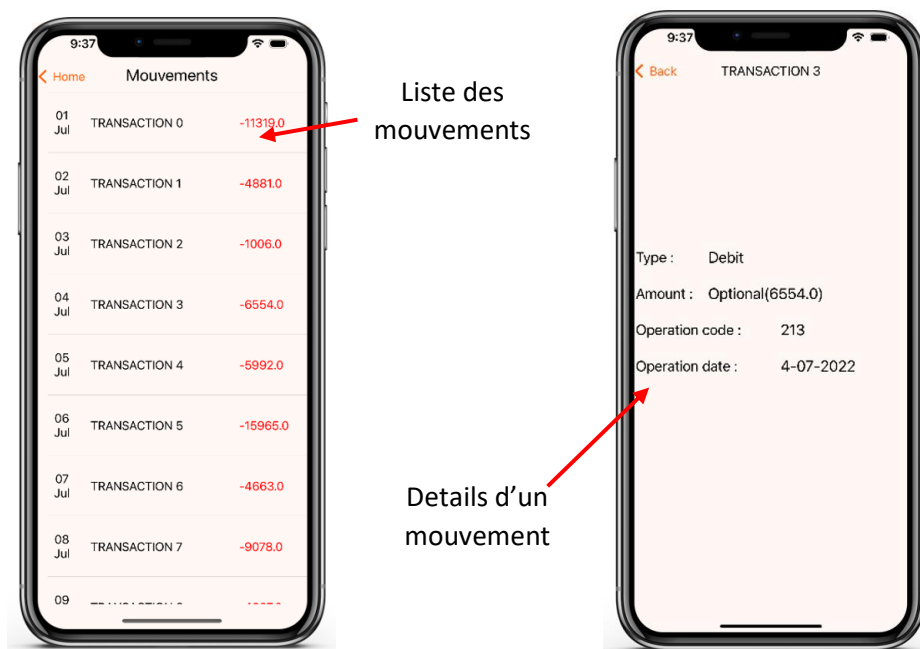


Figure 37 : Page des mouvements

2.3. Récupération des messages envoyées et reçus :

Cette fonctionnalité permet aux développeurs d'accéder et de récupérer tous les messages émis et reçus par un client à partir d'un serveur de messagerie spécifique. Cela offre la possibilité d'intégrer ces messages dans une application afin de faciliter la communication et la gestion des échanges pour les utilisateurs. La capture d'écran ci-dessous présente un exemple de la page affichant les messages récupérés grâce à cette interface :

