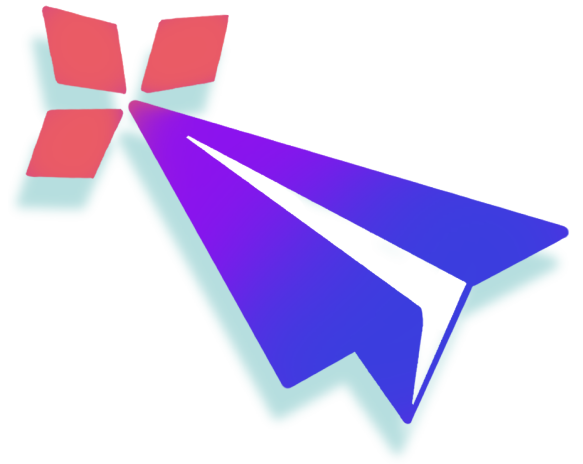




IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom



Rapport

Projet CODEVSI 101

Réseau social interne au campus

Étudiantes :

Alexia CARAVACA

alexia.caravaca@imt-atlantique.net

Leslie DAI

leslie.dai@imt-atlantique.net

Chloé DUCASSE

chloe.ducasse@imt-atlantique.net

Juliette RENAULT

juliette.renault@imt-atlantique.net

À destination de :

COPIL, ERIC COUSIN, LE CLIENT (ÉLÈVES D'IMT ATLANTIQUE
DU CAMPUS DE BREST)

Encadrant :

Eric COUSIN

Le 27 mai 2024

Version 1

Table des matières

Résumé	4
Introduction	1
1 Conception	3
1.1 Elaboration	3
1.2 Fil d'actualité et liste des clubs	4
1.3 Authentification et statut	5
1.4 Filtre	5
1.5 Messagerie	5
1.6 Lien de Téléchargement	6
1.7 Gestion des erreurs	6
2 Réalisation	7
2.1 Architecture globale	7
2.2 Fil d'actualité et liste des clubs	7
2.3 Authentification et statut	8
2.4 Filtres	9
2.5 Messagerie	10
2.6 Notification	10
2.7 Lien de téléchargement	10
3 Validation	12
3.1 Validation des fonctionnalités accessibles par un utilisateur	12
3.2 Validation des fonctionnalités accessibles par un président	14
3.3 Validation des fonctionnalités accessibles par un membre d'un BDX	15
4 Déploiement	16
4.1 Nécessités pour le déploiement	16
4.2 Contraintes de déploiement	17
4.3 Objectifs de déploiement/ lien de téléchargement	17
4.4 Plan de déploiement à la rentrée de septembre	18
Conclusion	19
Annexe	
Images	
Planning	

Résumé

Rédactrices : Alexia et Leslie

Relectrices : Chloé et Juliette

AtClub est une application que nous avons développée afin de faciliter la communication entre les clubs de l'école IMT Atlantique du campus de Brest et les étudiants. En effet, l'application regroupe les fonctionnalités suivantes : un accès à une liste de tous les clubs répertoriés sur le campus, un fil d'actualité avec tous les messages liés aux clubs que les utilisateurs postent sur l'application, la capacité d'envoyer des messages dans des groupes de discussion associés à chaque club et enfin une modération des accès et des droits sur l'application suivant son statut (président de club, responsable de BDX (Bureau des X) ou un étudiant lambda). Cette application répond au besoin de centralisation de l'information qu'ont émis les élèves de l'école et permet de mieux informer les étudiants pour qu'ils profitent pleinement de la vie associative du campus de Brest. Cette application fut réalisée en moins de 3 mois et montre des résultats concluants. Les tests du cahier des charges sont tous validés. A l'exception de la fonction de "notifications push" qui nécessite un financement.

Enfin, le déploiement de l'application se fera par lien APK (Android Application Package) sur Android seulement, malgré le choix d'un langage flexible, pour des raisons de coûts importants sur iOS.

Introduction

Rédactrices : Alexia et Leslie

Relectrices : Chloé et Juliette

Contexte

Un étudiant d'IMT Atlantique de première année, sur le campus de Brest, découvre le campus et ses fonctionnalités. Il veut vite participer à la vie associative et s'inscrire aux clubs qui correspondent à ses passe-temps ou qui l'intriguent tout simplement. Il a sûrement regardé au préalable la brochure envoyée par l'école qui répertorie la majorité des clubs et a donc déjà fait ses choix. Cependant, la réalité et la publicité ne convergent pas toujours. Certains clubs présentés sur la brochure peuvent avoir disparu, être inactifs, ne pas être accessibles aux amateurs.

En effet, le nouvel arrivant manque d'informations : "Quels sont les clubs présents sur le campus ?" et "Comment les rejoindre ?". Comme le montre notre sondage réalisé sur 104 enquêtés, ces préoccupations autour de la communication des clubs existent au sein des élèves.

Problématique

Ce manque d'informations engendre les problèmes suivants : les étudiants ne profitent pas pleinement des clubs à leur disposition (presque 4 élèves sur 5 ne connaissent pas l'intégralité des clubs), les clubs manquent de membres, le budget que l'école leur fournit n'est pas pleinement exploité. Actuellement, la communication des clubs s'éparpille sur plusieurs plateformes telles que Facebook, Messenger, Discord, WhatsApp et autres. Et rejoindre ces canaux de diffusion n'est pas souvent facile (plus d'un quart des étudiants sont perdus face à cette multiplicité des canaux). Pour rejoindre un club, l'étudiant doit :

- Savoir que ce club existe (plus d'un tiers des étudiants ont appris l'existence de leurs clubs grâce à des connaissances) ,
- Savoir qui est le/la président(e) du club pour lui demander l'adhésion,
- Avoir accès à la plateforme utilisée pour communiquer.

On remarque que plus de la moitié des étudiants peine à trouver certains clubs.

Plan

Notre application a pour objectif de rendre les clubs d'IMT Atlantique Brest plus accessibles, en leur offrant un lieu de centralisation des informations, de partage et de discussion. Elle donne accès à l'ensemble des élèves d'IMT Atlantique du campus de Brest, et aux alumni, à une liste des clubs en vigueur dans l'école et leurs responsables (présidents et bureau BDX). Les élèves peuvent communiquer à travers des groupes de discussion associés aux clubs et les présidents commu-

niquent à l'ensemble de la communauté à travers des posts sur le fil d'actualité. Nous avons nommé cette application AtClub.

Dans ce rapport nous vous exposerons comment fut pensée et conçue l'application, par quels moyens et de quelle manière elle fut réalisée, les tests de validation auxquels nous l'avons soumise pour valider ses fonctionnalités et enfin les stratégies de déploiement que nous comptons mettre en œuvre.

1 Conception

Rédactrices : Chloé et Juliette

Relectrices : Alexia et Leslie

L'application doit être composée de différentes pages permettant toutes de répondre aux fonctionnalités identifiées : pour l'utilisateur, 3 principales (FP_Utilisateur), 7 contraintes (FC_Utilisateur) et 3 facultatives (FF_Utilisateur) ; pour le président, 1 principale, 2 contraintes et 1 facultative ; et enfin, pour le membre d'un BDX, 2 principales et 1 contrainte. Un récapitulatif de ces différentes pages et de leur accessibilité se trouve en annexe, figure 5.

1.1 Elaboration

Tout d'abord, les outils choisis pour coder l'application sont Flutter (sur Android Studio) pour le frontend, et Firebase, pour le backend. En effet, Flutter permet de développer des applications pour Android et IOS à partir d'un seul code source, ce qui réduit considérablement le temps et les efforts de développement. Il donne des applications fluides et réactives et propose une vaste bibliothèque de widgets personnalisables qui facilitent la création d'interfaces utilisateur attractives et cohérentes. Quant à Firebase, cet outil permet de synchroniser les données entre les clients en temps réel, ce qui est essentiel pour réaliser des mises à jour instantanées (ce qui est nécessaire pour le fil d'actualité et la messagerie). De plus, il a un service d'authentification intégré. En outre, sur Firebase, nous pouvons stocker des fichiers (images, vidéos), ce qui est utile pour le fil d'actualité et la messagerie.

Ensuite, l'élaboration de l'architecture de la base de données et du visuel de l'application a été primordiale. L'architecture de la base de données a évolué au fur et à mesure des ajouts des différentes classes (les ajouts des fonctionnalités). La version finale est disponible en annexe, figure 2.

En ce qui concerne le visuel de l'application, une palette de couleur a été décidée, permettant la création d'une application visuellement agréable et lisible pour tous, avec un contraste suffisamment élevé (Annexe, figure 3). De plus, chaque avancée de la conception du projet a commencé par la modélisation d'une maquette s'assurant de la facilité d'utilisation de l'application (FC5_utilisateur, FC1_BDX, FC1_président). Un exemple d'une maquette peut être observé en annexe, figure 4. Par ailleurs, un logo a aussi été créé pour affirmer l'identité visuelle de l'application.

Au final, en utilisant l'architecture de la base de données et les différentes maquettes, nous sommes arrivées à l'architecture suivante :

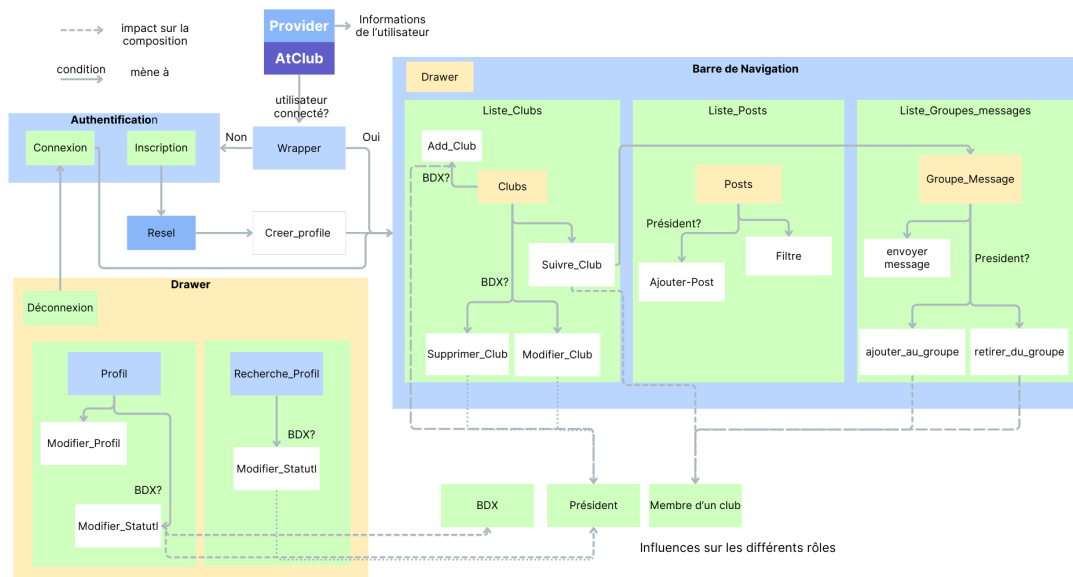


FIGURE 1 – Architecture de notre application

Les différents rectangles symbolisent les pages ou formulaires. Les rectangles imbriqués montrent les pages contenues les unes dans les autres.

1.2 Fil d'actualité et liste des clubs

La première page s'affichant, une fois l'authentification faite, est le fil d'actualité. Des posts composés d'un titre, d'un corps, d'un club, d'un auteur et de la date y sont présentés (dans l'ordre anti-chronologique) permettant à tout utilisateur de voir les informations des multiples clubs ainsi que leurs activités organisées (FP2_Utilisateur). De plus, si l'utilisateur possède le statut de président d'un club, alors il a la possibilité d'ajouter un post, relié au club qu'il préside, en remplissant un formulaire (FP1_Président).

En outre, tout utilisateur peut accéder à la liste des clubs (avec le nom, le président, le co-président si existant et la description du club). Ces derniers sont rassemblés par BDX. Cela permet à l'utilisateur de connaître tous les clubs présents sur le campus (FP1_Utilisateur). Si l'utilisateur possède le statut de membre d'un BDX, alors il a la possibilité de créer un club appartenant à son bureau (FP1_BDX). Il peut également modifier et supprimer un club associé à son bureau.

