

Conception détaillée et développement de l'application

Digiit

28 mars 2023



Mathias Pinto - Adrien Mussat - Cédric Brzyski - Hugo Moutinho
Célian Aldehuelo

Sommaire

1	Introduction	3
1.1	Secteur applicatif	3
2	Proposition de Projet	4
2.1	Besoins et étude de marché	4
3	Activités/Interrelations	5
3.1	Slides	8
4	Interfaces	9
4.1	Les différentes pages de l'application	10
4.2	Choix de l'UI	16
5	Services	16
5.1	API	16
5.1.1	API	17
5.1.2	Déploiement	17
6	Utilisation des capteurs	18
7	Base de données	19
7.1	Système d'authentification	19
7.2	Stockage global	20
8	Ressources	20
9	Répartition des tâches	22
10	Conclusion	22

1 Introduction

Notre équipe a développé une application qui permet de gérer efficacement les documents d'achat, tout en offrant des fonctionnalités d'analyse de ces derniers.

L'application, appelée "Digiit", a pour objectif de faciliter la gestion des documents d'achat pour les utilisateurs tout en offrant une vue d'ensemble de leurs habitudes d'achat.

Une information importante à noter pour cette application est que nous avons distingué deux types de documents : les tickets et les portefeuilles.

Les tickets correspondent aux tickets de caisse tandis que les portefeuilles rassemblent les bons d'achat, les cartes de fidélité ou tout autre document similaire que l'utilisateur souhaiterait catégoriser différemment.

Cette distinction a été faite pour répondre aux besoins principaux de l'application qui sont centrés sur les tickets, tout en offrant la possibilité pratique de centraliser d'autres types de documents dans une catégorie distincte appelée portefeuille.

1.1 Secteur applicatif

Digiit appartient à la catégorie des applications d'outils pratiques, car elle offre des fonctionnalités utiles pour la gestion des documents.

Elle se positionne également dans la catégorie des applications financières, offrant des fonctionnalités avancées d'analyse des données d'achat pour permettre aux utilisateurs de mieux gérer leurs finances personnelles.

2 Proposition de Projet

Suite aux conseils de Fructueux, nous avons décidé d'ajouter une fonctionnalité OCR à notre application pour permettre aux utilisateurs de numériser rapidement leurs documents de ventes et d'achats.

Bien que le résultat ne soit pas parfait, nous détaillerons son fonctionnement dans notre prochain rendu

2.1 Besoins et étude de marché

Il existe déjà des applications similaires à notre concept, telles que (liens) :

- Easy Expense
- saldoapps
- Okticket
- Wave
- QuickBooks
- Expensify

3 Activités/Interrelations

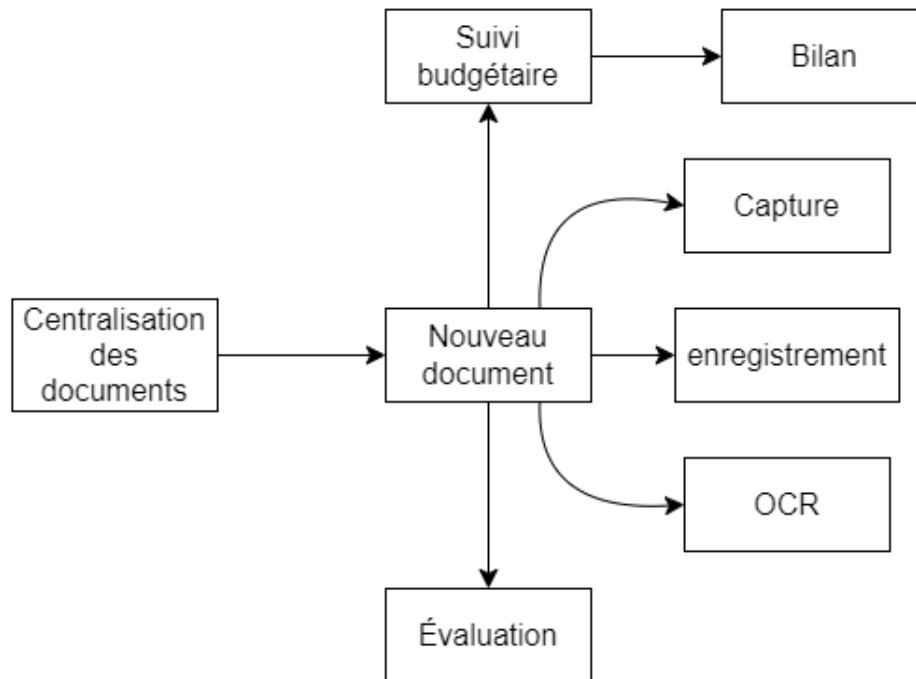
Notre application, se compose de plusieurs activités principales et secondaires qui interagissent les unes avec les autres pour fournir une solution complète de gestion des documents d'achat et d'analyse des données.

