

RAPPORT DE STAGE

3 juin 2025 au 4 juillet 2025

```
class BigFile:
```

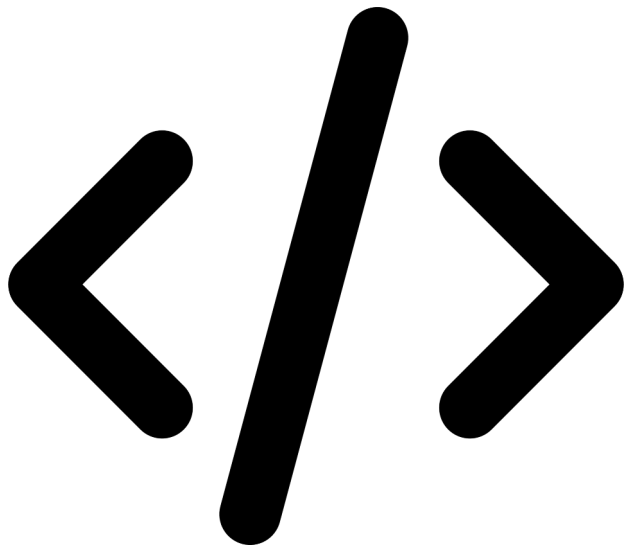
```
    def __init__(self, datadir, ndims):
        idfile = os.path.join(datadir, "id.txt")
        self.names = [x.strip() for x in str.split(open(idfile).read()) if x.strip()]
        self.name2index = dict(zip(self.names, range(len(self.names))))
        self.ndims = ndims
        self.featurefile = os.path.join(datadir, "feature.bin")
        print "[BigFile] %d features, %d dimensions" % (len(self.names), self.ndims)
        print "        binary: %s" % self.featurefile
        print "        txt: %s" % idfile
```

```
    def read(self, requested, isname=True):
        if isname:
            index_name_array = [(self.name2index[x], x) for x in requested if x in self.names]
        else:
            assert(min(requested) >= 0)
            assert(max(requested) < len(self.names))
            index_name_array = [(x, self.names[x]) for x in requested]
            index_name_array.sort()
            vecs = seq_read(self.featurefile, self.ndims, [x[0] for x in index_name_array])
            return [x[1] for x in index_name_array], vecs

    def shape(self):
        return [len(self.names), self.ndims]
```



SOMMAIRE



3. BNP PARIBAS

4. Service informatique

5. tache effectuer

6. Swift

7. Projet

11. conclusion



BNP PARIBAS

1. Identification de l'entreprise

- Nom : BNP Paribas
- Secteur d'activité : Banque, finance et assurance
- Forme juridique : Société Anonyme (S.A.)
- Siège social : 16 boulevard des Italiens, 75009 Paris, France
- Effectif : Environ 180 000 collaborateurs dans le monde
- Présence : Plus de 70 pays

2. Historique et évolution

BNP Paribas est née en 2000 de la fusion entre la Banque Nationale de Paris (BNP) et Paribas. Depuis, elle est devenue l'un des groupes bancaires les plus importants en Europe et un acteur majeur à l'échelle mondiale. Elle s'appuie sur une histoire riche de plus de 150 ans dans le domaine bancaire.

3. Activités principales

BNP Paribas propose une offre complète de services financiers, répartie en trois grands pôles :

Banque de détail : services aux particuliers, artisans, commerçants, PME.

Banque de financement et d'investissement : pour les grandes entreprises et les marchés financiers.

Services spécialisés et assurances : crédit à la consommation, location longue durée, assurances, immobilier.

4. Organisation et structure

Le groupe est organisé selon une structure internationale, avec une forte implantation en Europe (France, Belgique, Italie, Luxembourg) mais aussi une présence significative en Amérique, Asie et Afrique. Il fonctionne avec une gouvernance claire et de nombreuses filiales spécialisées.

6. Quelques chiffres (2024)

- Produit Net Bancaire : ~ 45 milliards d'euros
- Résultat net : ~ 10 milliards d'euros
- Total du bilan : ~ 2 600 milliards d'euros

Service informatique

Lors de mon stage, j'ai été accueilli au sein du service développement mobile de BNP Paribas. Ce service est chargé de concevoir et de maintenir les applications mobiles utilisées par les clients de la banque.