1.3 Authentification et statut

L'authentification se fait lors de la première utilisation de l'application ou si l'utilisateur souhaite changer de compte (s'il prête son téléphone à une autre personne, par exemple). Lors de l'inscription, l'identification se fait avec l'identifiant (ID) et le mot de passe du Resel. Cela permet de s'assurer de l'identité de l'utilisateur (FC2_utilisateur et FC1_utilisateur), c'est-à-dire, que c'est bien un élève ou alumni d'IMT Atlantique du campus de Brest. Suite à cela, le formulaire d'inscription s'ouvre, permettant à l'utilisateur de renseigner son email et un mot de passe. Ensuite, un dernier formulaire permettant la création d'un profil (nom, prénom, description, image de profil) s'affiche (FP3_Utilisateur) permettant aux autres utilisateurs d'identifier les membres (en particulier dans la messagerie). Ce profil peut être modifié par la suite et affiché depuis la page "Profil". Après envoi du formulaire, l'utilisateur a accès aux différentes fonctionnalités de l'application, en tant qu' "élève". Lors d'une reconnexion, seuls l'email et le mot de passe de la personne sont demandés.

Le changement de statut, permettant la gestion des différents rôles des utilisateurs (FP2_BDX), se fait par les utilisateurs membres d'un BDX via un formulaire de modification de statut. Ce dernier est accessible depuis les profils qui peuvent être trouvés grâce à la page permettant de rechercher un profil. Cette logique permet de donner accès à certaines fonctionnalités en fonction du statut de l'utilisateur (élève, président ou membre d'un BDX).

1.4 Filtre

Les posts présents sur le fil d'actualité peuvent être filtrés (FF1_Utilisateur) par clubs. En cliquant sur un bouton, l'utilisateur peut alors sélectionner les différents clubs (qui sont triés par BDX, pour une utilisation facilitée) dont il souhaite que les posts apparaissent. Une fois, cela fait, le fil d'actualité est mis à jour et seuls les posts appartenant aux clubs sélectionnés s'affichent dans l'ordre anti-chronologique. Cela permet d'avoir accès aux informations recherchées plus facilement et plus rapidement.

1.5 Messagerie

Une messagerie est aussi accessible par tous les utilisateurs (FF2_Utilisateur). Pour avoir accès au groupe de messages d'un club, l'utilisateur doit suivre le club en cliquant sur le bouton "suivre" présent dans la liste des clubs. Ces groupes permettent aux membres d'un club de discuter d'autres informations ou détails des activités postées sur le fil d'actualité. Cependant, si l'utilisateur est président d'un club, alors il fait automatiquement partie du groupe de discussion associé.

En tant que président du club, il peut aussi ajouter directement une personne ou en supprimer une (FF1_Président). Cela permet d'éviter les débordements.

1.6 Lien de Téléchargement

Un lien de téléchargement, sous forme d'APK, est disponible gratuitement (FC3_utilisateur, FC6_utilisateur), permettant aux personnes ayant le lien d'avoir accès à l'application. Cependant, seuls les élèves et alumni d'IMT Atlantique sur le campus de Brest pourront se connecter.

1.7 Gestion des erreurs

Pour chaque formulaire, les champs obligatoires s'affichent en rouge si l'utilisateur n'a pas saisi une information adéquate ou s'il n'a rien saisi. Tant que tous les champs obligatoires ne sont pas correctement renseignés, le formulaire ne peut pas être envoyé.

En somme, l'application répond au principal besoin (centraliser les clubs) et permet aux utilisateurs de se tenir informés des activités des clubs et d'interagir avec d'autres membres. La gestion des rôles permet d'assurer aux utilisateurs une expérience fluide et intuitive. Dans la suite de ce document, les aspects techniques de l'application et les technologies utilisées seront détaillés.

2 Réalisation

Rédactrices : Alexia et Leslie

Relectrices : Chloé et Juliette

2.1 Architecture globale

Chaque page de l'application correspond à une page de codes commentés. De même, les modèles éventuels utilisés dans les pages sont séparés dans un dossier à part. Cette factorisation permet d'éviter les répétitions, une meilleure relecture des différents codes, une maintenance plus facile et rapide, mais aussi une compréhension plus aisée. Elle facilite en somme le travail en équipe en séparant clairement les différents modules. Le frontend est aussi clairement séparé du backend (Annexe, figure 6) permettant une avancée plus rapide et une gestion plus facile en cas d'erreurs. Chaque partie a donc son dossier et son propre fichier.

2.2 Fil d'actualité et liste des clubs

Afin de réaliser la liste des posts pour le fil d'actualité, on construit un Widget avec un Stream Builder qui se reconstruit en fonction des modifications dans la base de données de Posts en temps réel. Il permet de gérer la fréquence des posts et le fait que les données concernant la liste des posts changent souvent. Ce Streambuilder [1] récupère la liste de ces derniers sous-forme de dictionnaire avec l'id du post en clé et ses données en valeur. Par la suite, les différents posts sont récupérés sous la forme d'un dictionnaire. En appelant récursivement le modèle des posts (le code permettant d'en afficher un seul), ils apparaissent tous sous la même forme, mais avec leurs propres informations (FP2-Utilisateur). Ce modèle permet d'éviter d'avoir un code répétitif et ainsi d'éviter les erreurs.

Pour créer un post (FP1-Président), nous utilisons un formulaire. Le formulaire est composé de modèles que nous avons codés pour le réutiliser de manière générique sur les autres formulaires. Il contient des champs obligatoires permettant l'interdiction de l'envoi des données si celles-ci ne sont pas correctement écrites (FC1-Président). De plus, le champ contenant le club auquel est relié le post est codé sous la forme d'une liste déroulante des différents clubs dont l'utilisateur est le président. Cela permet d'éviter à l'utilisateur de poster pour un club dont il n'est pas président (FP1-Utilisateur). Ensuite, de façon similaire, avec un modèle

appelé récursivement, la liste des clubs (FP1-Utilisateur) a été créée comportant un tri en fonction des BDX (que l'on peut trouver dans les informations du club). Sa mise en forme a évolué au fil du temps afin d'optimiser au mieux l'expérience

utilisateur. Au début, la liste des clubs était sur une page triée alphabétiquement et en fonction du BDX. Cependant, cela nécessitait un long défilement de l'écran afin de parcourir l'ensemble des clubs. Ainsi, nous avons décidé de faire une "tabbar" ordonnant les différents clubs par bureaux [2] (Annexe, figure 7). Pour créer un club (FP1-BDX), le même modèle de formulaire est utilisé en changeant seulement les noms des champs obligatoires .

2.3 Authentification et statut

Pour la réalisation de l'authentification, la création d'un formulaire d'inscription avec l'ajout dans la base de données de ce nouvel utilisateur est nécessaire. En effet, une exploitation de la base de données du Resel déjà existante a été envisagée. Cependant, en raison de la protection des données, le Resel ne peut nous partager sa base de données et les mots de passe. Toutefois, une récupération du nom et prénom de l'élève connecté a été suggérée en nous donnant l'accès avec une clé. Malheureusement, malgré les relances et les mails, le Resel n'a pas donné suite à la proposition. En conséquence, l'utilisateur doit créer un compte sur l'application avec un mail et un mot de passe, après s'être connecté au Resel (FP3-Utilisateur, FC1-Utilisateur, FC2-Utilisateur).

Cette inscription se fait grâce à Firebase Authentification [3] et son option authentification par email et mot de passe. On utilise cette méthode pour faciliter la sécurisation des données de l'utilisateur. En effet, cette option de Firebase permet un hachage et salage sécurisé des mots de passe en s'appuyant sur des bibliothèques cryptographiques reconnues (FC7-Utilisateur). Elle assure aussi la récupération de mot de passe par mail si l'utilisateur l'oublie, mais surtout la gestion de sessions sécurisées avec des jetons (tokens) d'accès et de rafraîchissement. Ce sont ces jetons qui permettent à l'utilisateur de rester connecté même si l'application est fermée.

Explication de la gestion des sessions avec des jetons :

1. Connexion : l'utilisateur se connecte avec son email et son mot de passe.
2. Génération de Jetons : Firebase génère un jeton d'accès (expirant dans 1 heure) et un jeton de rafraîchissement (expirant dans 30 jours).
3. Accès à une API (Application Programming Interface) : l'utilisateur fait une requête à une API protégée avec le jeton d'accès (nécessitant d'être identifié) donc utilisation du jeton d'accès.
4. Expiration du Jeton d'Accès : Après 1 heure, le jeton d'accès expire.
5. Renouvellement du Jeton : le client utilise le jeton de rafraîchissement pour obtenir un nouveau jeton d'accès après son expiration ou utilisation.
6. Révocation : si un administrateur de la base de données révoque le jeton de rafraîchissement, l'utilisateur sera déconnecté et devra se reconnecter pour

généraliser de nouveaux jetons.

Pour changer l’aspect de l’application selon la connexion de l’utilisateur, on enveloppe l’application d’un widget “Wrapper” qui écoute les actions de l’utilisateur (voir l’image de l’architecture du logiciel ci-dessous). S’il n’est pas connecté, on le renvoie au formulaire de connexion sinon on lui fait accéder à l’application. (Annexe, figure 8)

Enfin, le statut (FP1-Président, FP2-BDX, FC1-BDX) de l’utilisateur a été ajouté dans la base de données permettant de différencier les utilisateurs “élèves”, les présidents et les membres d’un BDX. L’aspect de l’application a alors changé en fonction de l’accessibilité de l’utilisateur. Cela se fait grâce au UserProvider qui permet de récupérer les informations de l’utilisateur en tout point du code. Parmi ces informations, on a tout ce qui est relatif au statut mais aussi les données du profil de l’utilisateur. Ainsi, s’il est président de nouvelles fonctionnalités apparaissent comme l’ajout de post (Annexe, figure 9). De même, s’il est dans un BDX alors il aura accès au formulaire d’ajout d’un club. (Annexe, figure 11) Il pourra par ailleurs supprimer ou modifier un club via l’affichage du club et le formulaire de modification de club (Annexe, figure 10).

2.4 Filtres

Pour réaliser les filtres (FF1-Utilisateur) nous avons tout d’abord fait une première version de l’interface graphique avec une liste de cases à cocher pour sélectionner le filtre que l’on désire. Si une case est cochée alors le club correspondant à la case cochée est ajouté à une liste. Au départ, nous comptions faire une variable globale stockée dans l’application. Ainsi, les filtres auraient été construits sur l’application et seraient envoyés dans un deuxième temps à la base de données. Cependant, le Provider ne donnait pas la variable globale mise à jour, nous avons dû opter pour une variable directement stockée dans la base de données de l’utilisateur pour que les filtres fonctionnent. Un des inconvénients de cette méthode est le rechargement de la page à chaque ajout ou suppression d’un élément de la liste.

Pour le frontend, nous avons eu une première version fonctionnelle (niveau backend) [4] avec juste une liste déroulante des clubs (tout BDX confondus) et des checkboxes pour chaque élément que l’on souhaite cocher. Après notre réunion avec notre encadrant, nous avons décidé de modifier entièrement l’interface pour une meilleure ergonomie et expérience utilisateur. Les clubs sont à présent classés par bureaux, les en-têtes des bureaux ont leurs codes couleurs respectifs (couleurs auxquelles ils sont associés dans l’école) pour une meilleure visualisation et les filtres déjà sélectionnés sont affichés en haut de la page comme “filtre déjà sélectionnés”.

(Annexe, figure 12)

2.5 Messagerie

La messagerie (FF2-Utilisateur, FF1-Président) a été construite de la même manière que la liste des posts et que celle des clubs. Des Streambuilders sont utilisés pour construire l'interface.

Pour le frontend de la messagerie, nous avons décidé de copier le design d'une messagerie classique. Les messages envoyés par l'utilisateur sont d'une couleur différente et mis sur la droite de l'écran pour bien les différencier des messages des autres personnes (à gauche de l'écran). Il peut taper son message dans une barre de saisie et l'envoyer à la base de données grâce à un bouton. Il peut ajouter des pièces jointes (vidéo, images, documents).

2.6 Notification

Pour les notifications (FF3-Utilisateur), on doit utiliser Firebase Messaging [5] afin d'envoyer des notifications à tous les utilisateurs à chaque fois qu'un nouveau post est réalisé. Pour cela, on abonne tous les utilisateurs à un topic (un sujet), ici le topic "Posts". Ensuite, à chaque fois qu'un utilisateur appuie sur le bouton permettant d'envoyer un post alors tous ceux abonnés au topic "Posts" devraient recevoir une notification. Cependant, pour que cela fonctionne, l'utilisateur doit être considéré comme administrateur et doit avoir accès à la base de données pour pouvoir envoyer une notification à tous. Pour des raisons de sécurité, cette solution n'est pas envisageable. Une autre option a été envisagée en utilisant une option de Firebase appelée Firebase Functions qui nous permet de créer des fonctions internes à Firebase. Ainsi, on peut créer une fonction qui écoute la base de données "Posts" et, dès que celle-ci est modifiée, tous les utilisateurs devraient recevoir une notification. Cependant, cette option nécessite un compte payant sur Firebase qui prélève au bout d'une certaine utilisation. Donc, il faudrait envisager un certain budget pour l'application si l'on veut rajouter les notifications. Ainsi, les notifications n'ont pas été finalisées en raison de ces problèmes.