- Capture et/ou enregistrement d'un nouveau document : Cette activité permet à l'utilisateur de prendre en photo ou d'enregistrer un nouveau document, tel qu'un reçu ou une facture, afin de le stocker dans la base de données de l'application. Cette activité est directement liée à la centralisation des documents, car elle permet de stocker de nouveaux documents dans la base de données centrale.
- Suivi budgétaire : Cette activité permet à l'utilisateur de visualiser ses données de dépenses sous forme de graphiques et de tableaux, afin de mieux comprendre ses habitudes d'achat et de prendre des décisions plus éclairées en matière de gestion de ses finances. Cette activité est reliée à la centralisation des documents, car elle implique d'avoir une base de données centrale de documents pour pouvoir suivre les dépenses de l'utilisateur.
- Bilan selon des périodes : Cette activité permet à l'utilisateur de définir des périodes de temps pour lesquelles il souhaite voir un récapitulatif de ses dépenses. Cette activité est liée à la partie du suivi budgétaire et à la centralisation des documents, car elle nécessite de disposer d'une base de données centrale pour pouvoir extraire les informations nécessaires.
- Évaluation et annotation des documents : Cette activité permet à l'utilisateur de noter les différents produits ou marchands afin d'en évaluer les qualités et les défauts. Cette activité est également liée à la centralisation des documents, car elle implique

de stocker ces informations dans la base de données centrale pour pouvoir les récupérer ultérieurement.

- OCR : Cette activité permet d'automatiser certaines tâches en utilisant la reconnaissance optique de caractères pour extraire les informations importantes des documents. Cette activité est reliée à la partie Capture et/ou enregistrement d'un nouveau document, car elle permet d'automatiser la saisie d'informations dans la base de données centrale.
- Centralisation des documents : Cette activité permet de rassembler tous les documents enregistrés par l'utilisateur dans un seul et même endroit, facilitant ainsi la gestion de ses achats et la réduction du gaspillage de papier. Cela offre également une solution pratique pour l'utilisateur de stocker et d'organiser ses documents d'achat tout en contribuant à la lutte contre le réchauffement climatique.

Schéma présentant les activités principales et les liens entre elles:



Lorsqu'un utilisateur crée un compte dans une application, des slides interactifs sont fournis pour un guide de démarrage sur les activités principales. En fournissant un guide de démarrage détaillé et interactif, l'application peut aider les nouveaux utilisateurs à comprendre rapidement les fonctionnalités clés de l'application et à les utiliser plus efficacement, ce qui peut améliorer l'expérience utilisateur et l'engagement à long terme.