L'équipe utilise principalement deux langages de programmation:

- Swift: un langage développé par Apple, utilisé pour créer des applications destinées aux appareils fonctionnant sous iOS (iPhone, iPad).
- Kotlin: un langage moderne recommandé par Google, utilisé pour le développement d'applications sous Android.

Le fait d'utiliser deux langages différents permet de garantir une compatibilité multiplate-forme. Cela signifie que les clients, qu'ils utilisent un smartphone Apple ou Android, peuvent accéder aux mêmes services bancaires via une application adaptée à leur système d'exploitation. Cette organisation permet d'assurer une expérience utilisateur fluide et homogène, quel que soit l'appareil utilisé.

Organisation du service

Le service est divisé en plusieurs équipes de développement, chacune ayant une mission bien spécifique. Ces équipes travaillent souvent en parallèle sur différents aspects de l'application ou sur des projets distincts. Voici quelques exemples d'équipes au sein du service :

- Stellar
- Wero
- Hello bank !
- Tokyo (l'équipe à laquelle j'ai été rattaché)

Mission effectuée

Tout au long de mon stage, j'ai eu l'opportunité de découvrir le monde du développement d'applications mobiles dans un environnement professionnel. Mon objectif principal était de m'initier au langage Swift, utilisé pour développer des applications sur iOS, tout en observant et en participant aux différentes étapes du cycle de développement.

Mon stage m'a permis de mieux comprendre le fonctionnement d'une équipe de développement ainsi que les méthodologies utilisées pour concevoir, tester et améliorer une application mobile. À la fin de mon stage, j'ai pu présenter un projet développé en Swift, mettant en application les connaissances acquises.

Voici un aperçu des missions que j'ai pu suivre ou sur lesquelles j'ai pu intervenir :

- **Tests d'applications** : j'ai participé à des phases de tests pour assurer que les fonctionnalités étaient conformes aux attentes et qu'aucun bug ne perturbait l'utilisation.
- **Résolution de problèmes** : j'ai observé comment les développeurs identifient et corrigent les bugs.
- **Ajout de fonctionnalités** : j'ai pu assister à l'ajout de nouvelles fonctionnalités dans l'application, ce qui m'a permis de voir comment une idée passe de la conception au code.
- **Analyse et amélioration du code** : j'ai appris à lire et comprendre du code existant afin de proposer des améliorations en termes de lisibilité, d'efficacité ou de bonnes pratiques.

Ces missions m'ont permis d'avoir une vision concrète du métier de développeur mobile, de comprendre le travail en équipe, l'importance des tests et du code bien structuré, ainsi que les outils utilisés au quotidien.

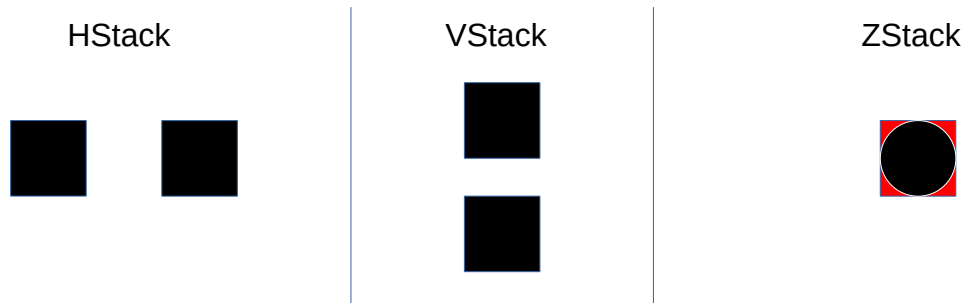
Swift

Durant mon stage, j'ai principalement utilisé le langage Swift, un langage de programmation moderne développé par Apple. Swift est aujourd'hui le langage principal pour créer des applications iOS, macOS, watchOS et tvOS. Il a été conçu pour être à la fois puissant et simple à apprendre, ce qui le rend accessible même pour des développeurs débutants, tout en restant très performant pour des projets complexes.

Swift repose sur une syntaxe claire et intuitive. Cela m'a permis de rapidement comprendre la structure du code, même sans avoir une longue expérience en programmation mobile. L'un des aspects que j'ai particulièrement apprécié est la lisibilité du code Swift : les fonctions, les conditions, les boucles ou encore la gestion des erreurs sont faciles à repérer et à comprendre. De plus, Swift propose de nombreux outils intégrés pour éviter les erreurs courantes, comme les variables optionnelles, qui obligent le développeur à anticiper l'absence de valeur.