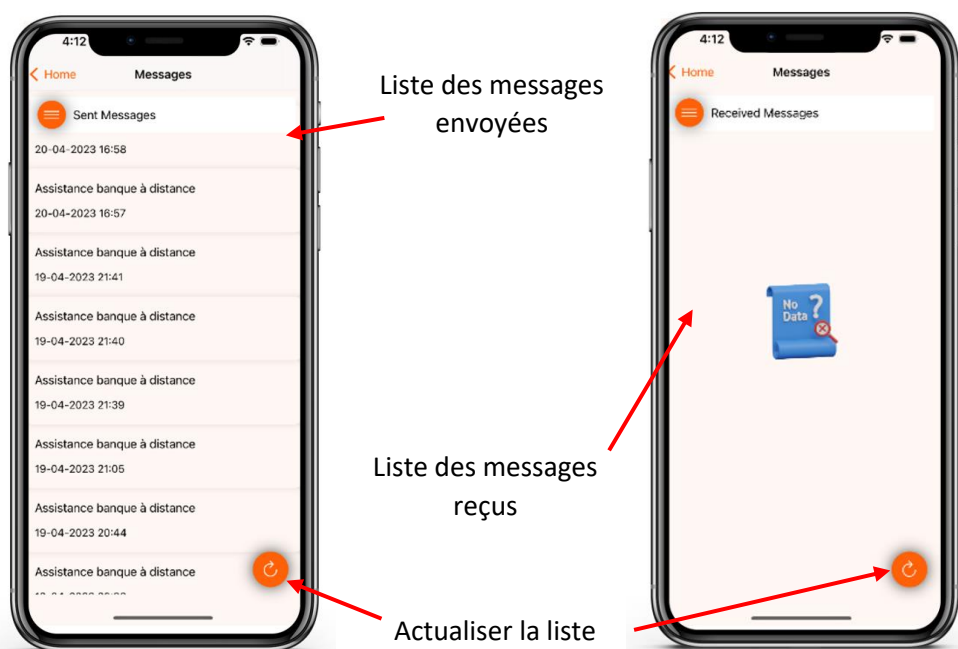


Figure 38 : Page des messages envoyées et reçus

2.4. Récupération des destinataires et sujets de messages :

Notre package offre également la fonctionnalité de récupération de la liste des destinataires avec lesquels l'utilisateur final peut être en contact, ainsi que la liste des sujets de messages reçus ou envoyés. L'image ci-dessous illustre l'intégration de ces deux fonctionnalités dans notre application :



Figure 39 : Page de liste des destinataires et sujets de messages

2.5. Enregistrement d'un nouveau message :

Notre package intègre également la fonctionnalité d'envoi de messages à un destinataire spécifique, avec la possibilité d'ajouter un fichier en pièce jointe. L'interface ci-dessous montre comment cette fonctionnalité est implémentée :

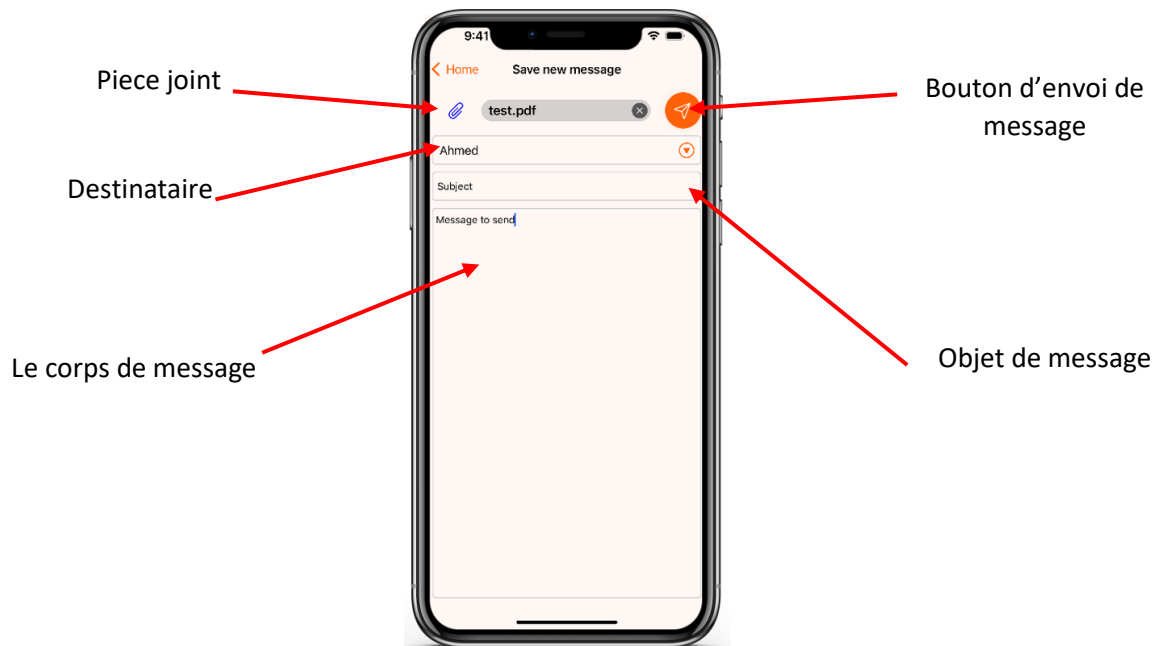


Figure 40 : Page d'envoi d'un message

2.5. Récupération de la liste des documents d'un client :

Cette fonctionnalité permet aux utilisateurs d'accéder à la liste des documents associés à un client spécifique, lesquels sont généralement au format PDF et renferment des informations essentielles. Elle offre une gestion simplifiée et une consultation aisée de ces documents pour faciliter l'accès aux informations pertinentes. De plus, nous illustrons ci-dessous cette fonctionnalité à travers les deux images suivantes :