3.1 Slides

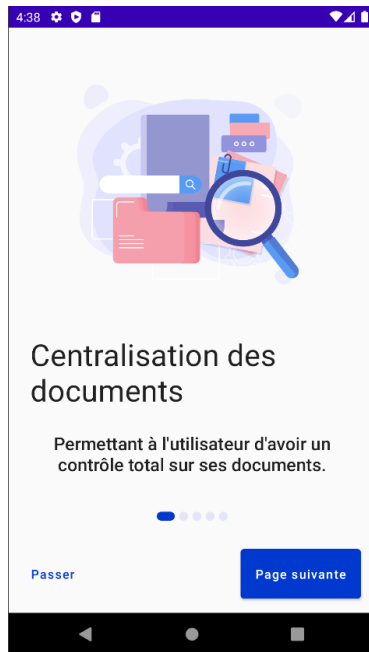


Figure 1: slide 1

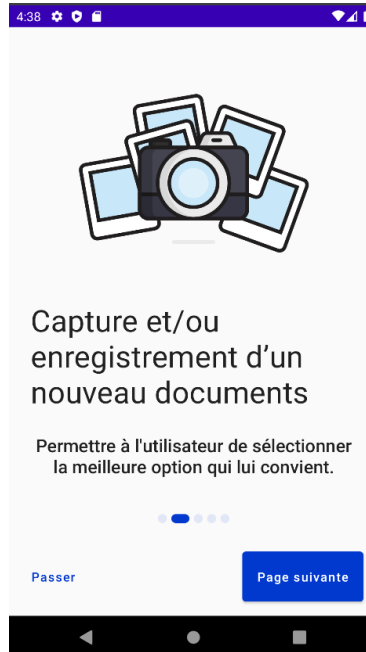


Figure 2: slide 2



Figure 3: slide 3



Figure 4: slide 4

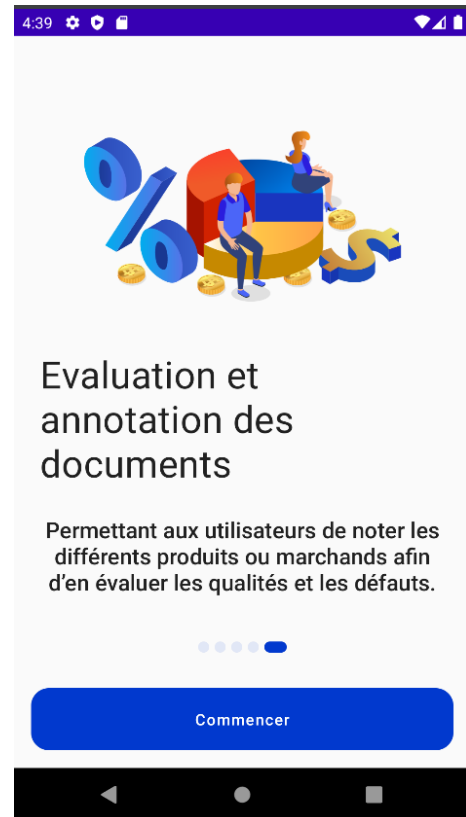
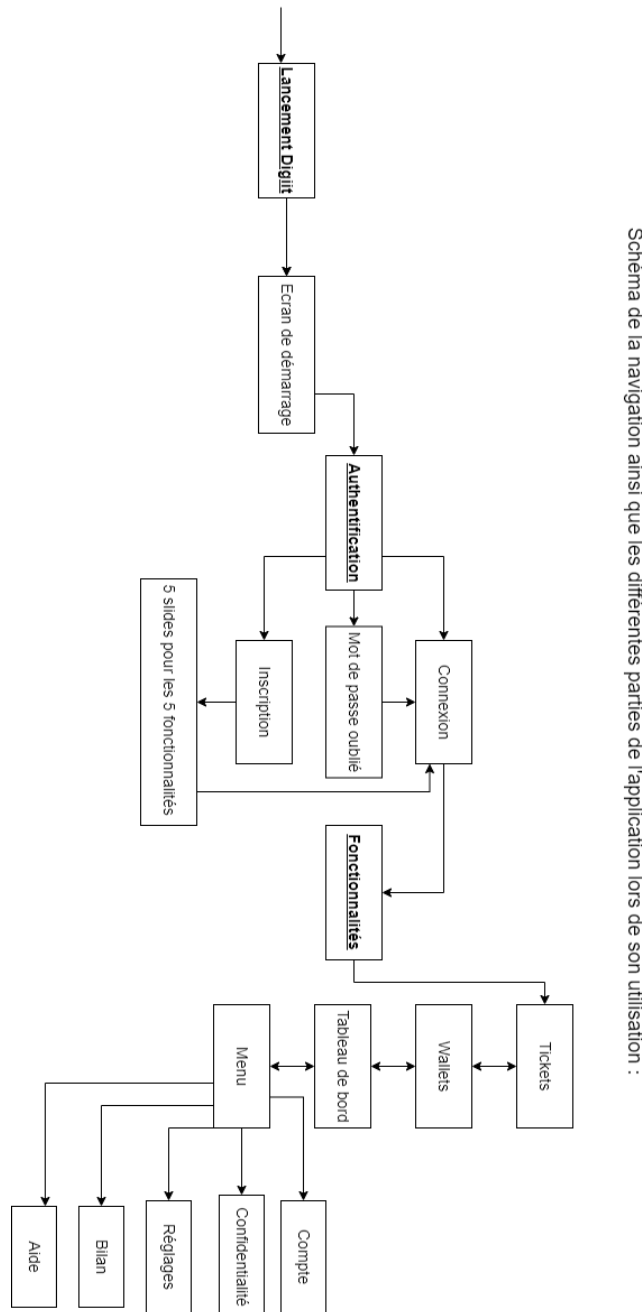