Un autre point fort de Swift est son intégration avec SwiftUI, le framework utilisé pour construire des interfaces graphiques. Grâce à SwiftUI, on peut créer une interface en quelques lignes, tout en la rendant responsive et dynamique.

Au cours de mon stage, j'ai pu utiliser SwiftUI pour créer différentes interfaces dans mon application bancaire. J'ai appris à utiliser les Stacks (VStack, HStack, ZStack) pour organiser les éléments visuellement, à personnaliser les composants avec des couleurs, tailles, bordures, et à gérer les actions avec des boutons ou des champs de texte.



J'ai également appris à structurer mon code avec des structs, des fonctions, et à créer des vues réutilisables. Le fait de pouvoir voir en temps réel les modifications dans l'aperçu Xcode m'a beaucoup aidé à mieux comprendre le lien entre le code et le résultat visuel.

Enfin, Swift m'a permis d'améliorer mes compétences en programmation orientée objet, en manipulation des données, en logique conditionnelle et en gestion d'événements. Ce langage, bien que différent de ceux que j'avais pu apprendre auparavant comme Python ou Java, m'a apporté une nouvelle vision du développement, orientée vers l'expérience utilisateur et l'efficacité.

Projet

Dans le cadre de mon stage, une des missions principales qui m'a été confiée fut la réalisation d'un projet concret permettant de mettre en application les compétences que j'ai développées durant la période. Ce projet consistait à créer une application bancaire destinée à iOS, en utilisant le langage Swift.

Démarrage et apprentissage

Avant de me lancer dans le développement de l'application, j'ai dû me former au langage Swift. Cette phase d'apprentissage a été accompagnée par :

Mon tuteur ainsi que certains membres de l'équipe de développement, qui ont pris le temps de m'expliquer les bases et les mécanismes du langage.

Une formation via l'application "Playgrounds" sur Mac, outil très utilisé pour s'entraîner au code Swift de manière interactive. Cette application m'a permis d'apprendre progressivement les notions fondamentales du langage tout en pratiquant.

Grâce à cette formation sur Playgrounds, j'ai pu réviser certaines notions que j'avais déjà acquises durant ma scolarité, tout en les appliquant dans un cadre concret. Elle m'a également permis de développer mon autonomie : face aux difficultés rencontrées, j'ai pris l'habitude de chercher des solutions par moi-même, en explorant la documentation, en testant différentes approches et en échangeant avec mes collègues. Cela m'a permis de mieux comprendre le fonctionnement du langage Swift et de progresser rapidement.