2.7 Lien de téléchargement

Le lien de téléchargement est directement généré sur Android Studio à travers l'option Build>Flutter>Build APK [6]. Cet APK a été placé sur un lien Google Drive accessible depuis Internet si l'utilisateur possède un compte Google (ce qui est le cas de la majorité des étudiants). Ainsi, l'application est facile à télécharger (FC3-utilisateur), rapide à installer (FC4-Utilisateur) et gratuite, car on ne passe pas par des magasins en ligne (FC6-Utilisateur).

En somme, la réalisation a pu s'écarter des maquettes établies durant la phase de conception pour obtenir une application plus intuitive (FC5-Utilisateur), mais aussi pour des raisons techniques. En effet, il a fallu parfois adapter les maquettes initiales car il était difficile de les coder exactement telles quelles, ce qui a conduit à de légères modifications. Finalement, toutes les fonctionnalités du cahier des charges (à l'exception des notifications) ont pu être réalisées. Ainsi, il ne restait plus qu'à faire les différents tests afin de valider l'application.

3 Validation

Rédactrices : Chloé et Juliette

Relectrices : Alexia et Leslie

Pour valider la réalisation de l'application, nous avons dû passer par plusieurs phases de tests. Premièrement, nous l'avons testée entre nous. Dans un second temps, nous avons organisé une bêta-test (Annexe, figure 17) et demandé à une dizaine d'étudiants d'IMT Atlantique d'essayer l'application et de tester différentes fonctionnalités. Cela nous a permis de repérer et de corriger les bugs, ainsi que de prendre en compte les améliorations à apporter à l'application pour une meilleure expérience utilisateur.

3.1 Validation des fonctionnalités accessibles par un utilisateur

FP1-Utilisateur : Lister l'ensemble des clubs présents sur le campus

En tant que membre du BDX, à l'aide d'un algorithme, 100 clubs ont été ajoutés avec leurs présidents dans la base de données des clubs initialement vide (Annexe, figure 13). Les 100 clubs ajoutés se sont bien affichés par catégorie (par BDX) et par ordre alphabétique, dans la liste des clubs. Cela a permis de vérifier que l'espace alloué par Firebase était suffisamment important pour contenir l'ensemble des clubs présents sur le campus.

Le détail du code ayant permis les tests de capacité de Firebase est en annexe (Annexe, figure 16).

Ensuite, en tant que président, puis en tant qu'utilisateur "élève", nous avons essayé d'ajouter un club. Toutefois, le bouton permettant d'ajouter un club étant inexistant, cet ajout est impossible.

FP2-Utilisateur : Afficher les posts relatifs aux activités des clubs

En étant connecté sur un compte étant président de différents clubs, il a été possible d'ajouter 20 posts en personnalisant le texte et/ou en ajoutant des fichiers. Les posts se sont bien affichés dans l'ordre anti-chronologique (le plus récent en premier), validant le cahier des charges.

FP3-Utilisateur : Permettre de s'identifier et de réaliser un profil

La création d'un profil respectant les normes a été effectuée. Ce dernier a bien été créé avec toutes les informations avec le statut "élève". Au contraire, si l'utilisateur ne renseigne pas son nom ou son prénom, alors le profil n'est pas validé, l'utilisateur ne peut pas quitter cette page sans remplir tous les champs

obligatoires.

En allant sur la page d’affichage du profil, la modification est bien accessible. Après l’avoir modifié en respectant les normes, il s’affiche correctement avec les corrections faites. Toutefois, si la modification ne respecte pas les normes, alors sa validation ne peut pas être réalisée et l’utilisateur est obligé de remplir correctement les différents éléments.

FC1-Utilisateur : Doit pouvoir être utilisé par tout élève d’IMT Atlantique du campus de Brest

À l’aide d’un algorithme comportant une boucle, 1000 comptes respectant les normes de création de profil ont été ajoutés à la base de données (Annexe, figure 14). Tous les 1000 comptes ont bien été pris en compte et leurs informations ont bien été recensées dans la base de données. Nous avons ainsi pu vérifier que Firebase et l’application supportent le nombre de comptes déterminé dans le cahier des charges.

FC2-Utilisateur : Doit être seulement accessible par des élèves d’IMT Atlantique

Nous avons essayé de nous connecter au Resel avec un identifiant et un mot de passe inexistant. Cela est impossible. Il est donc, par la suite, impossible de passer à l’étape suivante consistant à créer un compte sur notre application. Ainsi, seuls les élèves (ou alumni) d’IMT Atlantique, présents sur le campus de Brest, y ont accès.

FC3-Utilisateur : Doit pouvoir être téléchargé facilement

En utilisant un Android, l’application est bien téléchargeable à partir d’un lien. Elle apparaît sur le téléphone. Cette installation ne prend que 5 secondes (avec une connexion stable) et son espace de stockage est de 500 Mo, ce qui est bien inférieur à 1 Go.

FC4-Utilisateur : Doit être rapide/performant

Lors de l’ouverture de l’application, la première page (celle de connexion ou le fil d’actualité, en fonction de si l’utilisateur s’est déjà connecté ou non) met moins d’une seconde à charger, ce qui est inférieur à 5 secondes.

Se déplacer sur la page “liste des clubs”, puis sur le fil d’actualité et enfin sur des groupes de discussion prend moins d’une seconde. La navigation nécessite donc moins de 5 secondes.

FC5-Utilisateur : Doit être ergonomique

Lors de la bêta-test, nous avons demandé à 10 utilisateurs de trouver plusieurs informations et nous avons chronométré le temps que cela leur prenait. En

moyenne, le temps pour trouver une information est de 30 secondes et ce temps est toujours inférieur à 3 minutes, ce qui correspond aux attendus du cahier des charges. Le détail de la bêta-test se trouve en annexe.

FC6-Utilisateur : Doit être gratuit

Aucuns frais ne sont demandés à l'utilisateur lors du téléchargement de l'application.

FC7-Utilisateur : Doit protéger les données de l'utilisateur

La sécurité des mots de passe étant faite par Firebase. On ajoute des règles de sécurité pour l'accessibilité à la base de données via la page "règles" de Firestore. Firebase interdit tout accès à la base de données par défaut. Il faut donc accorder des accessibilités. Par exemple, on accorde aux utilisateurs la possibilité de lire toutes les données hors mot de passe pour pouvoir les afficher (Annexe, figure 18).

FF1-Utilisateur : Permettre à l'élève de choisir ce qui apparaît dans son fil d'actualité

Il est possible de filtrer le fil d'actualités en cliquant sur le bouton correspondant, et en sélectionnant ensuite les filtres désirés. Le fil d'actualité n'affiche ainsi plus que les posts relatifs aux clubs sélectionnés. La fonction a été testée et validée par tout type d'utilisateur.

FF2-Utilisateur : Permettre aux élèves d'avoir des groupes de discussions liés à un club

Un utilisateur peut cliquer sur le bouton "suivre" d'un club sur la liste des clubs et est ainsi ajouté au groupe de discussion. Nous avons testé l'envoi de messages comportant du texte et différents types de fichier. Lors de la bêta-test, les participants ont pu tester cela. Le temps moyen nécessaire pour rejoindre un groupe et envoyer un message (texte, image, vidéos, fichiers) était de 27 secondes et le temps maximum était d'une minute. L'application respecte donc bien les attendus du cahier des charges et les messages s'affichent correctement.

3.2 Validation des fonctionnalités accessibles par un président

L'utilisateur connecté possède à présent les accessibilités d'un président.

FP1-Président : Permettre de gérer les posts d'un club

Nous avons créé plusieurs posts pour tester les différentes possibilités de personnalisation du texte : police, taille de police, gras, italique, souligné, barré, indice, exposant, couleur de police, couleur de surlignage, styles préenregistrés (titres),

liste à numéros, liste à puces, cases à cocher, citation, augmentation ou réduction du retrait, lien hypertexte, recherche, couper, copier, coller. Ces posts se sont correctement affichés. Nous avons également testé l'ajout de fichiers : images, vidéos, documents.

Créer un post sans auteur n'est pas possible : le champ auteur est automatiquement rempli par le nom de l'utilisateur connecté. Pour ce qui est de créer un post sans titre ou sans club sélectionné, l'ajout du post n'est pas possible. Si l'on tente d'ajouter un post sans titre, un message s'affiche pour demander de compléter le champ. De même, si aucun club n'est sélectionné, l'utilisateur doit en sélectionner un dans la liste déroulante qui correspond aux clubs dont il est président.

Lorsque l'on tente d'ajouter un fichier de plus de 300 Mo à un post, le chargement ne s'effectue pas entièrement et l'application affiche une erreur de fichier non sélectionné puisque le fichier sélectionné n'est pas conforme.

FC1-Président : Doit être facile d'utilisation

Les 10 utilisateurs de la bêta-test ont dû créer un post contenant plusieurs personnalisations de texte, un lien hypertexte et une image. Le temps nécessaire pour créer et ajouter le post est en moyenne de 1 minute et 34 secondes. Le temps le plus long étant de 4 minutes (inférieur à 5 minutes), cela correspond au cahier des charges.

FF1-Président : Permettre d'ajouter des utilisateurs au groupe de discussion d'un club

A l'aide d'un algorithme comportant une boucle, 500 membres ont été ajoutés à un groupe de discussion (Annexe, figure 15). Cela a permis de confirmer la capacité de la base de données à supporter un nombre important d'utilisateurs reliés à un groupe (les 500 membres avaient bien accès au groupe.).

3.3 Validation des fonctionnalités accessibles par un membre d'un BDX

L'utilisateur connecté est maintenant un membre d'un BDX.

FP1-Membre BDX : Permettre de créer un club et modifier

Un membre du BDX peut créer un club en renseignant le président du club. S'il ne le renseigne pas et tente d'ajouter le club, la page affiche un message demandant de sélectionner un président et le club ne sera pas ajouté tant que ce champ n'est pas rempli. De même, si le membre du BDX tente d'ajouter un club en laissant le champ correspondant au nom vide, l'application affiche un message demandant de compléter ce champ.

Un membre d'un BDX peut modifier ou supprimer n'importe quel club associé à son BDX.

FP2-Membre BDX : Attribuer des rôles de présidents de clubs aux utilisateurs concernés

Il est possible pour l'utilisateur de modifier le président associé au club, mais pas de supprimer tout président de ce club. S'il tente de modifier un club sans renseigner un président, de nouveau, le club n'est pas modifié et l'utilisateur est informé de la nécessité d'en sélectionner un.

Un membre du BDX peut donc supprimer le titre de président d'un utilisateur mais seulement en le transmettant à quelqu'un d'autre. Qu'il y ait un coprésident renseigné ou non, la modification du club n'est pas prise en compte tant que le champ "Président" n'est pas complété.

Il peut aussi supprimer le club. Le président de ce club perd alors automatiquement son titre de président (de ce club) et le groupe de discussion associé est bloqué.

FC1-Membre BDX : Doit être facile d'utilisation

Pour tester l'accessibilité de la création et modification de clubs, nous avons évalué le temps nécessaire pour créer un club et modifier ses présidents. Les participants de la bêta-test ont mis en moyenne 1 minute et 2 secondes pour effectuer cela, et toujours dans un temps inférieur à 2 minutes (ce qui est inférieur à 5 minutes), ce qui valide les critères du cahier des charges.

Les différents tests et validations ont donc permis la validation de l'application et des différentes fonctionnalités présentes dans le cahier des charges (à l'exception des notifications qui n'ont pas pu être réalisées). Il reste maintenant à la déployer. Pour ce fait, les contraintes, les objectifs et l'organisation prévue à cela sont expliqués dans la suite de ce rapport.

4 Déploiement

Rédactrice : Alexia

Relectrice : Chloé

4.1 Nécessités pour le déploiement

Pour répondre au besoin de circulation de l'information, le déploiement de l'application au sein de l'école est un élément primordial. En effet, les élèves d'IMT

Atlantique doivent globalement adopter l'application pour que celle-ci ne devienne pas un n-ième médium de communication ignoré de tous. Le but serait qu'elle remplace Messenger, Whatsapp, email, Instagram ou tout autre application, dans le cadre de la vie associative. Rassembler le plus d'étudiants sur ce canal permettrait une réelle centralisation de l'information et une communication plus efficace.