Figure 5: slide 5

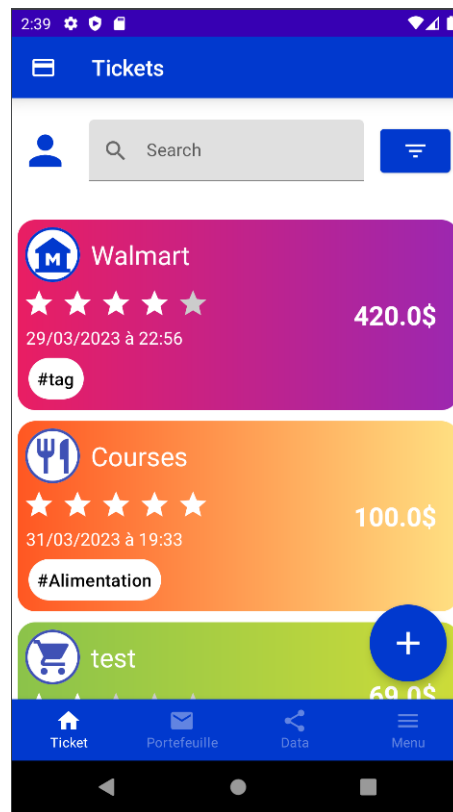
4 Interfaces

Dans cette parties nous détaillerons les pages et interface utilisateur de l'application qui nous paraissent importantes pour comprendre les activités de Digiit.

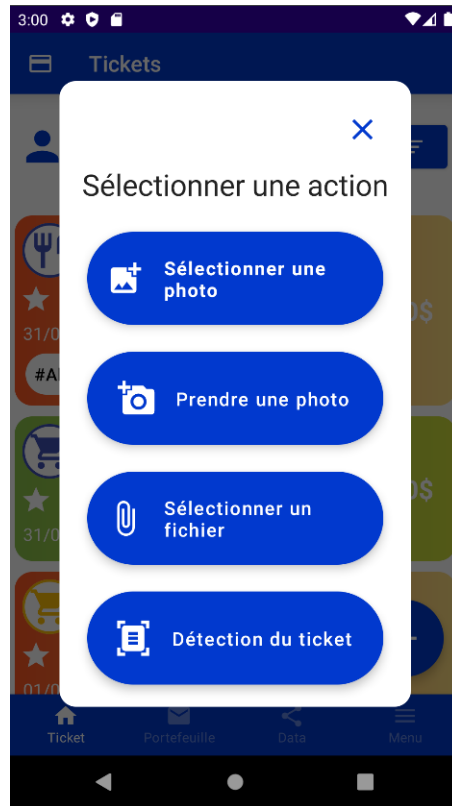
4.1 Les différentes pages de l'application