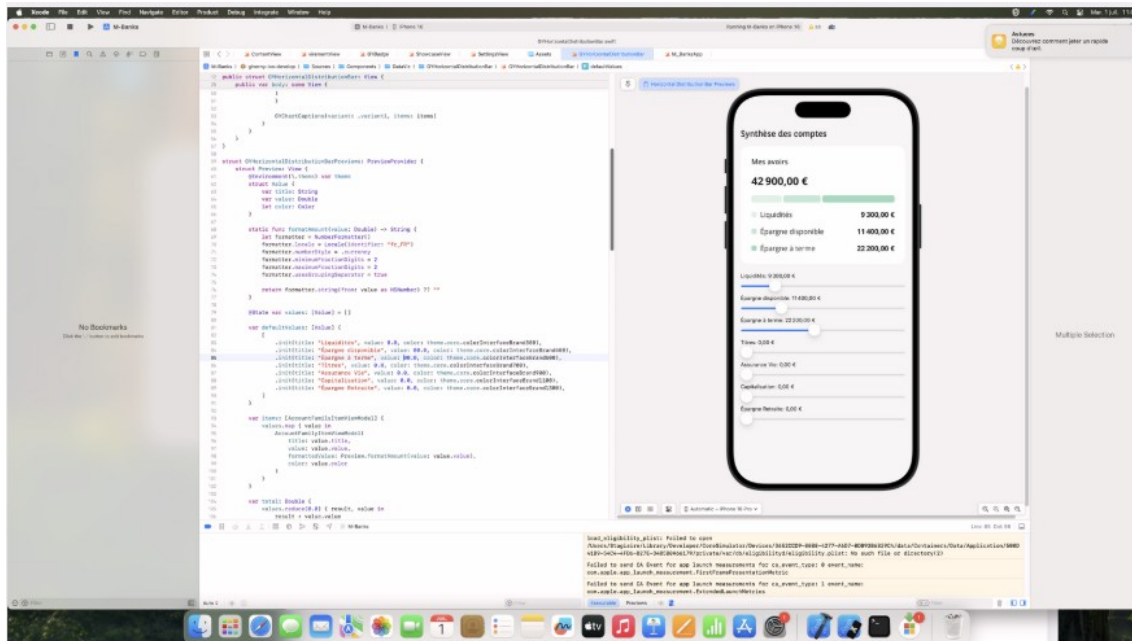


À la suite de cette formation, j'ai pu commencer à coder. Cependant, avant de débiter le développement, mon tuteur a intégré la bibliothèque de la BNP afin que je puisse utiliser les éléments graphiques et les composants propres à l'identité de l'entreprise.

Avant de commencer le développement, j'ai d'abord dû installer Xcode, l'environnement de développement d'Apple, ainsi qu'un simulateur d'iPhone. Cela m'a permis de coder et de tester mon application comme si j'étais sur un véritable appareil. Une fois l'environnement prêt, mon tuteur a intégré la bibliothèque interne de la BNP, me donnant accès aux composants visuels et aux éléments de design propres à l'identité de l'entreprise.

Pour m'aider dans l'utilisation de cette bibliothèque, une documentation m'a été fournie, expliquant le fonctionnement des différents éléments ainsi que la manière de les intégrer dans mon code.

La documentation utilisée



En cas de blocage, j'ai également pu compter sur l'aide de mes collègues de travail, qui ont pris le temps de répondre à mes questions et de m'expliquer certaines notions lorsque cela était nécessaire. Cela m'a permis de progresser plus rapidement et de mieux comprendre les outils mis à ma disposition.

J'ai également eu l'occasion de réaliser des maquettes afin de planifier l'interface de mon application. Grâce à l'outil Canva, j'ai pu trouver l'inspiration et concevoir des idées de maquettes représentant les différentes pages et fonctionnalités de l'application. Cette étape m'a permis de mieux visualiser la structure de mon projet avant de passer au développement.

En codant, j'ai pu mieux comprendre certaines lignes de code Swift et observer les effets qu'elles produisaient directement sur l'application via le simulateur. Cela m'a permis de faire des tests en temps réel et d'ajuster mes éléments en fonction du rendu visuel.

Voici quelques lignes de code que j'ai trouvées intéressantes et que j'ai appris à maîtriser durant mon stage :

```
219
220     struct SystemControlsListPreview: PreviewProvider {
221         static var previews: some View {
222             ContentView()
223                 .gyBackground(.default)
224         }
225     }
226
```

Ce code offre un aperçu en temps réel de la vue SwiftUI directement dans Xcode, sans nécessiter l'exécution de l'application sur un simulateur ou un appareil physique. Cela s'avère très utile pour effectuer rapidement de petites modifications.

Pendant toute la durée de ce projet, j'ai été confronté à plusieurs bugs que je ne comprenais pas au début. Grâce à l'aide de mes collègues, à la documentation, et à mes recherches personnelles, j'ai réussi à les corriger et à mieux comprendre leur origine.