4.2 Contraintes de déploiement

Afin que l'ensemble des étudiants accepte et intègre cette application mobile nous avons quelques contraintes à respecter quant à l'accessibilité de l'application. Celle-ci doit être accessible par tous, peu importe le système d'exploitation de son téléphone (IOS et Android). En choisissant Flutter/Firebase comme langage de programmation nous pouvons déployer AtClub sur IOS et Android et ainsi couvrir quasiment 100% de l'effectif étudiant d'IMT Atlantique. En effet, Flutter utilise principalement du Dart et contient un compilateur Dart-to-native permettant d'interagir directement avec les langages natifs respectif de Android et IOS (Kotlin et Swift). Pour l'installation de l'application nous avons pensé à plusieurs options. Dans un premier temps, nous avons envisagé de passer par des services AppStore ou Google Play Store pour mettre en ligne l'application. Les démarches à suivre sont les suivantes :

Sur Google Play Store :

- Créer un compte développeur et obtenir une licence coûtant 25€ (**impossible pour nous**)
- Vérifier que l'application respecte les politiques et directives de Google Play Store
- Implémenter l'application et attendre confirmation de la part de Google Play

Sur App Store :

- Avoir une licence développeur coûtant 99€/an (**impossible pour nous**)
- Vérifier que l'application respecte les politiques et directives de App Store
- Faire vérifier l'application par App Store

Dans le cadre du projet Codev, ces licences ne nous seront pas accessibles. Nous devons donc trouver un autre moyen de les déployer. Celui envisagé et abordé dans le cahier des charges est un lien APK (Android) et IPA (IOS).

4.3 Objectifs de déploiement/ lien de téléchargement

Comme spécifié dans la fonction FC3 du cahier des charges, nous avons créé des liens de téléchargement de l'application qui sera hébergée sur un site internet. Chaque lien héberge respectivement un APK et un IPA.

Nous avons fait un test d'upload et de téléchargement de l'APK généré avec Android (Annexe, figure 19) qui fut un succès. En téléchargeant, l'APK il suffit d'activer "l'installation des applications inconnues" sur Android et l'application s'installe rapidement.

4.4 Plan de déploiement à la rentrée de septembre

A la rentrée de septembre, nous souhaitons déployer notre application au sein de l'établissement. Pour cela nous avons plusieurs stratégies d'approche. En effet, pour que l'application soit adoptée par la majorité des élèves d'IMT Atlantique du campus de Brest, les présidents et BDX doivent être les moteurs du mouvement. C'est pourquoi nous envisageons une réunion avec eux pour discuter du déploiement de l'application au sein des clubs et de la vie associative. Notre encadrant nous proposait de faire une conférence pour les élèves afin de leur présenter notre application, ceci se ferait alors dans un second temps.

Enfin un dernier levier de communication serait de mettre des affiches faisant la promotion de l'application, et contenant un QR Code de téléchargement puis d'en parler sur les réseaux sociaux de l'école.

Conclusion

Rédactrices : Alexia et Leslie

Relectrices : Chloé et Juliette

Synthèse des résultats

Pour conclure, l'application répond aux différentes attentes fixées dans le cahier des charges. Tout d'abord, la fonctionnalité d'authentification permet seulement de donner accès à l'application aux élèves d'IMT Atlantique du campus de Brest (les clients du projet) ou alumni, ce qu'on a vérifié avec le test FC2-Utilisateur. L'application permet aussi de visualiser et de regrouper l'ensemble des clubs de l'école sur une plateforme, comme le montrent les tests sur la liste des clubs (FP1 et FP2) et l'accessibilité aux groupes de messages (FF2-Président). Ainsi elle répond à la problématique de centralisation émise par le client. En effet, l'utilisateur peut avoir accès aux informations primordiales suivantes : quels sont les clubs, qui sont les présidents, quelles sont les actualités de tous les clubs, que se dit-il entre les membres des clubs (groupes de messages). D'autre part, nous avons cherché à rendre la connaissance de l'existence des clubs facile pour l'utilisateur. En effet, l'interface de AtClub permet d'avoir accès à la liste des clubs, aux posts et aux groupes de messages des clubs via une seule barre de navigation, qui imite le fonctionnement des applications de réseaux sociaux classiques. La navigation est intuitive et fluide, ce qui apporte une plus-value au critère de facilité d'accès à l'information.

Enfin, la gestion d'adhésion à un club est facilitée à travers un bouton "suivre" directement accessible sur la liste des clubs. Cependant, suite aux tests réalisés sur l'application avec 10 personnes extérieures au développement, nous avons pu voir que l'application est intuitive mais l'utilisation d'un rapide tutoriel pourrait être bénéfique.

Enfin, pour faciliter l'installation et assurer la gratuité de l'application, nous avons pu remarquer à travers les tests que le déploiement de l'application doit se borner au système Android. En effet, l'installation de l'application se fait grâce à un lien et cette dernière respecte la taille limite de stockage, ce qui était prévu dans le cahier des charges.

Ainsi, l'utilisateur a une expérience fluide et simple de l'installation de l'application jusqu'à son utilisation.

Perspectives du projet

Une des perspectives envisagées dans le court terme pour le projet serait de réaliser un tutoriel d'utilisation lors de la première ouverture de l'application. Ainsi cela rendrait l'expérience utilisateur plus agréable et moins déroutante. Une des autres suggestions qui pourrait être implémentée au moyen terme serait l'archivage

des clubs disparus. En effet, si un club n'est plus utilisé par les étudiants, l'application permet aux responsables BDX de supprimer le club de la base de données. Or cette méthode génère des bugs et oblige les étudiants à recréer un club si celui-ci est repris plus tard, alors qu'une création antérieure existait déjà. L'application devrait garder des traces des clubs inactifs tout en cachant leur affichage dans la liste des clubs ou en indiquant qu'ils n'ont pas été repris par un nouveau président. Enfin une dernière piste d'amélioration sur le long terme serait de rendre l'application téléchargeable par des moyens légaux et vérifiés comme l'AppStore ou le Google Play Store. Cette amélioration nécessitera l'aide financière de l'école pour obtenir une licence pour chaque magasin en ligne.

Références

- [1] Flutter. *StreamBuilder<T> class*. [En ligne], Disponible sur : <https://api.flutter.dev/flutter/widgets/StreamBuilder-class.html>, (consulté le 25 mars 2024).
- [2] Mitch KOKO. *TAB BAR • Flutter Widget of the Day 13*. [En ligne], Disponible sur : <https://www.youtube.com/watch?v=XSheN4Lkhpc>, (consulté le 28 mars 2024).
- [3] Firebase. *Authentication Firebase*. [En ligne], Disponible sur : <https://firebase.google.com/docs/auth?hl=fr>, (consulté le 5 avril 2024).
- [4] Techness With Hussain. *How to add Filters to Flutter App || Convert front-end data into Filters*. [En ligne], Disponible sur : <https://www.youtube.com/watch?v=WCbYkfEfQeU&t=495s>, (consulté le 10 mai 2024).
- [5] Firebase. *Firebase Cloud Messaging*. [En ligne], Disponible sur : <https://firebase.google.com/docs/cloud-messaging?hl=fr>, (consulté le 14 mai 2024).
- [6] Android. *Compiler et exécuter votre application*. [En ligne], Disponible sur : <https://developer.android.com/studio/run?hl=fr>, (consulté le 21 mai 2024).

Annexe

Images

Table des figures

1	Architecture de notre application	4
2	Schéma conceptuel de la base de données	
3	Palette de couleur de l'application	
4	Maquette	
5	Accessibilité	
6	Architecture code	
7	Evolution liste des clubs	
8	Architecture authentification	
9	Changement de l'interface selon le statut de l'utilisateur, statut président	
10	Changement de l'interface selon le statut de l'utilisateur, ici statut de membre du BDX	
11	Changement de l'interface selon le statut de l'utilisateur	
12	filtre	
13	Ajout de 100 clubs	
14	Ajout de 1000 comptes	
15	Ajout de 500 membres à un groupe de discussion	
16	Code pour les tests de capacité	
17	Résultats de la bêta-test	
18	règle de sécurité	
19	Lien de téléchargement APK	
20	Tableau récapitulatif	

Images de la conception

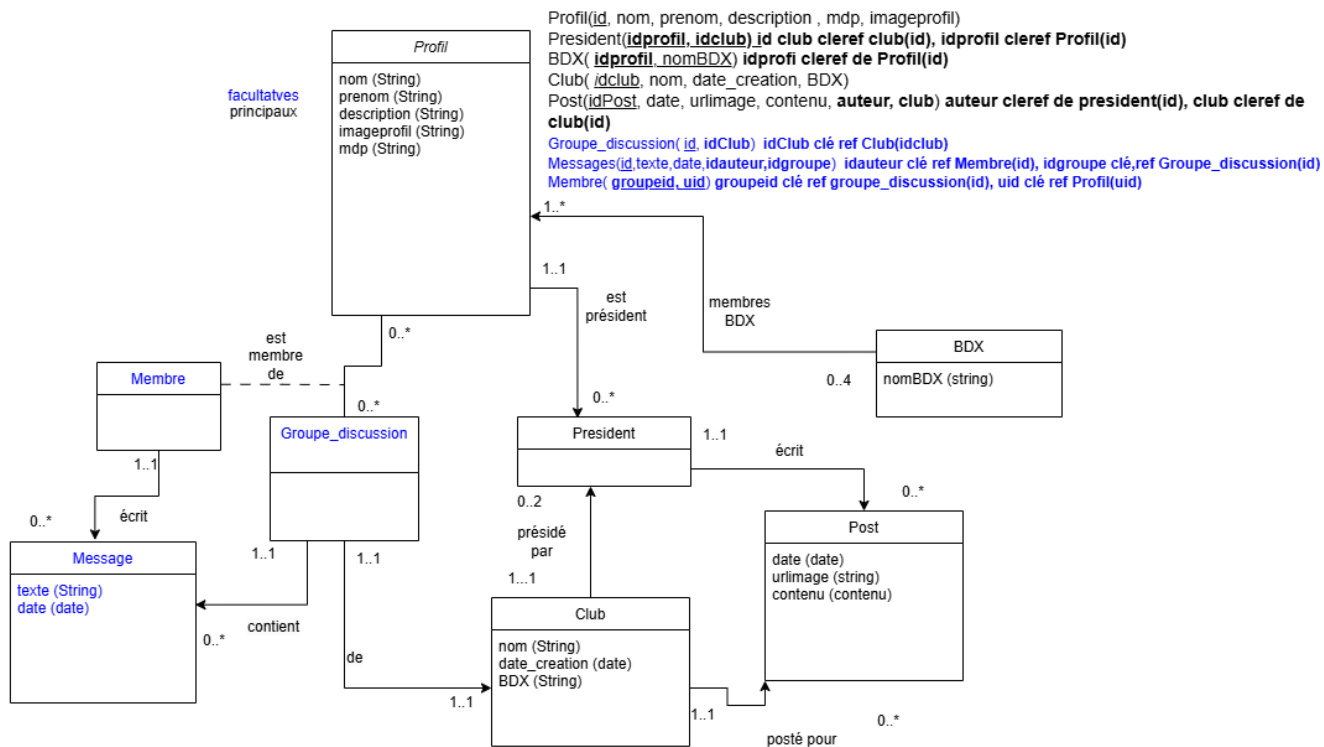


FIGURE 2 – Schéma conceptuel de la base de données

TEST PALETTE DE COULEUR



FIGURE 3 – Palette de couleur de l'application

Maquette de l'interface 'Fil d'actualités'. Le design est simple et fonctionnel, utilisant des rectangles pour structurer l'information.

- Barre de navigation supérieure :** Contient un menu hamburger (trois lignes horizontales) à gauche et le titre 'Fil d'actualités' à droite.
- Contenu principal :**
 - Article 1 :** Un rectangle contenant un champ 'TITRE', un champ 'auteur' à gauche et un champ 'Date' à droite, suivis d'un rectangle 'Description'.
 - Article 2 :** Un rectangle contenant un champ 'TITRE', un champ 'auteur' à gauche et un champ 'Date' à droite.
- Barre de navigation inférieure :** Deux boutons, 'Home' à gauche et 'Clubs' à droite.

FIGURE 4 – Maquette

Voici un tableau résumant les différentes pages (ou formulaire) présentes sur l'application ainsi que leur accessibilité. Les rectangles regroupant différentes pages symbolisent le fait que pour avoir accès à certaines fonctionnalités, il faut se trouver sur une page précise. Cette page correspond à la première fonctionnalité présente dans le rectangle. Par exemple, pour le premier rectangle, le formulaire de création d'un post (accessible seulement par un président de club) et celui pour les filtrer (accessible par tous) se trouvent sur la page "Fil d'actualités" (accessible par tous).