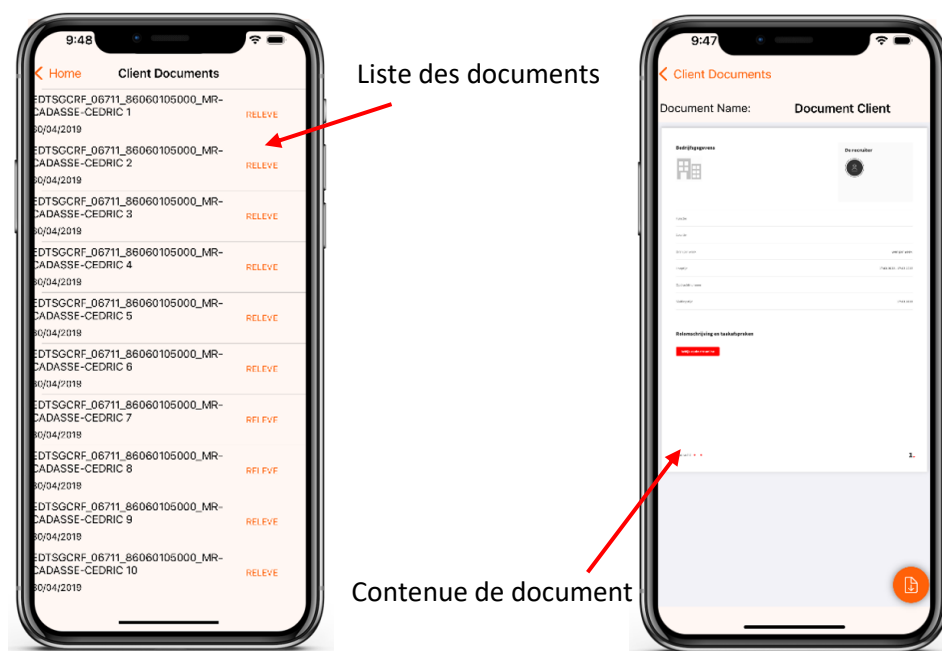


Figure 41 : Page de liste des documents et page de contenu du document

2.6. Téléchargement d'un document :

Nous avons intégré dans ce package une fonctionnalité qui permet de télécharger un document sur l'appareil en cours d'utilisation. Dans notre application, nous avons ajouté cette fonctionnalité sous la forme d'un bouton qui s'affiche sur la page d'affichage des documents, comme le montre l'image ci-dessous :

