

BTS SIO 2022

Option SLAM

Épreuve E5

EPSI Montpellier

BULKAN Mélanie

N° candidat : 02046081832

Situation n° 1/2 :

Application mobile avec gestion de livres

Flutter

Table des matières

Le contexte	3
Le client	5
Le réseau de l'entreprise	6
La réalisation.....	7
Les fonctionnalités	8
La démarche	10
La base de données + UML.....	11
Quelques fonctionnalités expliquées	12
Inscription et connexion sécurisée.....	12
Explication de la fonctionnalité :	12
Captures d'écran	12
Extrait de code.....	12
Affichage des livres	13
Explication de la fonctionnalité :	13
Captures d'écran	14
Extrait de code.....	14
Réservation d'un livre.....	15
Explication de la fonctionnalité :	15
Captures d'écran	15
Extrait de code.....	16
Page de profil	17
Explication de la fonctionnalité :	17
Captures d'écran	17
Extrait de code.....	18
Interface administration.....	20
Explication de la fonctionnalité :	20
Captures d'écran	20
Extrait de code.....	21
Paiement des dons	22
Explication de la fonctionnalité :	22
Captures d'écran	22
Extrait de code.....	24
Serveur de versions	25
Axes d'amélioration	25
Conclusion	26

Le contexte

Le contexte s'appuie sur la Maison des Associations d'Occitanie (MAO). Cette structure est chargée de mettre à disposition des locaux et des services destinés aux associations sportives régionales ainsi qu'à d'autres structures qu'elle héberge. Etant une institution régionale, la MAO bénéficie d'un financement octroyé par le Conseil Régional d'Occitanie et son administration est déléguée au CROSLR, autrement connu sous le nom de Comité Régional Olympique et Sportif de Languedoc-Roussillon.

Le directeur de cette association, Monsieur Sapin, a accueilli des invités pour une interview dans les nouveaux locaux de l'extension de la Maison des Associations d'Occitanie dans lesquels il évoque différents points. Tout d'abord, la Maison des Associations n'a pas de statut juridique en propre, mais une convention de cogestion a été signée avec le Comité Régional Olympique et Sportif de Languedoc-Roussillon. Les comités départementaux sont hébergés dans les locaux de la Maison des Associations, comme cela a été négocié lors de la reprise de la Maison des Ligues en 2000 par la Région Languedoc-Roussillon.

Ainsi, la MAO héberge la majorité des associations sportives régionales ainsi que quelques comités départementaux, le CROSLR et sa déclinaison départementale, le CDOS. La plus grande association hébergée est celle de tennis qui emploie 8 personnes. En ce qui concerne la présence des comités départementaux, elle est régulièrement remise en question mais sûrement rien ne sera fait avant la réforme des territoires et la transformation, par une fusion, des conseillers régionaux et généraux en conseillers territoriaux. Il est tout aussi important de noter que la MAO s'est diversifiée pour inclure tout type d'associations, dans des domaines autres que le sport, et donc, aider divers acteurs locaux.

La convention d'occupation du domaine public de 2003, impliquant la MAO, est en cours de révision pour inclure les agrandissements et les nouveaux services proposés. Cette convention inclut une obligation d'assurances en responsabilité civile pour les structures hébergées. Le montant est calculé sur une base forfaitaire de 5 € par mètre carré et par mois, avec une augmentation prévue d'un euro dans les mois à venir. Les charges comprennent le chauffage, l'électricité, le nettoyage et l'accès Internet. Les consommations sont refacturées chaque trimestre aux structures hébergées. Il n'y a pas de comptabilité analytique pour répartir les charges de manière précise, mais le téléphone ne fait pas partie des charges.

Du point de vue de la comptabilité, la MAO ne facture pas l'utilisation de ressources appartenant à la Région, mais elle facture des prestations annexes réalisées sur des équipements lui appartenant, comme l'affranchissement et les tirages sur les photocopieurs numériques. Concernant la réservation des salles de réunions, l'amphithéâtre ou encore la nouvelle salle de convivialité, il y a une demi-journée gratuite par an pour chaque ligue et comité départemental. Les salles de réunions sont réservables via l'Intranet, avec différents services associés dont certains sont payants. Il y a quatre niveaux de tarification, allant de gratuit pour six réservations par an pour les associations à un tarif plus élevé pour les organismes non hébergés. Les informations sont transmises au Conseil Régional qui émet un titre de paiement, qui est une facture d'occupation de locaux.

Parmi le personnel employé par la MAO, nous retrouvons le personnel prestataire (gardiennage, ménage) et les salariés des associations. Les bénévoles forment une autre branche de ce personnel. Les élus du Conseil Régional, les visiteurs et les stagiaires forment la dernière composante de cette structure. De plus, la MAO a droit au Plan Impact Emploi lui donnant la possibilité d'utiliser des logiciels permettant l'établissement des bulletins de salaire et des déclarations sociales et fiscales. Ce logiciel gratuit nécessite néanmoins un financement puisqu'elle requiert des ressources humaines. Ainsi, une cotisation annuelle 60€ ainsi que 5€ par fiche de paie est demandée à chaque association hébergée par la MAO. Ce tarif comprend les récapitulatifs Périodiques, les déclarations URSSAF et ASSEDICS ainsi que les organismes de retraite.

Le site de la Maison des Associations d'Occitanie est constitué de quatre bâtiments, deux avec plusieurs étages et deux de plain-pied. Au total, 80 bureaux sont situés dans les étages de deux bâtiments (A et C), tandis que les espaces mutualisés sont situés au rez-de-chaussée de chaque bâtiment. Les 24 associations hébergées occupent un ou plusieurs bureaux en fonction de leur taille, avec la possibilité de communiquer entre eux si les bureaux sont contigus. Elles ont également accès à des salles de réunion sur réservation, tandis qu'une salle de réunion libre est disponible en bout d'étage pour les employés et bénévoles des associations. La location est basée sur un forfait de charges locatives par bureau (5€ par m2 et par mois).

Divers équipements et matériel informatique sont disponibles dans les espaces dédiées aux associations. On y retrouve différentes connectiques tels que des prises Ethernet, des prises téléphoniques, etc...; des écrans allant jusqu'à 42" avec des systèmes de diffusion d'information, des imprimantes, photocopieuses, des baies de brassages comprenant des prises Ethernet, des routeurs et des commutateurs, ainsi que des bornes wifi car l'accès Wifi est gratuit pour l'ensemble des bâtiments que ce soit pour les visiteurs, les clients ou le personnel. Tout en sachant que le projet d'expansion constante de cette structure les pousse à agrandir leurs catalogues de services dont le matériel informatique mis à disposition au sein des locaux.

Concernant les réseaux informatiques du site, la MAO dispose actuellement d'un réseau commun aux associations et à l'administration et d'un réseau de type DMZ dans lequel on trouve un ensemble de serveurs accessibles depuis l'extérieur. De plus, l'administration de la MAO dispose de plusieurs serveurs dont un qui fait office de serveur DHCP, un serveur de fichier, un serveur web apache et IIS ainsi qu'un serveur de gestion des configurations et tickets (GLPI). Tout en sachant qu'une segmentation des réseaux est en projet pour augmenter la sécurité du site.

Le client

MyLibrary est une association caritative qui a pour mission de promouvoir l'accès à la lecture pour tous. Elle propose une large sélection de livres gratuits mis à la disposition de ses membres et organise des événements pour encourager la lecture et le partage de connaissances. L'association est basée à Paris, en France, et compte des milliers de membres à travers le pays.