ATCLUB

Accessibilité des pages

PAGES / FORMULAIRES	ÉLÈVE	PRÉSIDENT	MEMBRE D'UN BDX
Fil d'actualités	✓	✓	✓
Création d'un post sur le fil d'actualité		✓	
Filtre sur le fil d'actualités	✓	✓	✓
Liste des clubs	✓	✓	✓
Ajout d'un club			✓
Suppression d'un club			✓
Modification d'un club			✓
Suivre un club	✓	✓	✓
Authentification	✓	✓	✓
Formulaire d'inscription	✓	✓	✓
Création de son profil	✓	✓	✓
Affichage de son profil	✓	✓	✓
Modification de son profil	✓	✓	✓
Recherche d'un profil	✓	✓	✓
Modification d'un statut			✓
Messagerie	✓	✓	✓
Envoie de messages	✓	✓	✓
Ajout d'une personne		✓	
Suppression d'une personne		✓	

Vérification par
le Resel

FIGURE 5 – Accessibilité

Images de la réalisation

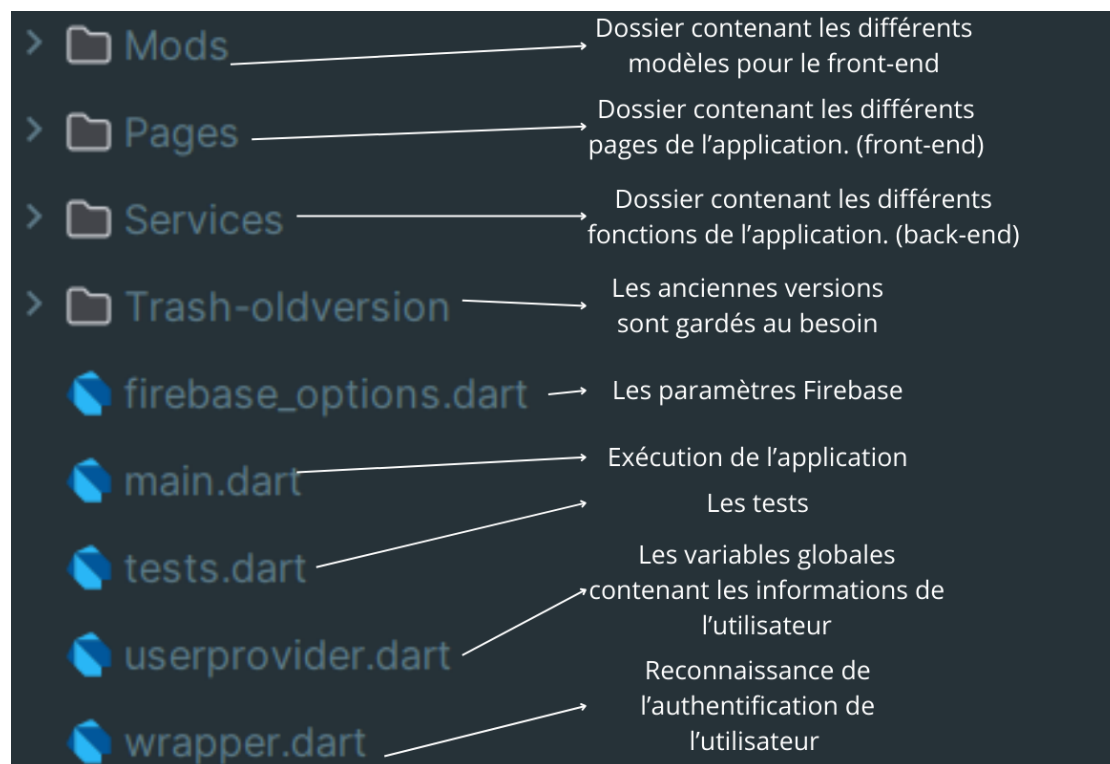


FIGURE 6 – Architecture code

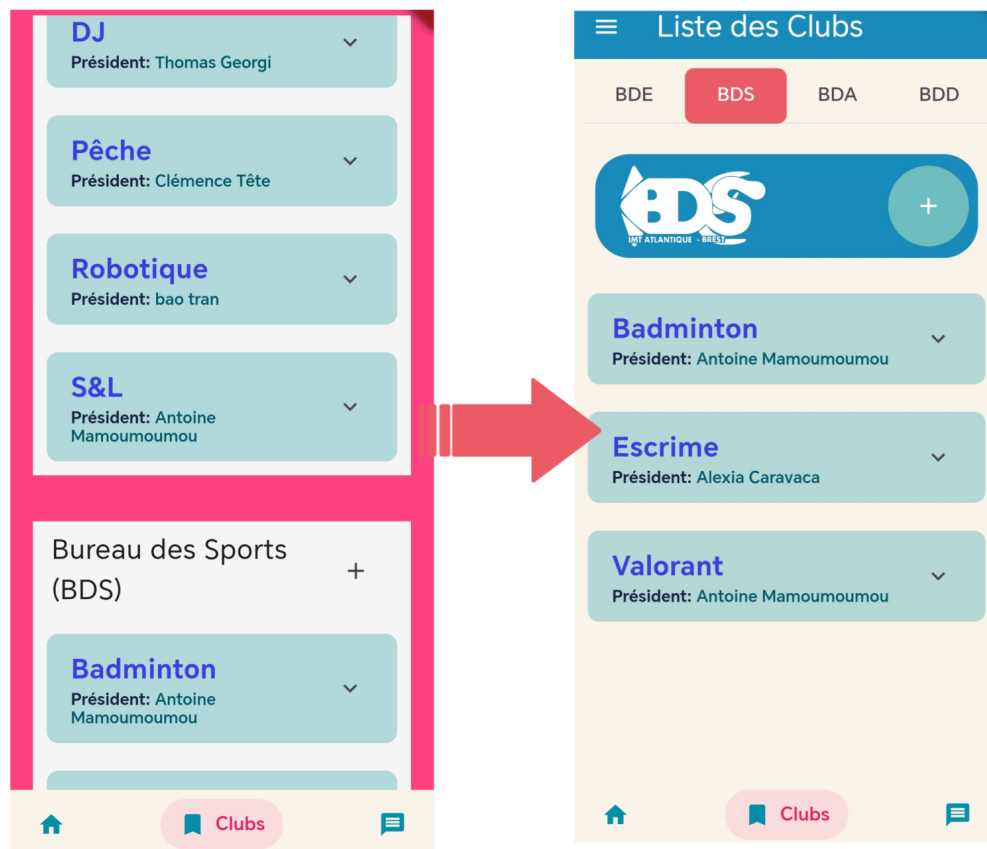


FIGURE 7 – Evolution liste des clubs

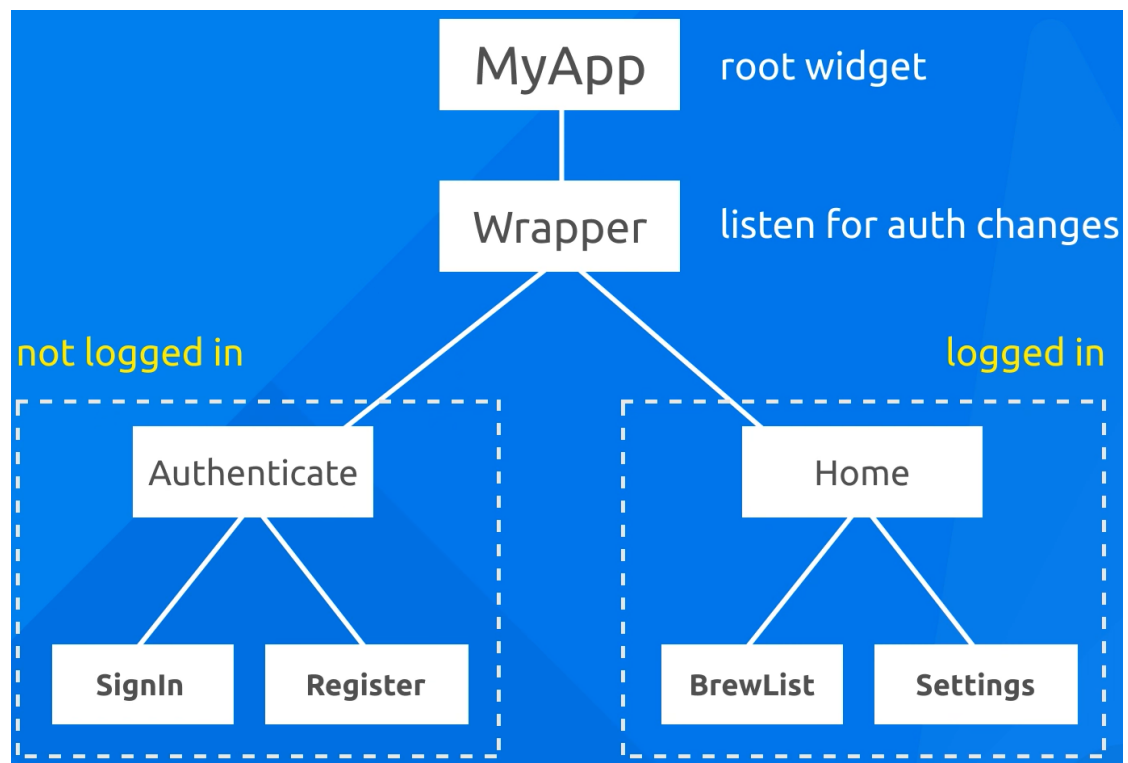
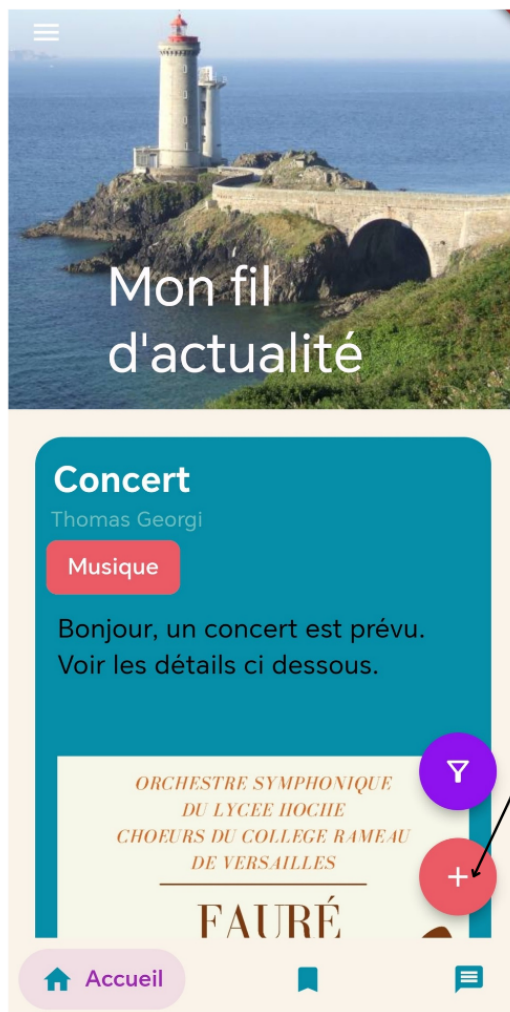


FIGURE 8 – Architecture authentication



Bouton pour ajouter un post

FIGURE 9 – Changement de l'interface selon le statut de l'utilisateur, statut président

Avec droit des BDX

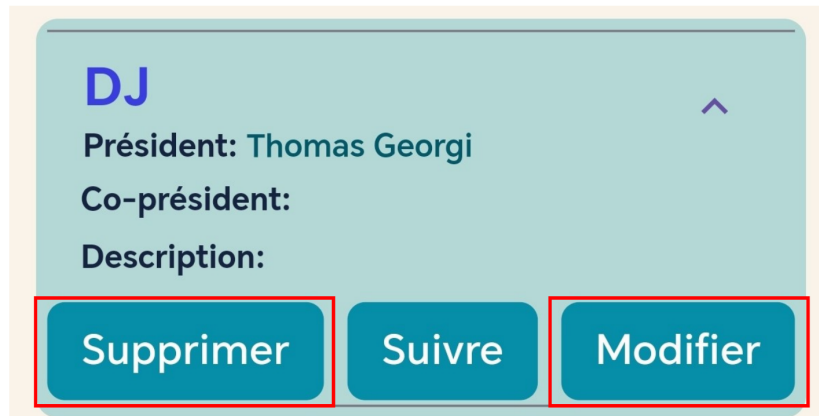


FIGURE 10 – Changement de l'interface selon le statut de l'utilisateur, ici statut de membre du BDX