Correction

`formattedValue: ContentView.formatAmount(value: value.value)`

Erreur

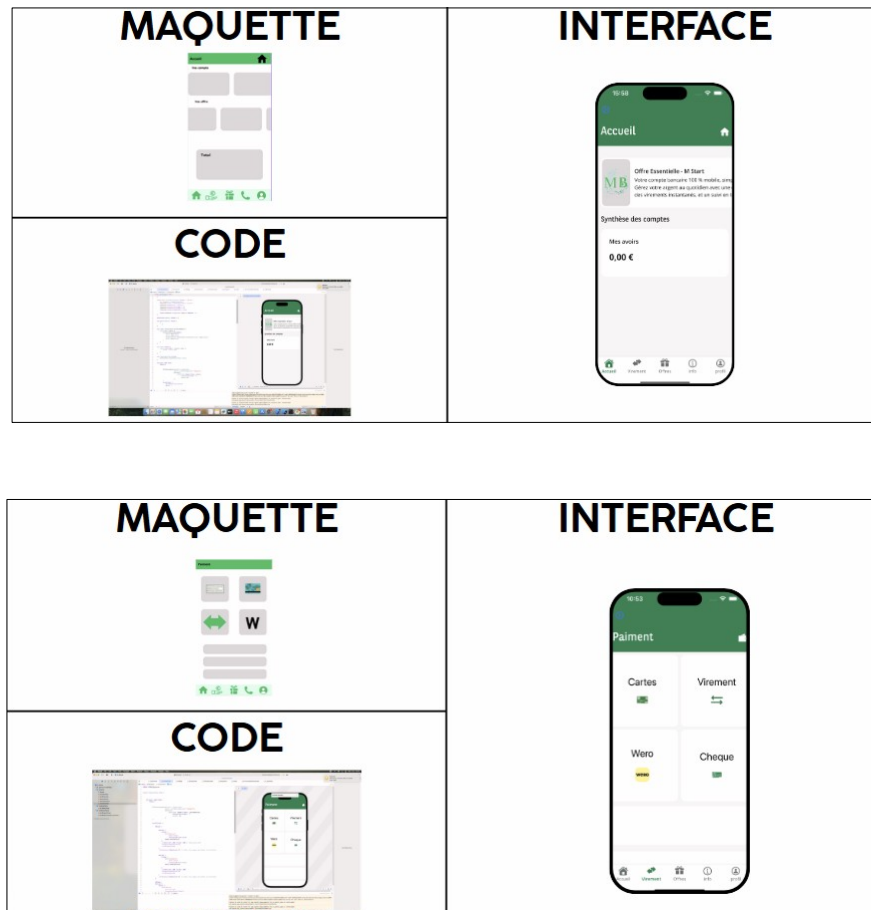
`ContentView.formatAmount(value: value.value)`

Dans cet exemple, la fonction `formatAmount` est appelée avec la variable `value.value` en paramètre. Elle retourne une chaîne de caractères déjà formatée, généralement utilisée pour afficher une valeur numérique de manière lisible dans l'interface utilisateur (par exemple avec un symbole monétaire ou un nombre arrondi).



Malheureusement, mon projet n'a pas pu être finalisé dans les délais impartis. Cependant, j'ai pu récupérer les fichiers pour continuer à travailler dessus chez moi, dans le but d'approfondir mes connaissances en Swift.

Ci-dessous, je présente quelques schémas des différentes pages de mon application, qui illustrent la structure et l'organisation de l'interface utilisateur.



Matériel utilisé

- Pour la réalisation de mon projet, j'ai utilisé différents outils et ressources, indispensables au développement et à la conception de mon application :
- Xcode : l'environnement de développement intégré (IDE) d'Apple, utilisé pour écrire, compiler et tester le code Swift.
- Un Mac : nécessaire pour faire fonctionner Xcode et développer des applications iOS.
- Le simulateur d'iPhone intégré à Xcode : utilisé pour tester l'application comme si elle était lancée sur un véritable appareil.
- La documentation technique fournie par la BNP : elle m'a guidé dans l'utilisation des composants spécifiques à la charte graphique de l'entreprise.
- Canva : un outil en ligne qui m'a permis de concevoir des maquettes et de visualiser le design des différentes pages de mon application.
- Internet : utilisé pour effectuer des recherches d'images, consulter des extraits de code, et trouver des solutions à certains problèmes rencontrés.

Conclusion

Ce stage m'a permis de découvrir concrètement le monde du développement en entreprise, . J'ai pu participer activement à un projet réel, ce qui m'a offert une vision plus claire du métier de développeur, de ses exigences techniques mais aussi de son organisation quotidienne.

Même si je n'ai pas pu terminer complètement le projet dans les délais impartis, cette expérience a été très enrichissante. J'ai pu récupérer les sources de mon application afin de continuer à travailler dessus chez moi, approfondir mes compétences, et continuer à apprendre par moi-même. Cela témoigne de ma motivation à progresser dans ce domaine.

Grâce à ce stage, j'ai également compris l'importance de la collaboration et de la communication au sein d'une équipe. J'ai souvent pu compter sur l'aide de mes collègues pour résoudre certains problèmes, comprendre des concepts ou simplement bénéficier de conseils pratiques. Cette dynamique m'a permis de gagner en autonomie et de développer une méthode de travail plus structurée.

J'ai aussi pris conscience de la richesse des outils mis à disposition des développeurs : documentation, simulateurs, bibliothèques internes, et ressources en ligne. Ces éléments sont essentiels pour mener à bien un projet, et j'ai appris à les utiliser au bon moment.

En résumé, ce stage a été une étape précieuse dans mon parcours. Il m'a donné envie d'aller plus loin dans le développement et m'a conforté dans l'idée que je souhaite évoluer dans ce secteur à l'avenir.