Afin de faciliter l'accès à ses livres et de renforcer son engagement auprès de ses membres, MyLibrary souhaite développer une solution mobile en Flutter. Cette application permettra aux membres de consulter les livres disponibles, de les télécharger pour une lecture hors ligne et de faire des dons pour soutenir l'association. L'application sera disponible sur Android et sera accessible à tous les membres de MyLibrary.

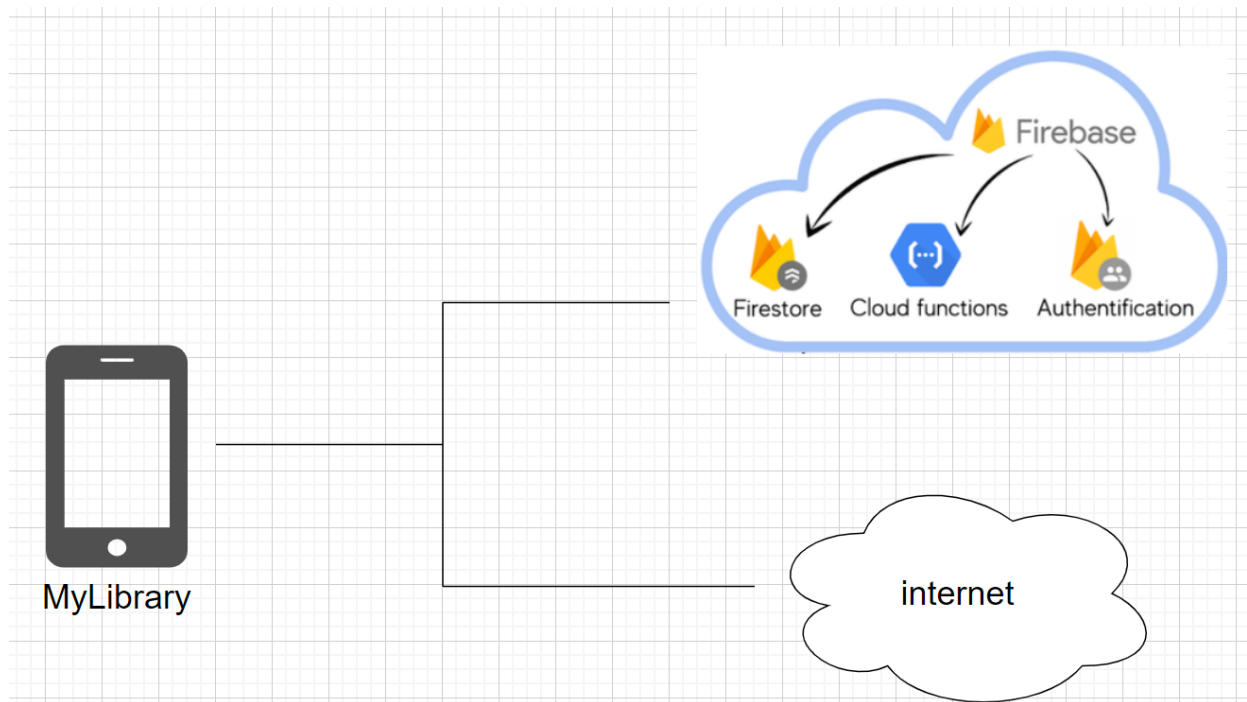
En tant que fournisseur de services informatiques, nous sommes en mesure de répondre aux besoins de MyLibrary en matière de développement d'application mobile. Nous avons une expérience avérée dans le développement de solutions sur mesure pour les entreprises et les organisations à but non lucratif, et nous sommes en mesure de fournir une application mobile qui répondra aux attentes de MyLibrary en termes de qualité, de fonctionnalité et de sécurité.

Notre application mobile sera développée en utilisant des technologies modernes et polyvalentes telles que Flutter, Firebase et Google Cloud, et sera capable de s'intégrer facilement aux systèmes existants de l'association. Elle sera également capable de gérer les informations de manière à assurer la confidentialité et la sécurité des données sensibles. L'application sera conçue pour offrir une expérience utilisateur optimale, avec une interface intuitive et facile à utiliser, ainsi qu'une navigation fluide et rapide.

Nous offrirons un soutien technique complet pour assurer une utilisation efficace de l'application mobile. Nous fournirons également une formation pour que les membres de MyLibrary puissent utiliser l'application en toute confiance et autonomie. Nous sommes déterminés à offrir un service de qualité supérieure à MyLibrary et nous nous engageons à fournir une application mobile de pointe qui répondra aux besoins de l'association tout en respectant les délais convenus.

En développant une application mobile pour MyLibrary, nous contribuerons à renforcer l'engagement de l'association auprès de ses membres et à promouvoir l'accès à la lecture pour tous. Nous sommes convaincus que notre solution sur mesure répondra aux attentes de MyLibrary en termes de qualité, de fonctionnalité et de sécurité, et nous sommes impatients de travailler avec l'association pour concrétiser ce projet.

Le réseau de l'entreprise



Tout simplement, nous allons nous retrouver avec une application (MyLibrary) utilisant les outils de Firebase et reliée à internet.

La réalisation

Notre réalisation consiste à concevoir et développer une application mobile en respectant les exigences fonctionnelles et techniques définies dans le cahier des charges. Cette application permettra aux utilisateurs de consulter les livres gratuitement mis à leur disposition, de rechercher des ouvrages, de les ajouter en favoris, de les réserver pour les emprunter en version physique, de les télécharger pour une lecture hors-ligne et de faire des dons à l'association pour soutenir ses actions. Nous devons également veiller à ce que l'application soit ergonomique, intuitive et facile à utiliser pour les membres de l'association.

De plus, nous devons penser aux administrateurs (personnel de MyLibrary) qui vont devoir utiliser l'application. Il convient donc de réaliser une interface d'administration pour permettre une gestion beaucoup plus facile des livres, des auteurs, des stocks, etc... Ainsi, pas besoin de passer par Firebase et son interface qui peut être fastidieuse et difficile à comprendre pour ajouter, supprimer ou mettre à jour des données.

Pour mener à bien ce projet, nous avons décidé de suivre une méthodologie de développement itérative et incrémentale. Cette approche nous permettra de développer l'application par étapes successives, en validant régulièrement les fonctionnalités développées avec l'association.

Dans un premier temps, nous avons réalisé une analyse détaillée des besoins de l'association et avons rédigé un cahier des charges fonctionnel et technique (voir le lien sur le drive). Ce document décrit précisément les fonctionnalités attendues de l'application, les contraintes techniques et les exigences en termes de performance et de sécurité.

Ensuite, nous avons élaboré un planning prévisionnel de réalisation, qui détaille les différentes étapes du projet, les livrables attendus et les jalons intermédiaires. Ce planning nous permettra de suivre l'avancement du projet et de garantir le respect des délais impartis.

Pour la conception de l'application, nous avons choisi d'utiliser la technologie Flutter, qui permet de développer des applications mobiles natives pour iOS et Android à partir d'un seul code source. Cette technologie offre de nombreux avantages en termes de performance, de qualité et de rapidité de développement.

Nous avons également choisi d'utiliser une architecture logicielle basée sur le modèle MVC (Modèle-Vue-Contrôleur), qui permet de séparer les différentes couches de l'application (présentation, logique métier et données) et de faciliter la maintenance et l'évolution de l'application.

Pour la gestion des données, nous avons choisi d'utiliser Firebase comme backend, pour stocker les données des livres et des membres, gérer l'authentification et envoyer des notifications push.