FIGURE 11 – Changement de l'interface selon le statut de l'utilisateur

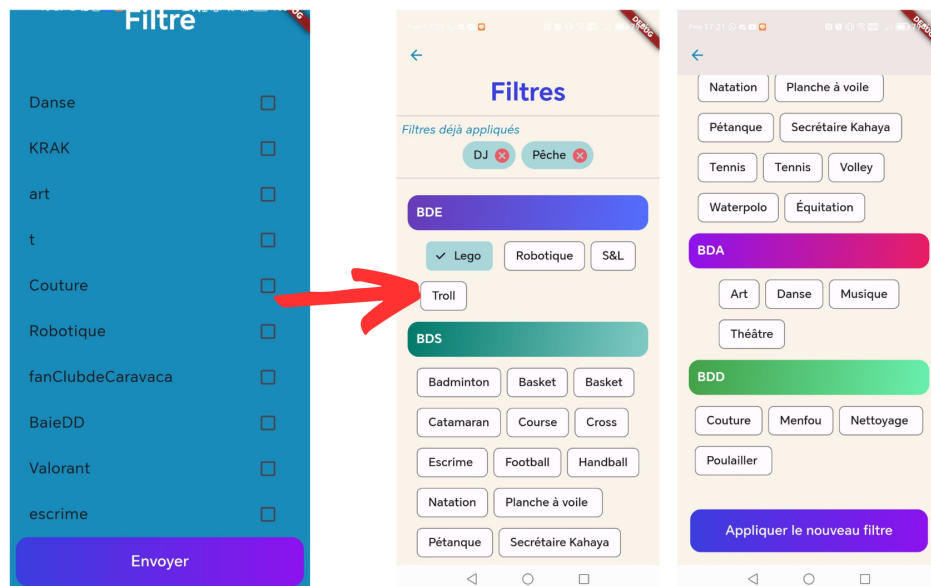


FIGURE 12 – filtre

Images de la validation

 (default)	 Liste_clubs  	 0Ej6VnqW1loitVlelXBc 
+ Commencer une collection BDX Groupes Liste_clubs > Membres_Groupes Posts Presidents Profils messages	+ Ajouter un document 22 mai 2024 à 10:51:22 UTC+2 0Ej6VnqW1loitVlelXBc > 22 mai 2024 à 10:51:22 UTC+2 ByGgv8UABOLCCqAcMFOT 22 mai 2024 à 10:51:22 UTC+2 ATwKQ8PwJvo4rnsZ2Qa 22 mai 2024 à 10:51:22 UTC+2 eFeapL5F5w8vlyh1xLJV 22 mai 2024 à 10:51:22 UTC+2 ZvMS1RK37EJTaa5mtzE6 22 mai 2024 à 10:51:22 UTC+2 prvVWEZyQB14jfb3Cew4 22 mai 2024 à 10:51:22 UTC+2 1GTfodpuB4vUF3VNY931 22 mai 2024 à 10:51:22 UTC+2 NSXnussMgA85O9xJDPjs 22 mai 2024 à 10:51:22 UTC+2 VXaHBLQ44tfagnYin42U 22 mai 2024 à 10:51:22 UTC+2 C19uTGNJ499vVdphZe93 22 mai 2024 à 10:51:22 UTC+2 LlPznEjyBg3w7VoYTDMP 22 mai 2024 à 10:51:22 UTC+2 GzbMjNukb8jUTxC1Fb3m 22 mai 2024 à 10:51:22 UTC+2 yLB1ChRDntF8M43TkQmW 22 mai 2024 à 10:51:22 UTC+2 nEOGVH2hHRL8f6AeKPy	+ Commencer une collection + Ajouter un champ BDX: "BDD" date_creation: 22 mai 2024 à 10:51:22 UTC+2 description: "description" nom: "100"

FIGURE 13 – Ajout de 100 clubs

 (default)	 Profils  	 1000 
+ Commencer une collection BDX Groupes Liste_clubs Membres_Groupes Posts Presidents Profils > messages	+ Ajouter un document "1" 1 "10" 10 "100" 100 "1000" 1000 > "101" 101 "102" 102 "103" 103 "104" 104 "105" 105 "106" 106 "107" 107 "108" 108 "109" 109 "11" 11	+ Commencer une collection + Ajouter un champ clubfiltre description: "description" email: "1000.mail.com" imageprofil: "" nom: "1000" prenom: "1000" statut: null uid: "1000"

FIGURE 14 – Ajout de 1000 comptes

(default) + Commencer une collection BDX Groupes Liste_clubs Membres_Groupes Posts Presidents Profils messages	Membres_Groupes + Ajouter un document projects/atclub-c780c/databases/(defau...> WZMvIDYaztJGP9eQk92A projects/atclub-c780c/databases/(defau...> 85q24vuDcXh9eptffQh5 projects/atclub-c780c/databases/(defau...> 67wYtmDStWj5qfT7HhJH projects/atclub-c780c/databases/(defau...> egPf8ILxtkqXI5KYrGX projects/atclub-c780c/databases/(defau...> LY1vAhtYWTaxRY13MqKa projects/atclub-c780c/databases/(defau...> wrmkUATN1yMbJXAh677F projects/atclub-c780c/databases/(defau...> N83GhxeSoyBHF4fzQZz projects/atclub-c780c/databases/(defau...> tHvqu1P1GQHFr3x9Ktbz projects/atclub-c780c/databases/(defau...> M8Xx8ZztVjJULFu0M0z1 projects/atclub-c780c/databases/(defau...> 2ve309vvK6u9eZD0xvQu projects/atclub-c780c/databases/(defau...> u7F7a29qzD1XEkg18pC projects/atclub-c780c/databases/(defau...> 1Nqeb0dv9GcI8h08mvS1 projects/atclub-c780c/databases/(defau...> Hs8nkvmxS1Tdn1Ir1g32 projects/atclub-c780c/databases/(defau...> wNuXZ019eHs7L6A9182m	WZMvIDYaztJGP9eQk92A + Commencer une collection + Ajouter un champ groupe: /Groupes/04ttf6lDWv5CAoUfqmrN uid: /Profils/500
--	---	---

FIGURE 15 – Ajout de 500 membres à un groupe de discussion

```
tests.dart x
4 import 'package:atlub/Services/groupeSMessagerie.dart';
5
6 import 'userprovider.dart';
7 import 'package:atlub/wrapper.dart';
8 import 'package:flutter/material.dart';
9 import 'package:provider/provider.dart';
10 import 'package:firebase_core/firebase_core.dart';
11 import 'firebase_options.dart';
12 import 'package:atlub/Services/groupeSMessagerie.dart';
13
14 Future<void> tests() async {
15   DatabaseService data = DatabaseService();
16   GroupesMessagerie grp = GroupesMessagerie();
17
18   //Ajout 1000 comptes à Profils
19   for (int i=1; i<=500; i++){
20     data.addProfile("$i", "$i", "$i", "description", null, "", "$i.mail.com");
21   }
22
23   //Ajout 500 comptes à un groupe "ART"
24   for (int i=1; i<=500; i++){
25     grp.addUserToGroup("04ttf6IDWv5CAoUfgmrN", "$i");
26   }
27
28   //Ajout de 100 clubs à la base de données
29   for (int i=1; i<=100; i++){
30     data.addClub("BDD", "$i", "description");
31   }
32 }
```

FIGURE 16 – Code pour les tests de capacité

Bêta-test

Production de la liste test:

tests à valider:

- 10 personnes doivent créer un compte **0**
- 10 personnes doivent trouver une information valide en moins de 5 minutes (cdc)
 - Liste des informations à trouver:
 - 1.a. Qui est le/la président(e) du club Waterpolo? (liste clubs)
 - 1.b. Quelles sont les clubs dont Alexis Caracawa est président(e)? (recherche profil)
 - 1.c. De quelles BDSs fait partie Othniel Doussan? (recherche profil)
 - 1.d Le club d'atome a-t-il posté hier? (chronologie des posts)
 - 1.e Combien de posts le club couture a-t-il posté en tout? (filtrage)
 - 1.f Quelle est le dernier événement organisé par le club musique? (filtrage/ post chronologie)
- 10 personnes doivent poster en moins de 5 minutes. (cdc)
 - Conditions à respecter:
 - 2 Le post doit contenir du texte en gras, en italique, souligné, en couleur, surfiné en jaune, banni
 - Le post doit aussi contenir un lien vers le site du RAK, du texte en trois tailles différents , deux polices
 - Le post doit contenir une image -> [voir le modèle de post à reproduire](#)
- 10 personnes doivent ajouter un Club en moins de 5 minutes (cdc)
 - Conditions à respecter:
 - 3 Le club doit être au BDS et le président doit être bien renseigné, le co-président peut être vide
- 10 personnes doivent écrire sur un groupe de messages
 - Condition à respecter:
 - 4 Regardez le club d'écriture et envoyez un message de salutation aux autres membres .

Profils recherchés: - élèves de IMT Atlantique n'ayant jamais testé l'application avant

Résultats tests

	non-prenoms/cléps	0 1.a	1.b	1.c	1.d	1.a	1.f	2	3	4	temps moyen
1	TRAN Bao Chau	00:48	00:12	00:40	00:08	00:13	00:41	00:08	00:59	00:55	00:45 5min 30
2	DEMARE Alexandre	01:03	00:13	00:09	00:33	00:37	00:54	00:18	02:12	01:20	00:21 7min40
3	MANOU Antoine	02:16	00:27	00:14	00:13	00:29	00:32	00:20	01:50	00:52	00:22 7min31
4	DE LA BARRE Clémence	03:35	01:00	00:23	00:14	00:27	02:09	00:12	04:00	01:17	01:13 14min33
5	TETE Clémence										
6	VEIGNANT Suzanne	01:13	00:20	00:28	00:14	00:39	00:30	00:21	01:51	01:47	00:30 7min33
7	GEORGI Thomas	00:48	00:19	00:14	00:12	00:57	00:39	00:10	01:10	01:21	00:08 5min55
8	FLUCH Astid	00:45	00:57	00:11	00:12	01:02	00:15	00:07	00:30	00:16	00:20 4min40
9	DUPASQUIER Thibaut	01:09	00:10	00:15	00:10	01:09	01:26	00:13	02:05	00:35	00:22 7min1
10	BOTTIN Etude	01:03	00:52	00:13	00:18	00:21	02:52	00:08	02:26	00:36	00:04 6min3
11	DERRIEN Luc	01:52	00:27	00:22	00:16	00:37	00:43	00:13	01:20	01:15	00:30 7min33
12											

Planning testeur

	dimanche	lundi	dimanche	lundi	
TRAN Bao Chau	x	x			
DEMARE Alexandre	x	x	19h	Clémence	
MANOU Antoine	x		15h20	Suzanne	20h20
DE LA BARRE Clémence			10h40	Alexandre	20h40
TETE Clémence		x	16h	Etude	
VEIGNANT Suzanne			16h20	Thibaut	
GEORGI Thomas			16h40	Antoine	
FLUCH Astid			17h	Luc	
DUPASQUIER Thibaut	x	x	17h20	Bao	
BOTTIN Etude	x	x	17h40	Thomas	
DERRIEN Luc	x	x	18h	Astid	

titre comme on veut

titre

écriture

Grand
Enorme
Enorme (monospace)