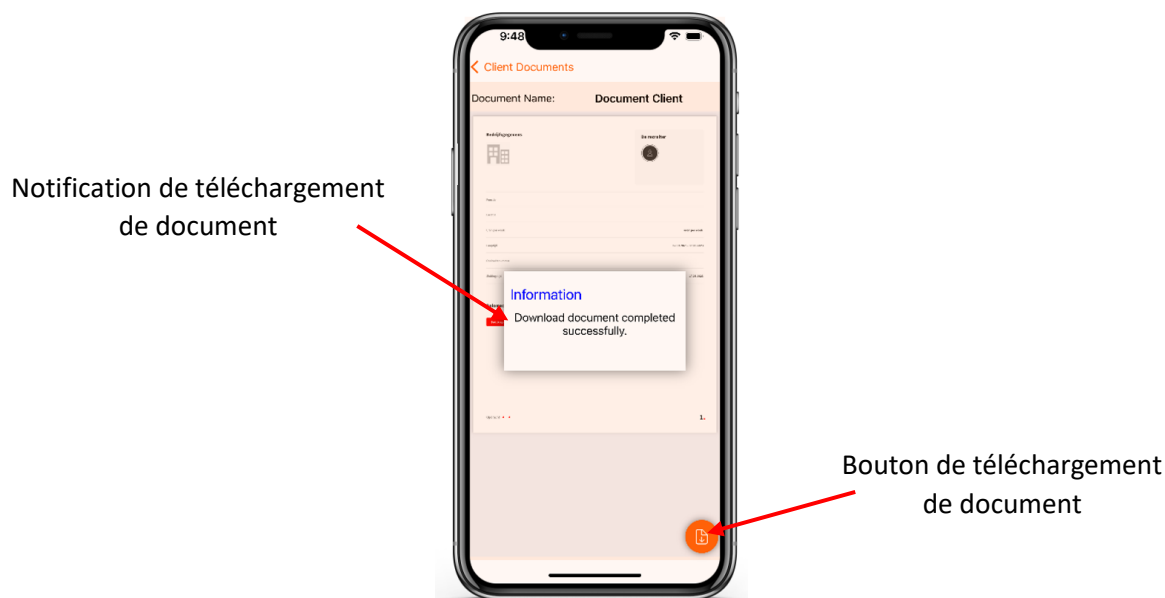


Figure 42 : Page de contenu de document avec le bouton de téléchargement

4. conclusion :

En conclusion, les réalisations accomplies dans notre projet, notamment le développement des interfaces et des fonctionnalités dans ce SDK, témoignent de notre engagement à fournir des solutions de qualité. Nous avons réalisé des progrès significatifs dans notre objectif de créer un package robuste et fonctionnel. Cependant, il reste encore du travail à faire pour atteindre pleinement nos objectifs. Nous sommes confiants dans notre capacité à surmonter les défis restants et à mener à bien ce projet avec succès.

Conclusion générale :

En conclusion, mon stage de fin d'études dans le développement d'applications iOS a été une expérience enrichissante qui m'a permis d'acquérir de nouvelles compétences en utilisant le langage Swift et en explorant les outils de développement tels que Xcode. J'ai pu mettre en pratique mes connaissances théoriques, renforcer mes compétences techniques et me familiariser avec les meilleures pratiques de l'industrie.

Je suis reconnaissant envers entreprise Adria B&T de m'avoir offert cette opportunité de travailler sur des projets concrets et de développer ma capacité à résoudre des problèmes techniques. J'ai également appris à collaborer efficacement en équipe, ce qui est essentiel dans le domaine du développement d'applications.

Je suis confiant que les compétences acquises lors de ce stage me serviront dans ma future carrière. Je suis déterminé à continuer à me former et à me tenir informé des dernières avancées technologiques pour rester compétent et contribuer à des projets innovants dans le domaine de développement iOS. Je suis enthousiaste à l'idée de relever de nouveaux défis et d'apporter une valeur ajoutée à mon futur employeur grâce à mon expérience de stage et à mes connaissances actualisées.

Webographie

1. Site web officiel de l'entreprise Adria B&T :

- [Adria Business & Technology - Online Banking Solutions \(adria-bt.com\)](http://adria-bt.com)

2. Documentation technique du projet ou de la technologie utilisée :

- [Alamofire/Alamofire: Elegant HTTP Networking in Swift \(github.com\)](https://github.com/Alamofire)
- [Swift - Apple Developer](https://developer.apple.com/swift/)
- [Wormhole on CocoaPods.org](https://www.cocoapods.org/)
- [Xcode 14 - Apple Developer](https://developer.apple.com/xcode/)
- [Git \(git-scm.com\)](https://git-scm.com/)
- https://bitbucket.org
- [Postman API Platform | Sign Up for Free](https://www.postman.com/)
- [XCTest | Apple Developer Documentation](https://developer.apple.com/documentation/xctest/)