Enfin, pour garantir la qualité et la fiabilité de l'application, nous avons prévu de mettre en place des tests unitaires et des tests d'intégration tout au long du projet. Ces tests permettront de valider les fonctionnalités développées et de détecter d'éventuels dysfonctionnements avant la mise en production de l'application.

Les fonctionnalités

L'application mobile MyLibrary offrira les fonctionnalités suivantes :

-Inscription et connexion : Les utilisateurs pourront s'inscrire et se connecter à l'application en utilisant leur adresse e-mail et leur mot de passe. Ils pourront également se connecter via leur compte Google.

-Page profil : Les utilisateurs trouveront toutes les informations utiles (leurs dons, leurs réservations, leurs données personnelles...) sur leur page profil.

-Recherche de livres : Les utilisateurs pourront rechercher des livres via une barre de recherche. Les résultats de la recherche seront affichés sous forme de liste ou de grille.

-Détails du livre : Les utilisateurs pourront consulter les détails d'un livre, tels que la couverture, le résumé, l'auteur, l'éditeur, le nombre de pages, etc. Ils pourront aussi consulter la disponibilité du livre (quantité en stock).

-Détails de l'auteur : Les utilisateurs pourront consulter les détails d'un auteur tels que son nom, prénom, image, biographie, etc. Ils pourront aussi consulter des informations utiles tels que le nombre de livres ou de pages que l'auteur a rédigés.

-Téléchargement de livres : Les utilisateurs pourront télécharger des livres gratuitement et les lire hors ligne. Ils pourront également supprimer les livres téléchargés de leur appareil.

-Favoris: Les utilisateurs pourront ajouter ou supprimer des livres à leur liste de favoris et y accéder facilement.

Lecture de livres : Les utilisateurs pourront directement lire gratuitement des livres sur leur application. Ils disposeront d'une interface simple et accessible (avec des flèches pour changer de page par exemple).

Reservation de livres : Les utilisateurs pourront réserver un livre physique dans un stock choisi s'il est disponible. La durée de reservation est de un mois à compter de la date de réservation.

-Don : Les utilisateurs pourront faire des dons à l'association caritative MyLibrary via l'application en utilisant Stripe. Ils pourront choisir le montant du don et le mode de paiement (carte de crédit, Paylinks, etc.).

-Stock : L'application doit pouvoir afficher (pour les livres disponibles physiquement) le lieu et la quantité en stock.

-Administration : Les administrateurs devront pouvoir ajouter, modifier et supprimer les livres, les auteurs et les stocks. Ils devront aussi pouvoir gérer les livres et leurs quantités pour chaque stock directement depuis l'application.

Synthèse du cahier des charges :

L'application mobile MyLibrary offrira aux utilisateurs la possibilité de rechercher, de télécharger, de lire et de réserver gratuitement des livres ainsi que de créer une liste de favoris personnalisée et de faire des dons à l'association caritative MyLibrary. L'application sera conçue pour être facile à utiliser, sécurisée et évolutive. Elle disposera d'une interface administration regroupant l'ensemble des fonctions nécessaires à la gestion des données (ajout, modification et suppression d'auteurs, stocks et livres). Les exigences fonctionnelles et techniques seront définies en détail dans le cahier des charges, ainsi que les plans de test, de déploiement et de maintenance.

La demarche

Tout d'abord, nous avons réalisé une analyse approfondie des besoins de l'association en termes de fonctionnalités, d'ergonomie et de design. Cette analyse nous a permis de définir les spécifications fonctionnelles et techniques de l'application, ainsi que les contraintes liées à son développement.

Ensuite, nous avons procédé à la conception de l'architecture de l'application, en utilisant les meilleures pratiques en matière de développement mobile. Nous avons opté pour une architecture modulaire, basée sur le framework Flutter, qui permettra une grande flexibilité et une maintenance facilitée.

Une fois l'architecture définie, nous avons entamé la phase de développement proprement dite. En sachant qu'en parallèle du développement, nous avons également réalisé des tests unitaires et fonctionnels réguliers, afin de garantir la qualité et la fiabilité de l'application.

Enfin, nous avons prévu une phase de déploiement et de mise en production de l'application, qui sera précédée d'une phase de recette et de validation par l'association MyLibrary.

Nous avons également prévu une formation des utilisateurs finaux, afin de leur permettre de prendre en main l'application de manière optimale.

Le planning prévisionnel :

Ce planning est bien entendu susceptible d'être ajusté en fonction des éventuels imprévus ou changements de besoins. Nous avons prévu des points de synchronisation réguliers avec l'association MyLibrary, afin de suivre l'avancement du projet et de nous adapter au mieux à leurs attentes.

Analyse des besoins : quelques jours

Conception de l'architecture : 1 semaine

Développement : 5 semaines

Tests unitaires et fonctionnels : en continu tout au long du développement

Recette et validation : 2 semaines

Déploiement et mise en production : un ou deux jours

Formation des utilisateurs finaux : 1 semaine

La base de données + UML

Voici le schéma relationnel (MCD) que va suivre notre application :