Ainsi on trouve :



-La page principale de notre application est l'endroit où les utilisateurs peuvent visualiser tous les tickets qu'ils ont déposés sur l'application. Comme vous pouvez le voir sur la capture d'écran, cette page affiche une liste de tous les tickets qui ont été créés par l'utilisateur, ainsi que leur statut actuel.



En plus de visualiser les tickets existants, les utilisateurs peuvent également créer de nouveaux tickets à partir de cette page. Nous offrons plusieurs méthodes de création de ticket pour offrir une expérience utilisateur optimale. Les utilisateurs peuvent prendre une photo de l'objet, sélectionner une image de leur galerie, mettre des documents parmi des fichiers ou encore utiliser la technologie OCR pour saisir automatiquement les informations. De plus l'utilisateur peut choisir de filtrer les différents documents comme par exemple en ordre croissant par rapport à la date ou par rapport au prix des tickets.



-On retrouve également la page Portefeuille qui permet aux utilisateurs de stocker les bons d'achats ou bons de commandes dans une page similaire à celle des tickets.

Pour ajouter des documents dans cette page on a les mêmes options que pour les tickets sauf l'OCR qui a été fait seulement pour des tickets.

Pour cette partie il nous reste simplement à ajouter et supprimer les documents.

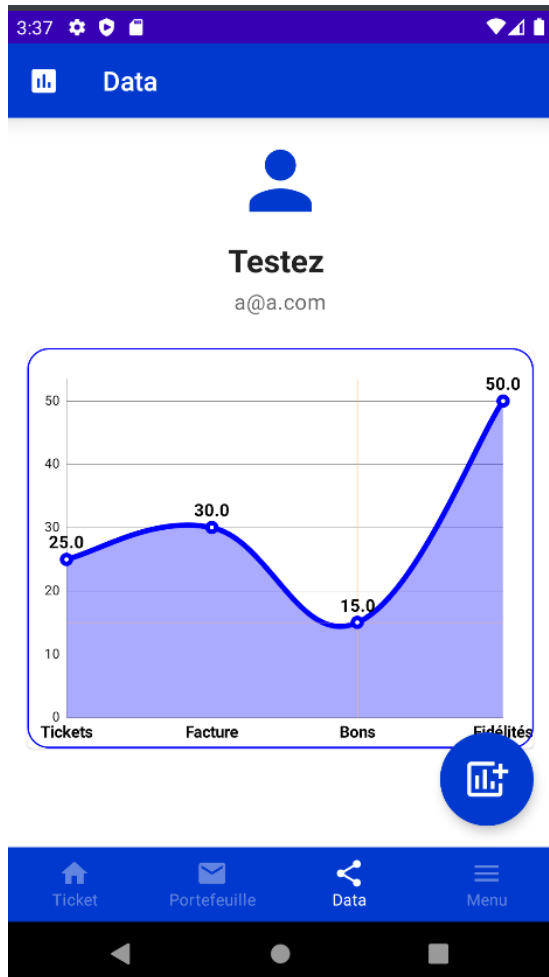


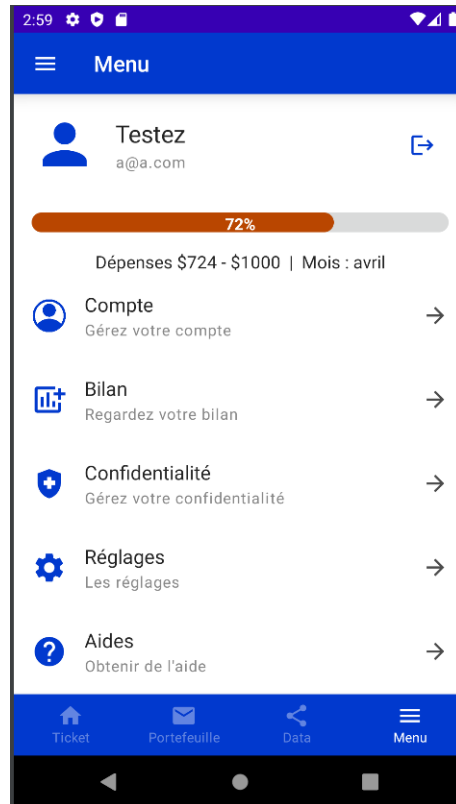
Figure 6: Data Page



Figure 7: Ajout graphe

-On retrouve également la page des données qui permet aux utilisateurs de pouvoir visualiser les différents graphiques comme par exemple des diagrammes circulaires ou encore des graphiques avec des lignes. Pour ajouter des graphiques dans cette page l'utilisateur doit choisir entre deux dates et des graphiques sont générés à partir des données de l'utilisateur.

Nous avons mis en place 8 différents graphiques sélectionnable.



Enfin, l'utilisateur peut accéder à la page de menu qui lui permet d'accéder à diverses informations, telles que ses données personnelles, son bilan financier et d'autres informations relatives à l'application, telles que les paramètres de confidentialité, les réglages et les aides.