FIGURE 17 – Résultats de la bêta-test

```

service cloud.firestore {
  match /databases/{database}/documents {
    // Collection Profils
    match /Profils/{uid} {
      // Un utilisateur non connecté peut créer un profil
      allow create: if request.auth == null;
      // Un utilisateur connecté peut lire tous les profils
      allow read : if request.auth == null;
      // Un utilisateur peut écrire et modifier seulement son propre profil
      allow write: if request.auth != null && request.auth.uid == uid;
    }
    // Collection BDX
    match /BDX/{nomBDX} {
      // Un utilisateur connecté peut lire tous les BDX
      allow read: if request.auth != null;
      // Aucune modification n'est permise
      allow write: if false;
    }
    // Collection Groupe
    match /Groupe/{groupeId} {
      // Un utilisateur connecté peut lire tous les groupes
      allow read: if request.auth != null;
      // Un utilisateur doit être président pour écrire un groupe
      allow create: if request.auth != null && exists(/databases/{database}/documents/Presidents/{request.auth.uid});
      // Un utilisateur membre du groupe peut mettre à jour les messages du groupe
      allow update: if request.auth != null && exists(/databases/{database}/documents/Membres_groupe/{request.auth.uid});
    }
    // Collection Liste_clubs
    match /Liste_clubs/{clubId} {
      // Un utilisateur connecté peut lire tous les clubs
      allow read: if request.auth != null;
      // Un utilisateur doit être au BDX pour écrire dans Liste_clubs
      allow create: if request.auth != null && exists(/databases/{database}/documents/BDX/{request.auth.uid});
      // Un président peut mettre à jour les clubs
      allow update: if request.auth != null && exists(/databases/{database}/documents/Presidents/{request.auth.uid});
    }
    // Collection Membres_groupe
    match /Membres_groupe/{membreId} {
      // Un utilisateur peut lire les membres_groupe seulement si l'id ajouté est le sien
      allow read : if request.auth != null;
      // Un utilisateur peut écrire les membres_groupe seulement si l'id ajouté est le sien
      allow create: if request.auth != null && request.resource.data.uid == request.auth.uid;
      // Un utilisateur ne peut pas supprimer ou modifier les membres_groupe
      allow delete, update: if false;
    }
    // Collection Posts
    match /Posts/{postId} {
      // Un utilisateur connecté peut lire tous les posts
      allow read: if request.auth != null;
      // Un utilisateur peut écrire un post s'il est président ou au BDX
      allow create: if request.auth != null && (exists(/databases/{database}/documents/Presidents/{request.auth.uid}) || exists(/databases/{database}/documents/BDX/{request.auth.uid}));
      // Un utilisateur ne peut pas mettre à jour ou supprimer les posts
      allow update, delete: if false;
    }
    // Collection Presidents
    match /Presidents/{presidentId} {
      // Un utilisateur connecté peut lire tous les présidents
      allow read: if request.auth != null;
      // Un utilisateur doit être dans le BDX pour écrire ou supprimer un président
      allow create, delete: if request.auth != null && exists(/databases/{database}/documents/BDX/{request.auth.uid});
      // Un utilisateur ne peut pas mettre à jour un président
      allow update: if false;
    }
    // Collection messages
    match /messages/{messageId} {
      // Un utilisateur connecté peut lire tous les messages
      allow read: if request.auth != null;
      // Un utilisateur peut écrire un message s'il est membre du groupe
      allow create: if request.auth != null && exists(/databases/{database}/documents/Membres_groupe/{request.auth.uid});
      // Un utilisateur peut mettre à jour les messages s'il est membre du groupe
      allow update: if request.auth != null && exists(/databases/{database}/documents/Membres_groupe/{request.auth.uid});
      // Un utilisateur ne peut pas supprimer les messages
      allow delete: if false;
    }
  }
}

```

FIGURE 18 – règle de sécurité

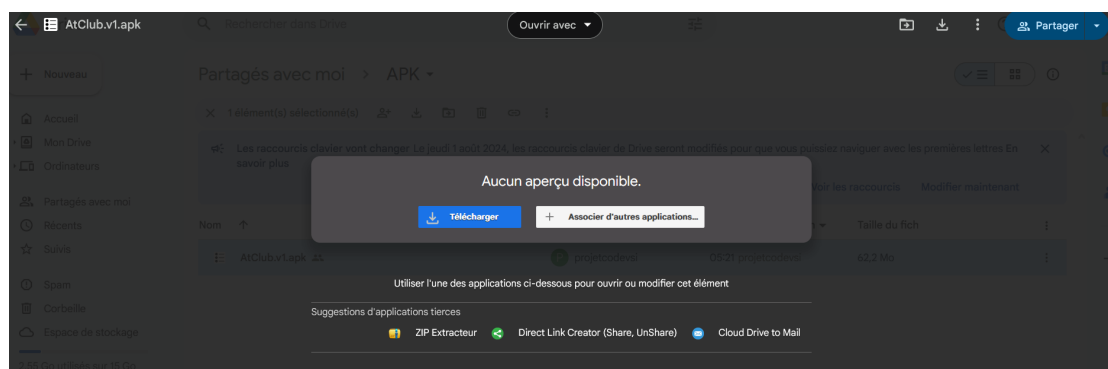


FIGURE 19 – Lien de téléchargement APK

Planning

Rédactrices : Chloé et Juliette

Relectrices : Alexia et Leslie

Le planning a été suivi sans problème jusqu'à la version des filtres (version 2). Nous avons eu quelques difficultés lors de la réalisation de cette fonctionnalité, ce qui nous a mis en retard. Afin de rattraper ce retard, nous nous sommes divisées le travail dans le but de réaliser la messagerie (version 3) en parallèle. Cela nous a permis de finir de coder les groupes de messages dans les temps. Cependant, la réalisation des notifications (fin de la version 3) a posé problème et nous n'avons finalement pas été capable de la coder sans avoir à payer un service supplémentaire sur Firebase. La recherche infructueuse d'une solution non payante, nous a fait perdre 6 jours. En travaillant le week-end, nous avons pu commencer la fin du projet le 21 mai au lieu du 17 mai et ainsi réaliser la validation de l'application le 25 mai au lieu du 21 mai.

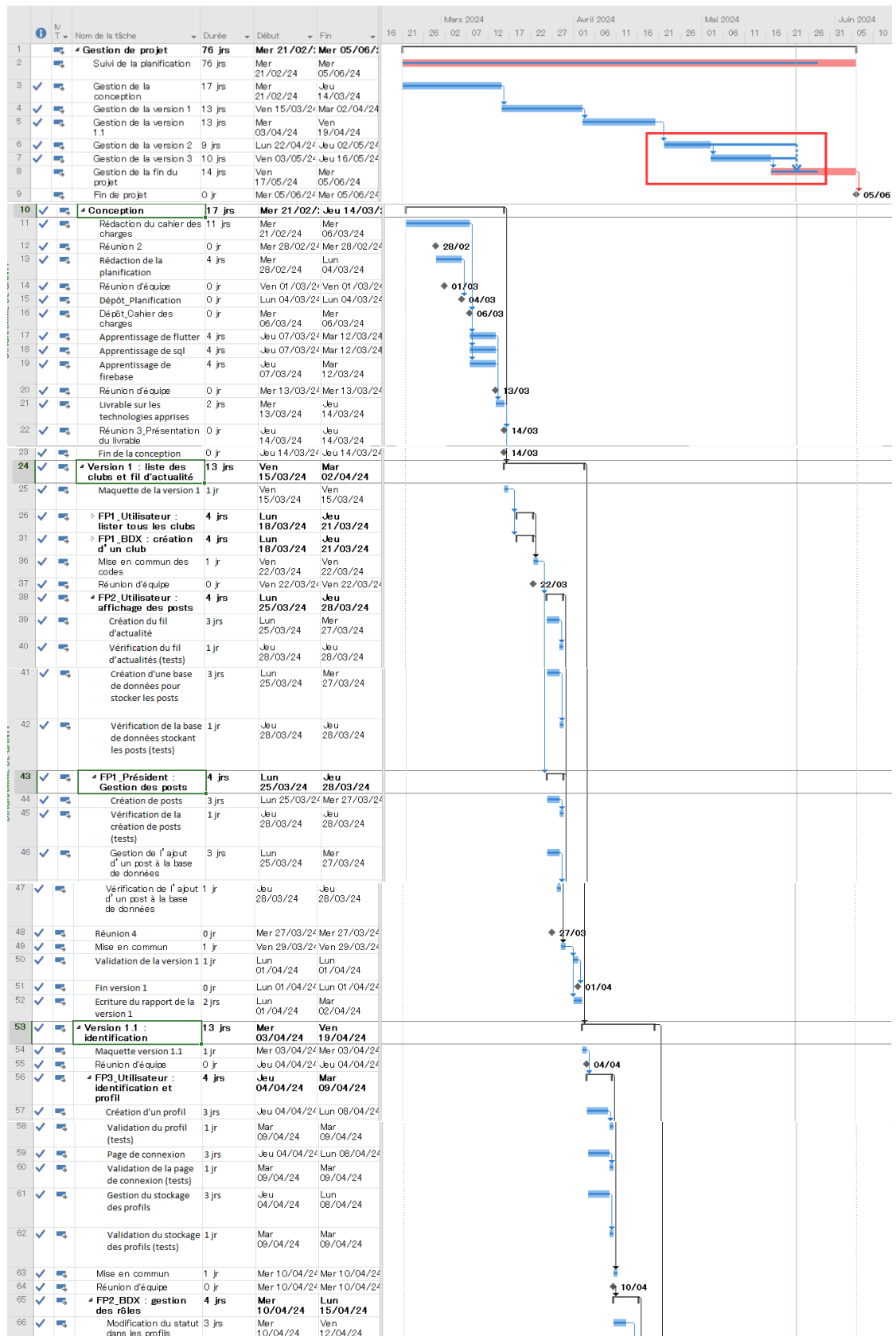
Ainsi, avec seulement 4 jours de retard, nous avons produit une application répondant au cahier des charges (à l'exception des notifications (fonctionnalité facultative) non réalisée). L'application est donc fonctionnelle.

Voici le tableau récapitulatif comparant le planning prévisionnel et celui réel :

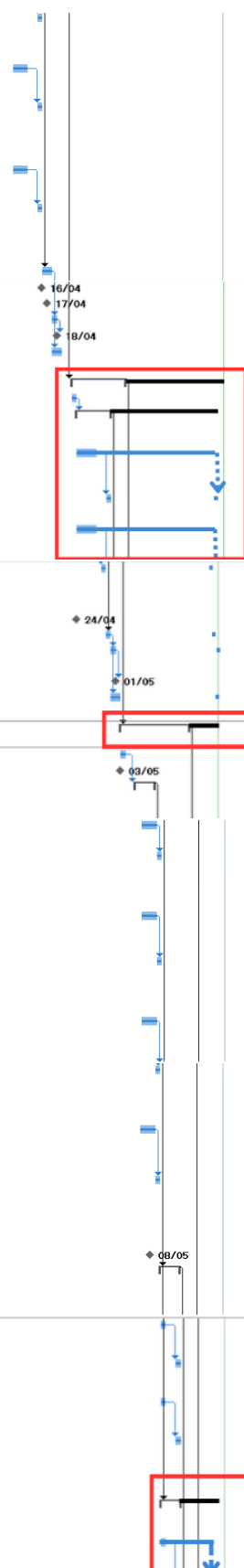
	Dates prévues	Dates réelles
Gestion de la conception	21/02 - 14/03	21/02 - 14/03
Gestion de la version 1	15/03 - 02/04	15/03 - 02/04
Gestion de la version 1.1	03/04 - 19/04	03/04 - 19/04
Gestion de la version 2	22/04 - 02/05	22/04 - 21/05
Gestion de la version 3	03/05 - 16/05	03/05 - 21/05
Gestion de la fin du projet	17/05 - 05/06	21/05 -

FIGURE 20 – Tableau récapitulatif

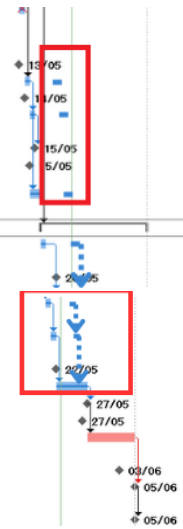
Voici la comparaison détaillée entre le planning prévisionnel (rectangle plein clair) et celui réel (trait foncé). Les rectangles qui ne sont pas pleins indiquent les lieux où le planning prévisionnel n'a pas été suivi.