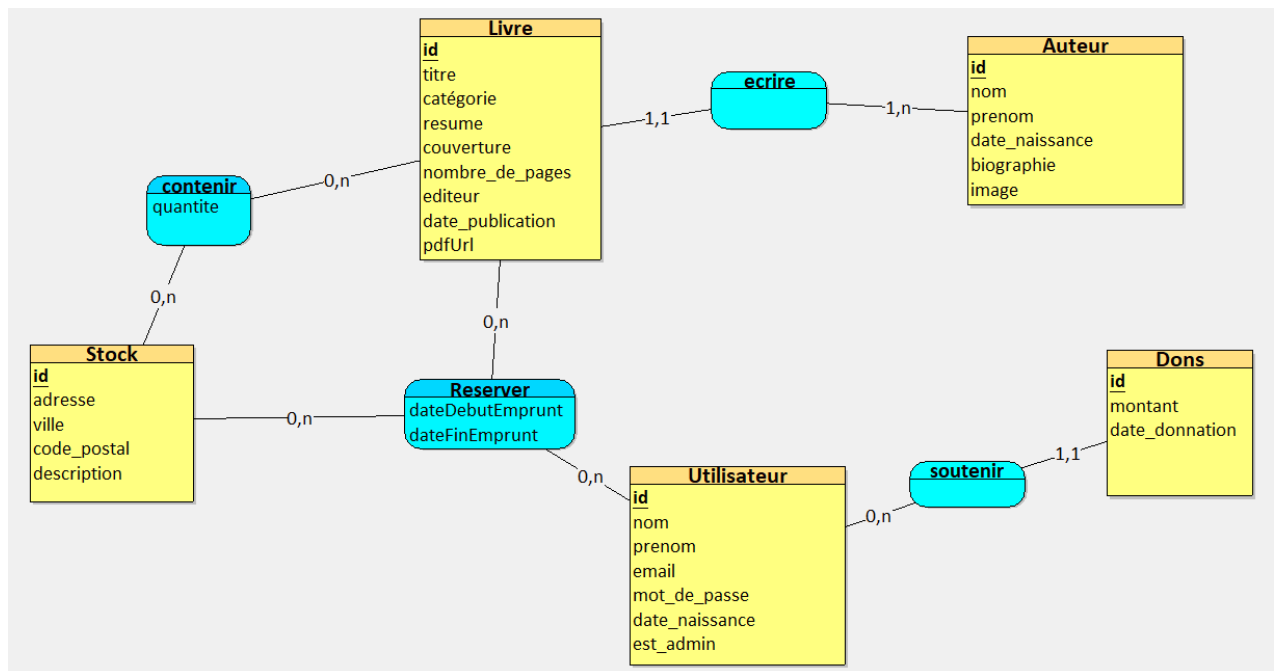
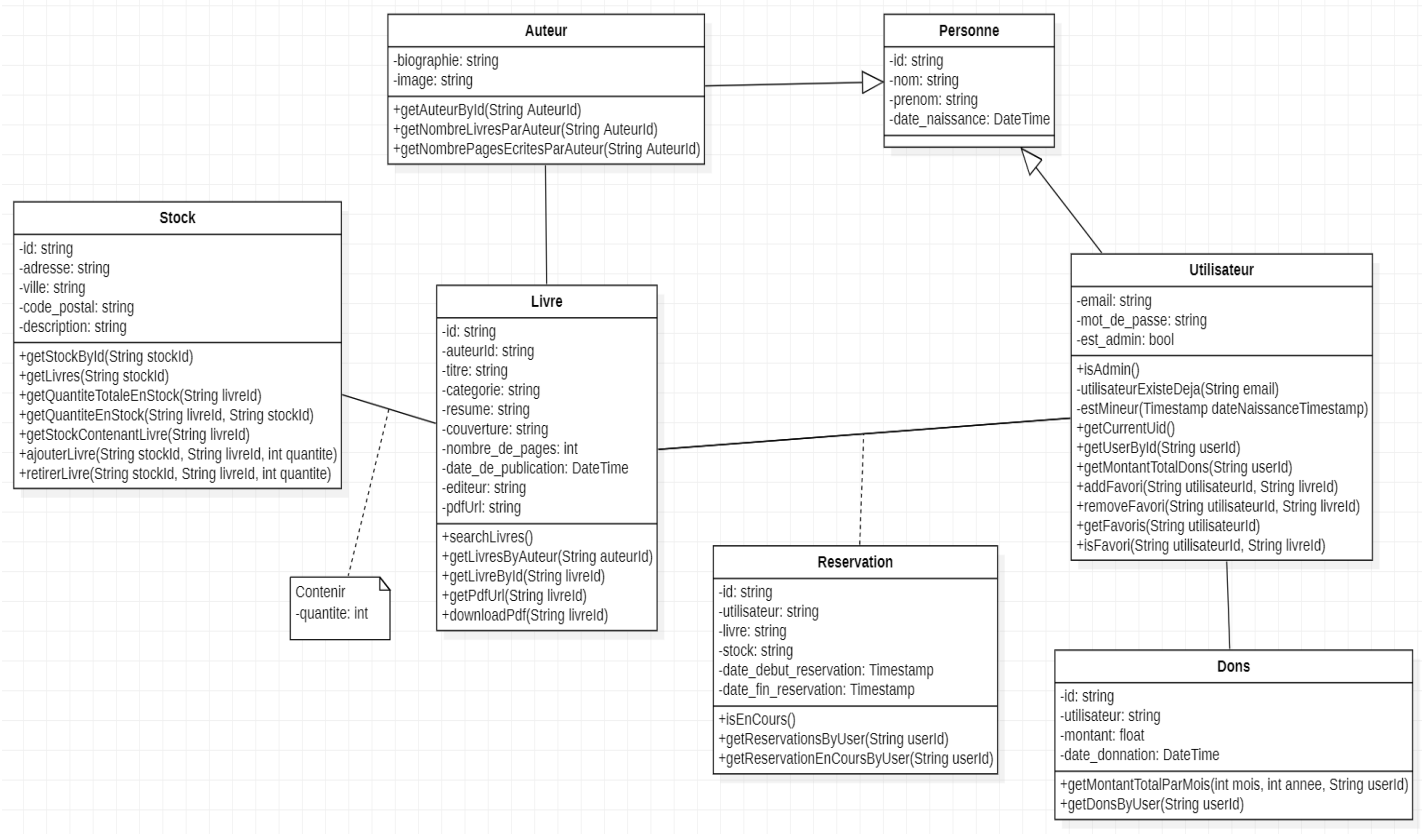


Diagramme UML : (diagramme des classes)



Quelques fonctionnalités expliquées

Inscription et connexion sécurisée

Libellé de la fonctionnalité : Inscription et connexion sécurisée

Explication de la fonctionnalité : Les utilisateurs peuvent créer un compte en fournissant leur adresse e-mail et leur mot de passe. Ils peuvent également se connecter à l'application en utilisant leur compte Google. Une fois connectés, les utilisateurs peuvent accéder à toutes les fonctionnalités de l'application.

Les informations d'authentification sont stockés sur Firebase. Les informations relatives à l'utilisateur (son nom, ses favoris, sa date de naissance...) se trouvent dans la collection « utilisateurs » de Firebase et sont automatiquement enregistrés lors de la première connexion de l'utilisateur.

Des tests sont réalisés à l'inscription pour voir si le mot de passe est robuste (au moins 8 caractères), si un autre utilisateur a une adresse mail identique, si l'utilisateur est mineur ou non, etc...

Captures d'écran :

Identifiant	Fournisseurs	Date de création ↓	Dernière connexion	UID utilisateur
claudetaure@gmail.com		7 avr. 2024	7 avr. 2024	6bxxVYC5IJUuomOWLESjU4U...
getlono.contact@gmail...		7 avr. 2024	7 avr. 2024	0WYv09dQ3HfJ2u6gfyCpPN...

(Interface Firebase Authentification)

🏠 > utilisateurs > KUmRP6kJXnV6...			☁ Plus de fonctionnalités dans Google Cloud ▼
🏠 (default)	📁 utilisateurs	📄 KUmRP6kJXnV6vMH5gF7BnhaReKF2	
+ Commencer une collection	+ Ajouter un document	+ Commencer une collection	
auteurs	0WYv09dQ3HfJ2u6gfyCpPNMfr53	favoris	
dons	20QNuNoMC5NYtBN7FmIgMjQJC02		
livres	6bxxVYC5IJUuomOWLESjU4UEih92		
reservations	⋮ KUmRP6kJXnV6vMH5gF7BnhaReKF2 >	+ Ajouter un champ	
stocks		date_naissance: 6 avril 1976 à 02:00:00 UTC+2	
utilisateurs >		email: "pierre.dupont@gmail.com"	
		est_admin: true	
		nom: "DUPONT"	
		prenom: "Pierre"	

(Informations des utilisateurs dans Firebase Database)

Extrait de code :

```

Future<UserCredential?> _signInWithGoogle() async {
  final GoogleSignInAccount? googleUser = await GoogleSignIn().signIn();

  if (googleUser == null) {
    return null;
  }

  final GoogleSignInAuthentication googleAuth = await googleUser.authentication;

  final credential = GoogleAuthProvider.credential(
    accessToken: googleAuth.accessToken,
    idToken: googleAuth.idToken,
  );

  final userCredential = await FirebaseAuth.instance.signInWithCredential(credential);

  if (userCredential.user != null) {
    if (userCredential.additionalUserInfo?.isNewUser == true) {
      Routes.router.navigateTo(context, '/welcome');
    } else {
      Routes.router.navigateTo(context, '/profil');
    }
  }
}

```

(Fonction de connexion avec google, si c'est la première fois que l'utilisateur se connecte, il est redirigé vers une landing page puis un formulaire qui permet de collecter ses données tels que sa date de naissance)