De plus, l'application dispose d'une barre de progression qui affiche le pourcentage d'utilisation actuel par rapport à un montant défini par l'utilisateur. Cette fonctionnalité permet à l'utilisateur de mieux contrôler ses dépenses en fixant des objectifs chaque mois, comme le montre l'exemple de la barre de progression à 72% pour le mois de mars.

4.2 Choix de l'UI

Nous avons choisit dans l'application l'omniprésence du bleu et du blanc, couleur que nous retrouvons également dans le logo de l'application.

Dans l'ensemble de l'environnement nous avons gardé une interface moderne et simple pour faciliter l'expérience utilisateur. Les tickets possèdent une couleur et un icône définissable lors de leur création, de manière à les retrouver rapidement d'un simple regard.

L'interface utilisateur a été entièrement conçu avec jetpack compose.

5 Services

Notre application utilise un service qui nous permet de récupérer des données à partir d'une API que nous avons développée en interne. Cette API nous permet de communiquer avec un serveur distant pour récupérer des informations pertinentes et les transmettre à l'application.

Le service est configuré pour fonctionner en arrière-plan, permettant à l'application de continuer à fonctionner normalement tout en récupérant des données à partir de l'API. Le service utilise des appels d'API pour récupérer les données nécessaires et les stocker localement pour un accès rapide.

5.1 API

Ainsi nous avons réalisé une API qui permet de scanner et de prendre des informations à partir d'une image.

5.1.1 API

Nous avons créé une API en utilisant du code Python qui utilise plusieurs bibliothèques. Pour le traitement de l'image, nous avons utilisé la bibliothèque OpenCV.

Pour la reconnaissance de caractères, nous avons utilisé la bibliothèque Pytesseract. Après le traitement de l'image, nous avons utilisé la bibliothèque re pour la partie regex, afin de détecter des informations importantes dans le texte de l'image.

Finalement, nous avons utilisé la bibliothèque JSON pour générer un fichier JSON contenant les informations obtenues à partir de la partie regex.

Vous pouvez trouver le lien du dépôt github de cet OCR ici : [Lien](#)

5.1.2 Déploiement

Pour rendre fonctionnel notre OCR, nous avons utiliser Flask, un framework python permettant de gérer les routes (GET POST) des différentes requêtes et ainsi renvoyer des résultat au format JSON.