67	✓	✎	Vérification de la modification du statut dans les profils (tests)	1 jr	Lun 15/04/24	Lun 15/04/24
68	✓	✎	Gestion de l'accessibilité de la création des clubs	3 jrs	Mer 10/04/24	Ven 12/04/24
69	✓	✎	Vérification de la gestion de l'accessibilité de la création des clubs (tests)	1 jr	Lun 15/04/24	Lun 15/04/24
70	✓	✎	Gestion de l'accessibilité de la création des posts	3 jrs	Mer 10/04/24	Ven 12/04/24
71	✓	✎	Vérification de la gestion de l'accessibilité de la création des posts (tests)	1 jr	Lun 15/04/24	Lun 15/04/24
72	✓	✎	Mise en commun	2 jrs	Mar 16/04/24	Mer 17/04/24
73	✓	✎	Réunion d'équipe	0 jr	Mar 16/04/24	Mar 16/04/24
74	✓	✎	Réunion 5	0 jr	Mer 17/04/24	Mer 17/04/24
75	✓	✎	Validation de la version 1	1 jr	Jeu 18/04/24	Jeu 18/04/24
76	✓	✎	Fin version 1.1	0 jr	Jeu 18/04/24	Jeu 18/04/24
77	✓	✎	Ecriture du rapport de la version 1.1	2 jrs	Jeu 18/04/24	Ven 19/04/24
78	✓	✎	Version 2 : filtre	9 jrs	Lun 22/04/24	Jeu 02/05/24
79	✓	✎	Maquette version 2	1 jr	Lun 22/04/24	Lun 22/04/24
80	✓	✎	FF1 Utilisateur : filtres sur les clubs du fil d'actualité	5 jrs	Mar 23/04/24	Lun 29/04/24
81	✓	✎	Création d'un filtre par club sur le fil d'actualité	4 jrs	Mar 23/04/24	Ven 26/04/24
82	✓	✎	Valider les filtres sur un compte (tests)	1 jr	Lun 29/04/24	Lun 29/04/24
83	✓	✎	Gestion de l'ajout d'un filtre dans la base de données	4 jrs	Mar 23/04/24	Ven 26/04/24
84	✓	✎	Validation de l'ajout d'un filtre dans la base de données (tests)	1 jr	Lun 29/04/24	Lun 29/04/24
85	✓	✎	Réunion d'équipe	0 jr	Mer 24/04/24	Mer 24/04/24
86	✓	✎	Mise en commun	1 jr	Mar 30/04/24	Mar 30/04/24
87	✓	✎	Validation de la version 2	1 jr	Mer 01/05/24	Mer 01/05/24
88	✓	✎	Fin version 2	0 jr	Mer 01/05/24	Mer 01/05/24
89	✓	✎	Ecriture du rapport de la version 2	2 jrs	Mer 01/05/24	Jeu 02/05/24
90	✓	✎	Version 3 : groupe de messages	10 jrs	Ven 03/05/24	Jeu 16/05/24
91	✓	✎	Maquette version 3	1 jr	Ven 03/05/24	Ven 03/05/24
92	✓	✎	Réunion d'équipe	0 jr	Ven 03/05/24	Ven 03/05/24
93	✓	✎	FF2 Utilisateur : création de groupes de discussion	4 jrs	Lun 06/05/24	Jeu 09/05/24
94	✓	✎	Création de groupe de messages	3 jrs	Lun 06/05/24	Mer 08/05/24
95	✓	✎	Validation de la possibilité de créer un groupe de messages (tests)	1 jr	Jeu 09/05/24	Jeu 09/05/24
96	✓	✎	Création d'un bouton "Rejoindre un groupe"	3 jrs	Lun 06/05/24	Mer 08/05/24
97	✓	✎	Validation de la possibilité de rejoindre un groupe (tests)	1 jr	Jeu 09/05/24	Jeu 09/05/24
98	✓	✎	Gestion des groupes de message	3 jrs	Lun 06/05/24	Mer 08/05/24
99	✓	✎	Validation de la gestion des groupes de messages (tests)	1 jr	Jeu 09/05/24	Jeu 09/05/24
100	✓	✎	Gestion de la base de données des demandes et des membres actuels	3 jrs	Lun 06/05/24	Mer 08/05/24
101	✓	✎	Vérification de la gestion des demandes et des membres actuels dans la base de données (tests)	1 jr	Jeu 09/05/24	Jeu 09/05/24
102	✓	✎	Réunion d'équipe	0 jr	Mer 08/05/24	Mer 08/05/24
103	✓	✎	FF1 Président : ajout d'un membre dans un groupe de discussion	2 jrs	Ven 10/05/24	Lun 13/05/24
104	✓	✎	Création d'un bouton « ajout du membre »	1 jr	Ven 10/05/24	Ven 10/05/24
105	✓	✎	Vérification du bouton « ajout du membre » (tests)	1 jr	Lun 13/05/24	Lun 13/05/24
106	✓	✎	Gestion de l'ajout d'un membre à la base de données	1 jr	Ven 10/05/24	Ven 10/05/24
107	✓	✎	Vérification de la base de données à la suite des ajouts de nouveaux membres (tests)	1 jr	Lun 13/05/24	Lun 13/05/24
108	✓	✎	FF3 Utilisateur : notification sur le fil d'actualité	2 jrs	Ven 10/05/24	Lun 13/05/24
109	✓	✎	Gestion des notifications des posts	1 jr	Ven 10/05/24	Ven 10/05/24



110	✓	Validation de la gestion des notifications des posts (tests)	1 jr	Lun 13/05/24	Lun 13/05/24
111	✓	Réunion d'équipe	0 jr	Lun 13/05/24	Lun 13/05/24
112	✓	Mise en commun	1 jr	Mar 14/05/24	Mar 14/05/24
113	✓	QCM Moodle	0 jr	Mar 14/05/24	Mar 14/05/24
114	✓	Validation de la version 3	1 jr	Mer 15/05/24	Mer 15/05/24
115	✓	Fin version 3	0 jr	Mer 15/05/24	Mer 15/05/24
116	✓	Atelier de communication	0 jr	Mer 15/05/24	Mer 15/05/24
117	✓	Ecriture du rapport de la version 3	2 jrs	Mer 15/05/24	Jeu 16/05/24
118	✓	Fin du projet	14 jrs	Ven 17/05/24	Mer 05/06/24
119	✓	Création d'un lien de téléchargement	1 jr	Ven 17/05/24	Ven 17/05/24
120	✓	Réunion d'équipe	0 jr	Lun 20/05/24	Lun 20/05/24
121	✓	Validation du lien de téléchargement (tests)	1 jr	Lun 20/05/24	Lun 20/05/24
122	✓	Validation de l'application	1 jr	Mar 21/05/24	Mar 21/05/24
123	✓	Réunion 6	0 jr	Mer 22/05/24	Mer 22/05/24
124	✓	Ecrire le rapport	4 jrs	Mer 22/05/24	Lun 27/05/24
125	✓	Dépôt_Rapport	0 jr	Lun 27/05/24	Lun 27/05/24
126	✓	Réunion d'équipe	0 jr	Lun 27/05/24	Lun 27/05/24
127	✓	Préparation de la présentation	7 jrs	Mar 28/05/24	Mer 05/06/24
128	✓	Réunion d'équipe	0 jr	Lun 03/06/24	Lun 03/06/24
129	✓	Réunion 7 présentation	0 jr	Mer 05/06/24	Mer 05/06/24
130	✓	Fin du projet	0 jr	Mer 05/06/24	Mer 05/06/24



Glossaire

Android Système d'exploitation mobile développé par Google, utilisé principalement sur les smartphones et les tablettes. Android est basé sur le noyau Linux et est principalement écrit en langage de programmation Java.. 4

API (Application Programming Interface) Ensemble de définitions et de protocoles permettant à différentes applications logicielles de communiquer entre elles. Les API définissent les méthodes et les données que les développeurs peuvent utiliser pour interagir avec des services logiciels.. 8

APK (Android Application Package) APK signifie Android Application Package (paquet d'applications Android). Il s'agit d'un type de fichier qui regroupe tous les composants d'une application, ce qui permet de l'installer sur un système d'exploitation compatible. Les fichiers APK fonctionnent sous Android et sont donc compatibles avec les téléphones portables, les lecteurs multimédias, les téléviseurs intelligents ou les tablettes.. 4

Authentification Processus permettant de vérifier l'identité d'un utilisateur. Dans le contexte de l'application AtClub, cela se fait à l'aide de l'identifiant et du mot de passe de la Resel.. 4

Backend Partie du développement web qui se concentre sur la logique côté serveur, la gestion de la base de données, et l'architecture de l'application.. 3

Base de données Ensemble organisé de données structurées et interconnectées, stockées électroniquement dans un système informatique, permettant un accès et une manipulation efficaces des informations.. 3

BDX (Bureau des X) Association étudiante gérant les clubs et activités sur le campus de l'IMT Atlantique. Chaque BDX peut créer et gérer des clubs via l'application.. 4

Bug Défaut ou une anomalie dans le code d'un logiciel qui provoque un comportement inattendu ou incorrect de l'application.. 12

Bêta-test Phase de test au cours de laquelle un échantillon du public visé essaie le produit.. 12

Canaux de diffusion Les canaux de diffusion désignent les moyens par lesquels des informations sont transmises à un public cible. Ces canaux peuvent inclure des plateformes en ligne telles que les réseaux sociaux, les sites web, les applications mobiles, ainsi que des médias traditionnels comme la télévision, la radio ou la presse écrite. L'objectif est de choisir les canaux les plus adaptés pour atteindre efficacement et pertinemment l'audience visée.. 1

- Checkbox** Élément d'interface utilisateur graphique permettant de faire des sélections multiples parmi un ensemble d'options.. 9
- Classe** Structure fondamentale de la base de données qui évolue progressivement à mesure de l'ajout de nouvelles fonctionnalités ou classes, reflétant ainsi l'extension et la complexité croissante du système.. 3
- Code source** Ensemble des instructions et des déclarations écrites par les développeurs en utilisant un langage de programmation spécifique.. 3
- Compiler** Processus de transformation du code source en langage machine exécutable.. 17
- Dart** Langage de programmation développé par Google, utilisé notamment pour le développement d'applications mobiles avec Flutter.. 17
- Dictionnaire** Structure de données associant des clés à des valeurs, souvent utilisée pour stocker des informations de manière organisée et facilement accessible dans une application Flutter.. 7
- Fil d'actualité** Section d'une application ou d'un site web affichant les contenus récents et pertinents, tels que les publications, les mises à jour ou les activités des utilisateurs.. 4
- Firebase** Plateforme de développement d'applications mobiles et web, utilisée pour la base de données et les services backend de l'application AtClub.. 3
- Flutter** Cadre de développement d'applications mobiles développé par Google, utilisé pour créer des applications nativement compilées pour Android et iOS à partir d'un seul code source.. 3
- Formulaire** Interface utilisateur permettant de saisir des données structurées.. 5
- Frontend** Partie du développement web qui se concentre sur l'interface utilisateur et l'interaction utilisateur.. 3
- Hachage** Processus qui transforme une entrée en une sortie de longueur fixe appelée "haché".. 8
- Identifiant (ID)** Chaîne de caractères unique permettant de reconnaître un utilisateur ou un objet spécifique dans un système informatique.. 5
- IMT Atlantique** Grande école d'ingénieurs française, ayant plusieurs campus dont celui de Brest.. 4
- iOS** Système d'exploitation mobile développé par Apple Inc. pour ses appareils mobiles, y compris l'iPhone, l'iPad et l'iPod Touch. iOS est dérivé de macOS et est basé sur le noyau Darwin.. 4

- Jeton de rafraîchissement (Refresh Token)** Type de jeton utilisé pour obtenir un nouveau jeton d'accès sans nécessiter une ré-authentification complète de l'utilisateur.. 8
- Jeton d'accès (Access Token)** Chaîne de caractères générée par un serveur d'authentification et utilisée par les clients pour accéder à des ressources protégées.. 8
- Notifications push** Messages envoyés directement à un appareil par une application ou un site web, apparaissant instantanément sur l'écran même si l'application n'est pas ouverte.. 4
- Post** Message publié sur le fil d'actualité de l'application, contenant des informations sur les activités des clubs.. 2
- Profil** Ensemble des informations personnelles d'un utilisateur dans l'application.. 5
- QR Code** Type de code-barres en deux dimensions, utilisé pour stocker des informations.. 18
- Resel** Service de réseau de l'IMT Atlantique, fournissant des identifiants de connexion pour les étudiants.. 5
- Réseau social** Plateforme en ligne permettant de créer des profils, de partager des informations et de communiquer avec d'autres utilisateurs.. 1
- Salage** Technique utilisée en cryptographie pour renforcer la sécurité des mots de passe stockés.. 8
- Statut** Rôle ou position d'un utilisateur dans l'application, déterminant les fonctionnalités et permissions disponibles.. 4
- Stream Builder** Widget utilisé dans le framework Flutter pour synchroniser les vues avec les données en streaming, permettant des mises à jour dynamiques de l'interface utilisateur en réponse aux changements des données.. 7
- Système d'Exploitation (Operating System)** Logiciel de base qui gère les ressources matérielles et logicielles d'un appareil informatique.. 17
- Tabbar** Élément d'interface utilisateur qui permet aux utilisateurs de naviguer entre différentes sections ou vues d'une application en cliquant sur des onglets.. 8
- Test de validation** Processus de vérification pour s'assurer que les fonctionnalités de l'application fonctionnent correctement et respectent les spécifications du cahier des charges.. 2

Téléchargement (DDL) Action de transférer un fichier depuis un serveur vers un appareil utilisateur.. 6

UserProvider Classe ou un composant utilisé pour gérer les informations et les états liés à l'utilisateur dans une application, souvent utilisé dans le contexte de la gestion de l'authentification et de la session utilisateur.. 9

Utilisateur Personne utilisant l'application AtClub, qui peut avoir différents rôles comme élève, président de club, ou membre d'un BDX.. 4

Widget Élément d'interface utilisateur permettant aux utilisateurs d'interagir avec une application.. 3

Wrapper Composant utilisé pour envelopper un autre élément ou une autre fonctionnalité dans le but de fournir une interface simplifiée ou des fonctionnalités supplémentaires.. 9