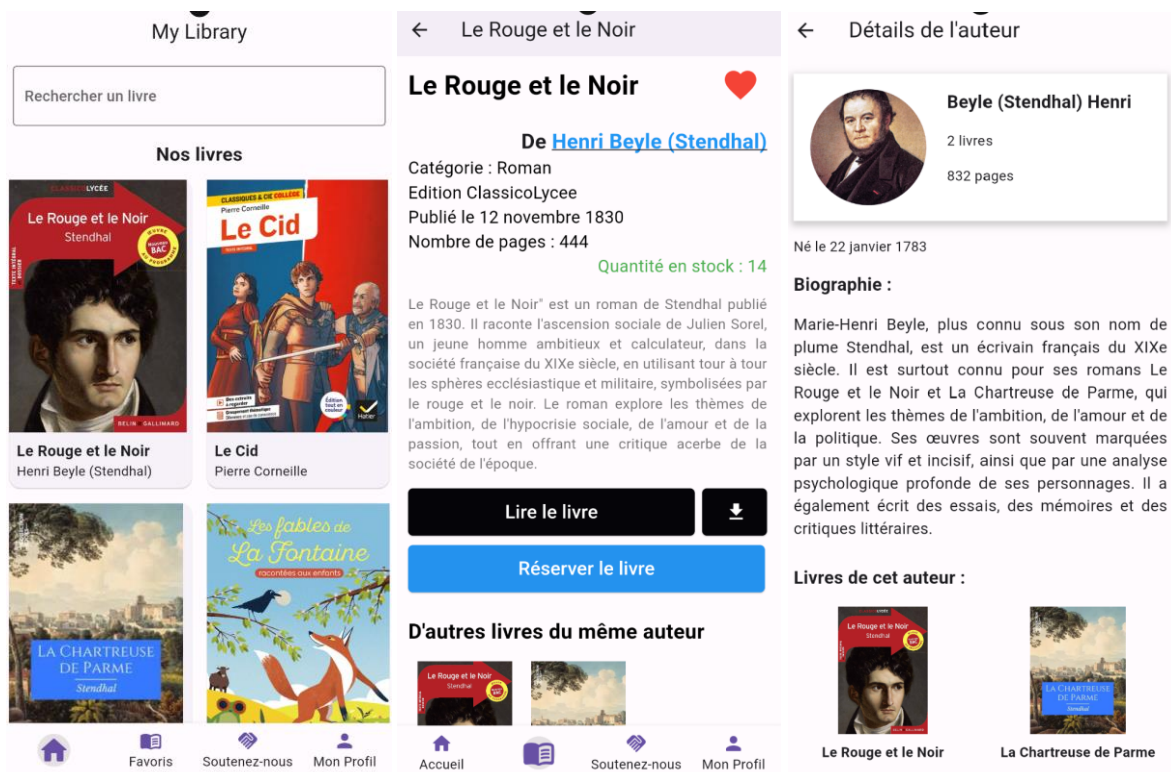
Affichage des livres

Libellé de la fonctionnalité : Affichage des livres et de leurs détails

Explication de la fonctionnalité : Cette fonctionnalité permet aux utilisateurs de parcourir la liste des livres disponibles dans l'application et d'afficher les détails de chaque livre. Les utilisateurs peuvent accéder à cette fonctionnalité à partir de l'écran d'accueil de l'application ou en recherchant un livre spécifique à l'aide de la barre de recherche.

Lorsque les utilisateurs accèdent à la liste des livres, ils peuvent voir les couvertures des livres, leurs titres et leur auteur. En cliquant sur un livre, les utilisateurs peuvent accéder à la page de détails du livre, qui affiche des informations plus détaillées sur le livre, telles que le résumé, l'éditeur, le nombre de pages, la date de publication, etc. Les utilisateurs peuvent également voir la disponibilité du livre (quantité en stock) . Ils peuvent via cette page accéder aux détails de l'auteur, télécharger le livre, le lire ou le réserver ou le rajouter en favoris. Ils ont aussi la liste des autres livres publiés par le même auteur.

Captures d'écran :



(pages d'affichage des livres et de leurs détails ainsi que le détail d'un auteur)

La quantité en stock deviens « Rupture de stock ! » affiché en rouge si le livre n'est disponible dans aucun stock.

Si l'utilisateur clique sur les autres livres du même auteur, il est redirigé vers la page affichant leurs de détails.

Extrait de code :

(page d'affichage des livres avec la barre de recherche)

Le code est découpé en plusieurs parties pour plus de visibilité. A noté que la barre de navigation (bottomNavigationBar) est un élément (widget) personnalité qui est situé dans un autre fichier. Il est réutilisé dans la plus part des fichiers view de l'application.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      automaticallyImplyLeading: false,
      title: const Text('My Library'),
      centerTitle: true,
    ), // AppBar
    body: Column(
      children: [
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: TextField(
            onChanged: (query) {
              _searchLivres(query);
            },
            decoration: const InputDecoration(
              hintText: 'Rechercher un livre',
              border: OutlineInputBorder(),
            ), // InputDecoration
          ), // TextField
        ), // Padding
        Container(
          padding: const EdgeInsets.all(8.0),
          child: const Text(
            'Nos livres',
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
          ), // Text
        ), // Container
        Expanded(
          child: _searchResults.isEmpty
            ? _buildBookGrid(_livres)
            : _buildBookGrid(_searchResults),
        ), // Expanded
      ], // Column
    ), // Column
    bottomNavigationBar: const BottomNavigationBarWidget(0),
  ); // Scaffold
}
```

Réservation d'un livre

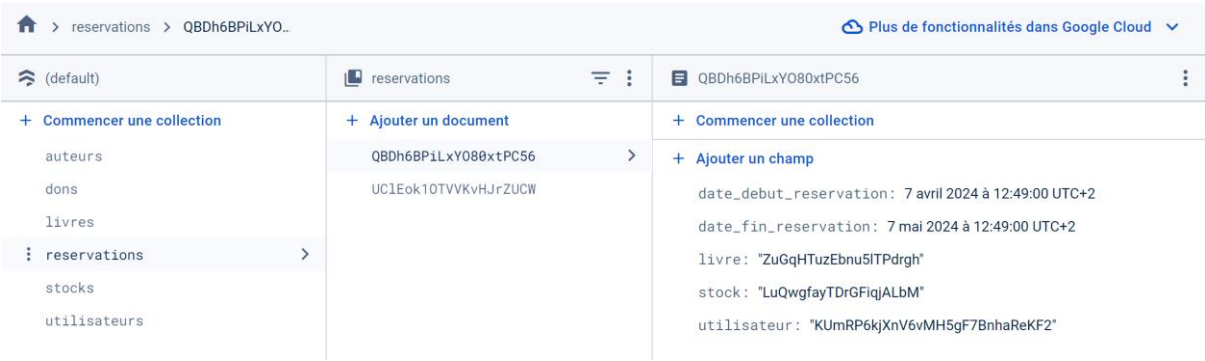
Libellé de la fonctionnalité : **Réservation d'un livre**

Explication de la fonctionnalité : La réservation de livres permet aux utilisateurs de réserver un livre physique dans un stock choisi s'il est disponible. La durée de réservation est d'un mois à compter de la date de réservation. Les utilisateurs peuvent consulter leurs réservations en cours et passées sur leur page de profil.

Pour réserver un livre, l'utilisateur doit sélectionner le livre souhaité et vérifier sa disponibilité dans le stock choisi. S'il est disponible, l'utilisateur peut cliquer sur le bouton "Réserver" et confirmer sa réservation via une pop up. Un utilisateur ne peut réserver qu'un livre à la fois.

Si le livre n'est pas disponible un stock choisi, l'utilisateur peut vérifier la disponibilité dans d'autres stocks.

Captures d'écran :

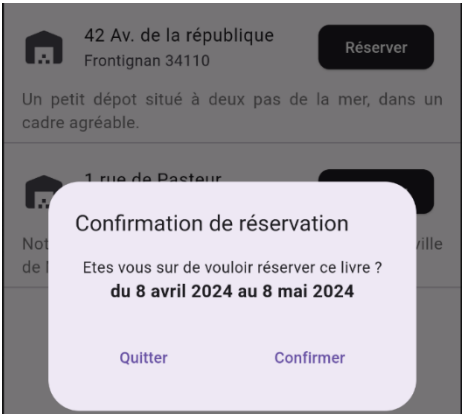


(Collection ‘reservations’ dans Firebase Database)

(Page de réservation pour un livre)

Seuls les stocks ayant au moins un seul livre (celui choisi par l'utilisateur) sont affichés.

Une pop up s’affiche pour confirmer la réservation.





(page d'affichage des réservations d'un utilisateur)

Les reservations en cours sont affich  en vert tandis que la liste des reservations pass es sont gris e.

Cet page est disponible dans la page profil de l'utilisateur, qu'il doit administrateur ou non.

Extrait de code :