Afin de déployer l'application de façon permanente, nous avons utilisé Render, un hébergeur web qui permet de déployer gratuitement d'ici sites ou API. Nous avons également utilisé la technologie Docker pour pouvoir exporter le logiciel tesseract sur un serveur web. Docker est une plate-forme logicielle qui permet de créer, tester et déployer rapidement des applications. Il regroupe les logiciels dans des unités standardisées appelées conteneurs qui contiennent tout ce dont le logiciel a besoin pour s'exécuter, y compris les bibliothèques, les outils système, le code et l'environnement d'exécution.

Source :

Flask : <https://flask.palletsprojects.com/en/2.2.x/>

Render : <https://render.com/>

Docker : <https://render.com/>

6 Utilisation des capteurs

Nous avons utilisé la caméra ainsi que le flash du téléphone dans une activité, pour cela nous avons utilisé la bibliothèque CameraX d'Android, qui est une API de caméra Android de niveau supérieur qui simplifie la configuration et l'utilisation de la caméra du téléphone.

CameraX offre des fonctionnalités telles que la prise de vue en haute qualité, la prévisualisation, la capture de photos.

Avec CameraX, il est plus facile d'ajouter des fonctionnalités de caméra à notre application et de les personnaliser selon nos besoins.

Nous avons dû refaire l'interface de la caméra avec un bouton de caméra. Cela a été fait pour améliorer l'expérience utilisateur, en permettant aux utilisateurs de prendre facilement des photos et de créer des documents avec un aspect visuel.

Si nous terminons en avance, nous aimerions explorer l'utilisation d'autres capteurs du téléphone pour améliorer l'expérience utilisateur. Par exemple, nous pourrions utiliser l'accéléromètre du téléphone pour détecter quand le téléphone est en mouvement.

En utilisant cette information, nous pourrions filtrer les différents tickets ou autres documents qui apparaissent à l'utilisateur sur l'écran de l'application. Cela pourrait rendre l'application plus pratique et plus facile à utiliser, en réduisant le temps nécessaire pour trouver les informations pertinentes.

Pour cela il faudrait utiliser le "SensorManager" qui est la classe fournie par la plateforme Android.

7 Base de données

L'application, utilise *Firebase*, un système de base de données NoSQL de *Google*. Notamment le Firestore pour le stockage des données et le système d'authentification sécurisé.

C'est est un outil de développement d'applications mobiles qui offre de nombreux avantages, tels que la scalabilité, la gestion des utilisateurs simplifiée et la sécurité renforcée grâce à son système d'authentification.

Firestore, en particulier, permet une synchronisation des données en temps réel, une évolutivité élevée pour des applications de grande envergure, ainsi qu'une gestion simplifiée des données grâce à des requêtes rapides et des index personnalisables.

En utilisant Firestore pour la base de données de notre application, nous sommes en mesure de fournir une expérience utilisateur de qualité supérieure avec une synchronisation rapide et une gestion efficace des données de ventes/achats.

7.1 Système d'authentification

Le système d'authentification qu'offre Firebase permet de créer et gérer des comptes d'utilisateurs de marinières très sécurisé. Le mot de passe des chaque comptes est hashé de manière a ce que nous ne puisse jamais le voir. Nous avons ainsi mis en place un système permettant depuis l'application de : créer un compte avec son adresse mail, se connecter, redéfinir un nouveau mot de passe en cas de perte de l'ancien.

7.2 Stockage global

En utilisant le module Storge de Firebase nous avons plus sauvegarder dans la base de donnés, les ticket et wallet de chaque utilisateur ainsi que des informations complémentaires aux comptes des utilisateurs.

Ainsi lorsque que l'utilisateur arrive sur l'écran principal de l'application après s'être authentifié, l'application récupérer dans la base, toutes les donnés propre à l'utilisateur actuel.

8 Ressources

Dans cette partie, nous allons énumérer les ressources qui nous ont permis d'ajouter diverses fonctionnalités à l'application.