```
Future<List<Reservation>> getReservationsByUser(String userId) async {  
    final QuerySnapshot snapshot = await _reservations  
        .where('utilisateur', isEqualTo: userId)  
        .orderBy('date_debut_reservation', descending: true)  
        .get();  
  
    return snapshot.docs.map((doc) {  
        final data = doc.data() as Map<String, dynamic>;  
        data['id'] = doc.id;  
        return Reservation.fromMap(data);  
    }).toList();  
}  
  
Future<List<Reservation>> getReservationEnCoursByUser(String userId) async {  
    final QuerySnapshot snapshot = await _reservations  
        .where('utilisateur', isEqualTo: userId)  
        .where('date_fin_reservation',  
            isGreaterThan: Timestamp.fromDate(DateTime.now()))  
        .get();  
  
    return snapshot.docs.map((doc) {  
        final data = doc.data() as Map<String, dynamic>;  
        data['id'] = doc.id;  
        return Reservation.fromMap(data);  
    }).toList();  
}
```

(Quelques fonctions pour r cup rer des r servations)

Page de profil

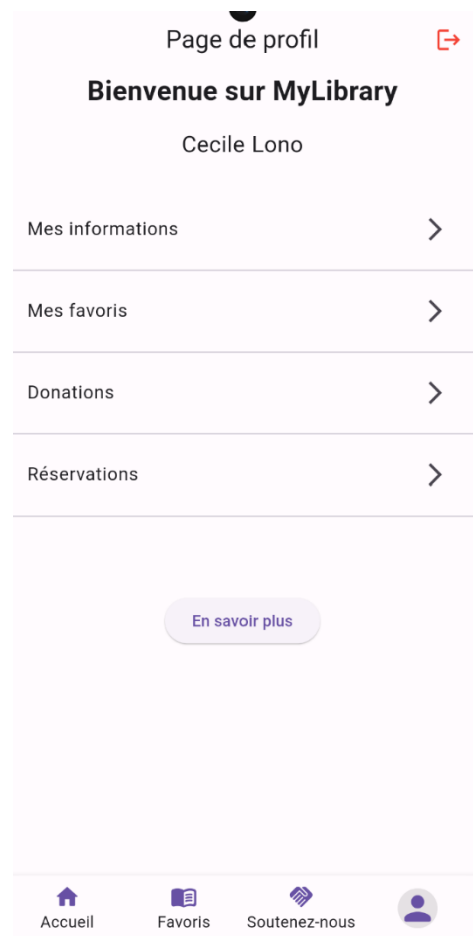
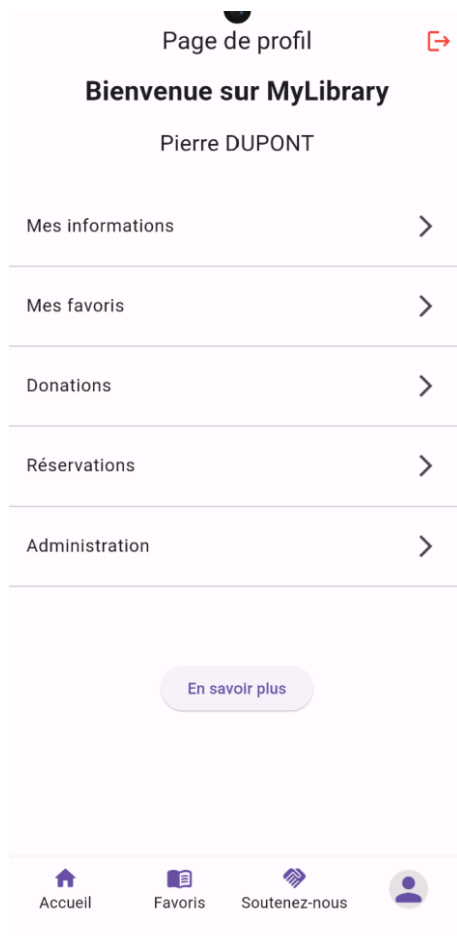
Libellé de la fonctionnalité : Page de profil d'un utilisateur

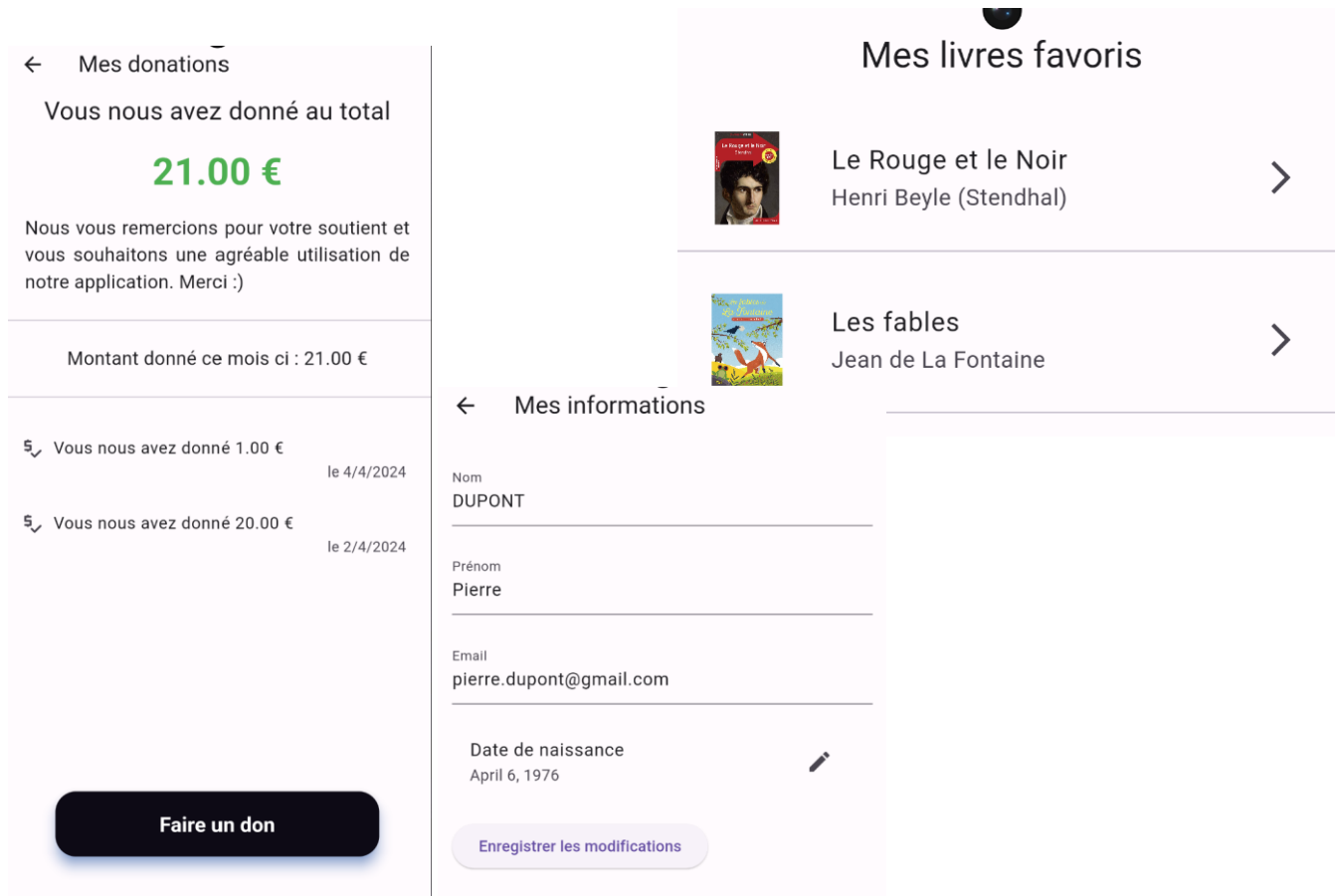
Explication de la fonctionnalité : La page de profil est l'endroit où les utilisateurs peuvent trouver toutes les informations utiles concernant leur compte. Elle reidirige vers 4 ou 5 page (selon si l'utilisateur est un administrateur) :

- Une qui affiche les données personnelles de l'utilisateur, telles que son nom, son prénom, son adresse e-mail, sa date de naissance, etc. Et lui permet de les modifier
- Une qui affiche une liste des livres favoris de l'utilisateur
- Une qui affiche leurs dons effectués, avec quelques données statistiques comme le montant donné au total, le montant donnée ce mois ci...
- Une qui permet aux utilisateurs de voir leurs réservations en cours et passés.
- Une page « administration » qui s'affiche uniquement si l'utilisateur est admin (nous verons son détail plus tard)

Captures d'écran :

(différences entre la page profil d'un utilisateur admin et non admin)





(quelques pages : Donnations, Mes informations et Mes reservations)

Extrait de code :

```
if (user.isAdmin())
...[
  ListTile(
    title: const Text('Administration'),
    trailing: const Icon(Icons.arrow_forward_ios),
    onTap: () {
      Routes.router.navigateTo(context, '/administration');
    },
  ), // ListTile
  const Divider(),
],
const SizedBox(height: 60),
ElevatedButton(
  onPressed: () {
    Routes.router.navigateTo(context, '/donation');
  },
  child: Text('En savoir plus'),
), // ElevatedButton
],
```

(page profil_view)

Ici, nous pouvons voir comment est géré l'affichage du lien vers la page d'administration en fonction de si l'utilisateur est un admin ou non.

En sachant que dans la classe Utilisateur (et sur firebase), il y a un attribut « est_admin » qui est un booléen (true or false). True si c'est un administrateur. De base, tous les utilisateurs ont cet attribut à « false » pour appliquer le principe de moindre privilège.

```

Future<double> getMontantTotalParMois(int mois, int annee, String? userId) async {
    QuerySnapshot snapshot;
    if (userId == null || userId.isEmpty) {
        snapshot = await _dons.get();
    } else {
        snapshot = await _dons
            .where('utilisateur', isEqualTo: userId)
            .get();
    }
    double montantTotal = 0;
    for (final DocumentSnapshot doc in snapshot.docs) {
        final data = doc.data() as Map<String, dynamic>;
        data['id'] = doc.id;
        final Don don = Don.fromMap(data);
        final DateTime dateDonation = don.dateDonation.toDate();
        if (dateDonation.month == mois && dateDonation.year == annee) {
            montantTotal += don.montant;
        }
    }
    return montantTotal;
}

Future<List<Don>> getDonsByUser(String userId) async {
    final QuerySnapshot snapshot = await _dons
        .where('utilisateur', isEqualTo: userId)
        .orderBy('date_donation', descending: true)
        .get();

    return snapshot.docs.map((doc) {
        final data = doc.data() as Map<String, dynamic>;
        data['id'] = doc.id;
        return Don.fromMap(data);
    }).toList();
}

```

(quelques fonctions utiles pour l'affichage des informations relatives aux dons)