En ce qui concerne l'ajout dans le fichier Gradle (qui permet de configurer et de gérer la construction de l'application, les dépendances ou encore les paramètres de build) de l'application:

- Les graphiques : nous avons utilisé la bibliothèque MPChart qui est disponible en openSource : [Lien](#)
- Les dialogues : Pour les dialogues avec l'utilisateur comme par exemple les sélecteurs de dates, les sélecteurs de couleurs ou encore des sélecteurs avec des liste avec une bibliothèque également openSource : [Lien](#)
- La caméra : nous avons utilisé la bibliothèque cameraX d'Android : [Lien](#)
- Les étoiles pour noter un document : nous avons utilisé la bibliothèque openSource : [Lien](#)
- Firebase : Pour le backend : [Lien](#)

- Lottie : Pour les animations dans les slides : [Lien](#)
- OkHttp : Pour faire des requête pour notre API pour l'OCR : [Lien](#)
- JSON : Pour pouvoir utiliser le fichier en format JSON envoyé par notre API : [Lien](#)

En ce qui concerne le fichier manifest nous avons rajouter les éléments suivants :

- Photo : Nous avons ajouté les permissions :

```
android.hardware.camera.any  
android.permission.CAMERA
```

- Sélectionner des photos dans la galerie :

```
android.permission.READ_EXTERNAL_STORAGE  
android.permission.WRITE_EXTERNAL_STORAGE
```

- Utiliser internet pour l'API par exemple :

```
android.permission.INTERNET
```

- Un itent pour utiliser la caméra dans une nouvelle activité :

```
<intent-filter>  
<action android:name="android.intent.action.MAIN" />  
<category android:name="android.intent.category.LAUNCHER"  
</intent-filter>
```

9 Répartition des tâches

Prénoms :	Mathias	Hugo	Cédric	Adrien	Célian
Frontend : Authentification	0%	0%	100%	0%	0%
Frontend : Tickets/Wallets	0%	0%	75%	25%	0%
Frontend : Menu	0%	0%	100%	0%	0%
Frontend : Graphiques	50%	50%	0%	0%	0%
Backend : Authentification	0%	0%	0%	75%	25%
Backend : Stockage	0%	0%	0%	25%	75%
Backend : Graphiques	35%	35%	0%	0%	30%
OCR	0%	0%	100%	0%	0%
Déploiement OCR	0%	0%	0%	100%	0%
Etude de l'application	33%	33%	0%	0%	33%
Mise en place de Firebase	0%	0%	0%	100%	0%

10 Conclusion

En conclusion, le projet d'application Android réalisé dans le cadre du cours d'informatique mobile a été une expérience très enrichissante pour toute l'équipe. Ce projet nous a permis de comprendre les différentes étapes de conception et de développement d'une application mobile, depuis la planification jusqu'à la publication sur le Google Play Store. Nous avons également découvert de nouvelles technologies telles que Kotlin, Android Studio et jetpack compose, qui ont été essentielles pour la création de l'application.

L'utilisation de Kotlin comme langage de programmation a été une réelle découverte pour nous. Sa simplicité et sa concision nous ont permis de coder plus efficacement et plus rapidement, ce qui a été très apprécié tout au long du développement de l'application. De même,

l'utilisation d'Android Studio comme environnement de développement a été très utile pour la gestion de notre projet, la création de l'interface utilisateur, la compilation du code, le débogage, etc.

Enfin, l'utilisation de jetpack compose nous a permis de créer une interface utilisateur moderne et personnalisable pour notre application, avec des animations fluides et des transitions élégantes. Nous avons également pu bénéficier de nombreux composants pré-construits pour simplifier le développement et gagner du temps.

En somme, ce projet nous a permis de découvrir de nouvelles technologies et de les appliquer dans un contexte pratique, tout en acquérant des compétences en matière de conception et de développement d'applications mobiles. Nous sommes très fiers du résultat final et nous sommes impatients de continuer à apprendre et à explorer de nouvelles technologies dans le futur.