```

child: _dons.isEmpty
? FractionallySizedBox(
  widthFactor: 0.8,
  child: Column(
    mainAxisAlignment: MainAxisAlignment.start,
    children: [
      const SizedBox(height: 32),
      const Text(
        'Vous n\'avez pas encore fait de donation, si vous souhaitez nous soutenir, cliquez sur le bouton ci dessous.',
        textAlign: TextAlign.justify,
        style: TextStyle(fontSize: 18),
      ), // Text
      const SizedBox(height: 32),
      ElevatedButton(
        onPressed: () {
          Routes.router.navigateTo(context, '/donation');
        },
        child: const Text('Faire un don'),
      ), // ElevatedButton
    ],
  ),
)

```

(si l'utilisateur n'a aucun don, sa page de donation affiche un texte « Vous n'avez pas encore fait de donation »)

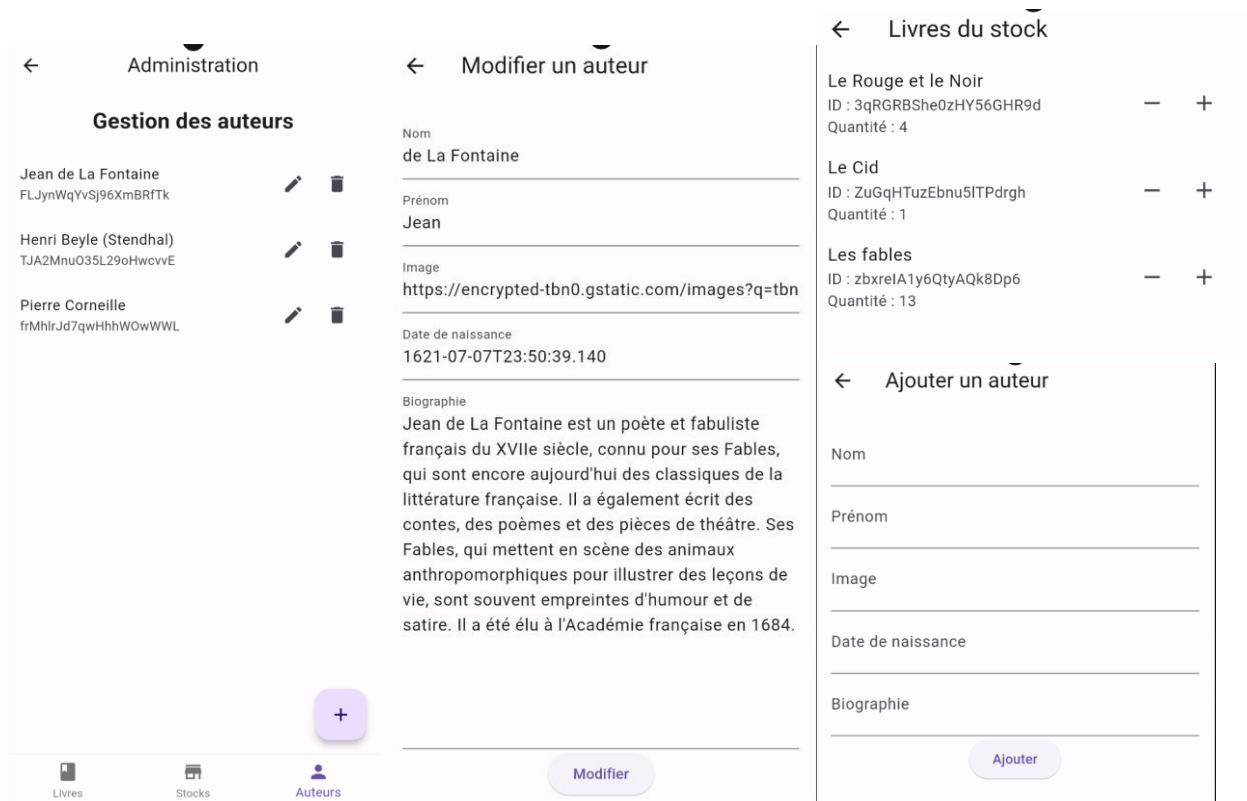
Interface administration

Libellé de la fonctionnalité : La page d'interface administration

Explication de la fonctionnalité : L'interface d'administration est une page réservée aux administrateurs de l'application MyLibrary. Cette page permet de gérer les livres, les auteurs et les stocks de l'application. Les administrateurs peuvent ajouter, modifier et supprimer des livres, des auteurs et des stocks, ainsi que gérer les quantités de livres disponibles dans chaque stock.

Lorsque les administrateurs accèdent à la page d'administration, ils peuvent voir trois onglets : "Livres", "Auteurs" et "Stocks". Pour chaque onglet, ils peuvent ajouter, modifier ou supprimer des éléments. Des vérifications sont faites à chaque fois pour vérifier la conformité des données (par exemple, la quantité de livres ne peut pas être négative).

Captures d'écran :

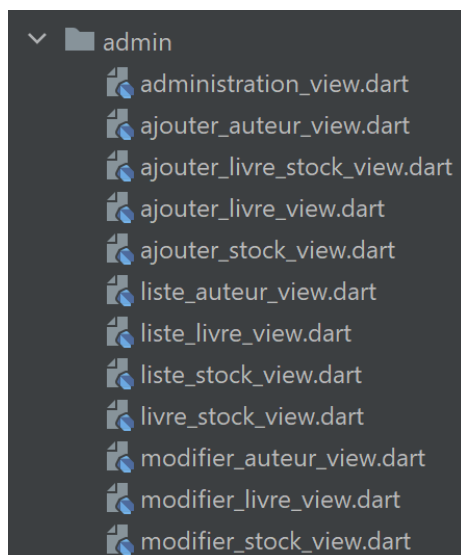


(Pages d’affichage, de modification et d’ajout d’auteurs)

(La page en haut à droite concerne les stocks, c’est la page de gestion des quantités de livre dans un stock)

Extrait de code :

(les pages liées à l’interface administration)



(à droite, la page administration_view qui va gérer toutes les pages liées à l’interface administration)

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Administration'),
      centerTitle: true,
    ), // AppBar
    body: _children[_currentIndex],
    floatingActionButton: (_currentIndex == 0)
      ? FloatingActionButton(
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => AjouterLivreView()),
          );
        },
        child: const Icon(Icons.add),
      ) // FloatingActionButton
      : (_currentIndex == 1)
        ? FloatingActionButton(
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => AjouterStockView()),
            );
          },
          child: const Icon(Icons.add),
        ) // FloatingActionButton
        : null,
    bottomNavigationBar: BottomNavigationBar(
      currentIndex: _currentIndex,
      onTap: onTabTapped,
      items: const [
        BottomNavigationBarItem(
          icon: Icon(Icons.book),
          label: 'Livres',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.store),
          label: 'Stocks',
        ), // BottomNavigationBarItem
        BottomNavigationBarItem(
          icon: Icon(Icons.person),
          label: 'Auteurs',
        ), // BottomNavigationBarItem
      ],
    ), // BottomNavigationBar
  ); // Scaffold
}
```

```

Future<void> _deleteLivres(String id) async {
  await _livreServices.deleteLivres(id);
  _getLivres();
}

Future<void> _showConfirmationDialog(String id) async {
  await showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: const Text('Suppression d\'un livre'),
        content: const Text('Êtes-vous sûr de vouloir supprimer ce livre ?'),
        actions: <Widget>[
          TextButton(
            child: const Text('Annuler'),
            onPressed: () {
              Navigator.of(context).pop();
            },
          ), // TextButton
          TextButton(
            child: const Text('Supprimer'),
            onPressed: () {
              _deleteLivres(id);
              Navigator.of(context).pop();
            },
          ), // TextButton
        ], // <Widget>[]
      ); // AlertDialog
    },
  );
}

```

(Extrait de la page
liste_livre_view)

Ici, nous pouvons voir qu'avant la suppression d'un livre (si l'icone poubelle est cliquée), une pop up de confirmation s'affiche pour demander à l'administrateur s'il est vraiment sûr de vouloir supprimer ce livre.

Cela évite la suppression d'un livre par erreur.

Cette méthode est également présente pour les pages auteur et stock.

Paieement des dons

Libellé de la fonctionnalité : Paiement des dons via Stripe Payment

Explication de la fonctionnalité : L'application MyLibrary permet aux utilisateurs de faire des dons à l'association caritative MyLibrary. Pour cela, l'application utilise le service de paiement en ligne tel que Stripe. Les utilisateurs peuvent choisir le montant du don qu'ils souhaitent faire et le mode de paiement (carte de crédit, Paylinks, etc.). Une fois le paiement validé, le montant du don est transféré à l'association caritative MyLibrary.

Malheureusement l'intégration de Stripe (via l'extension ou les fonctions Google Cloud) requiert un abonnement payant au niveau de Firebase. J'ai donc choisi une solution de contournement : un formulaire que l'utilisateur remplit avec le montant de son don, la date et son id étant récupérés automatiquement. Et les données s'enregistrent directement sur Firebase Database.

Captures d'écran :

Paielements

+ Cr  er un paiement

Tous les paiements

Litiges

Toutes les transactions

Tous

3

R  ussis

3

Rembours  s

0

Non captur  s

0

En   chec

0

Date et heure

Montant

Devise

  tat

Moyen de paiement

Plus de filtres

Exporter

Modifier les colonnes

<input type="checkbox"/> Montant	Moyen de paiement	Description	Client	Date
<input type="checkbox"/> 1,00 �� EUR <div>R��ussi</div>	<div>VISA</div> **** 4242	pi_3P2gHoGfqTvT216y1pXQkvH	pierre.dupont@gmail.com	6 avr. �� 22:56
<input type="checkbox"/> 20,00 �� EUR <div>R��ussi</div>	<div>VISA</div> **** 4242	pi_3P2fXEGfqTvT216y1uHOV1Lz	pierre.dupont@gmail.com	6 avr. �� 22:08
<input type="checkbox"/> 5,00 �� EUR <div>R��ussi</div>	<div>VISA</div> **** 4242	pi_3P2YsiGfqTvT216y14aQYeiv	test.test@test.com	6 avr. �� 15:02

(Liste des paiements au niveau de Stripe)

Stripe propose un dashboard pour afficher le d  tail de chaque paiement ainsi que les clients ayant effectu   un don.

Faire un don

Montant de votre don

0,00   

En faisant    don    MyLibrary, vous nous aider    g  n  raliser et faciliter l'ac  s    la culture litt  raire....

Apple Pay

Ou payer par carte

E-mail

Informations de la carte

1234 1234 1234 1234

MM / AA

CVC

My Library

D  tails

Merci pour votre don

Un paiement en faveur de My Library appara  tra sur votre relev  .

MY LIBRARY 19,00   

Propuls   par stripe

Conditions d'utilisation Confidentialit  

(page de paiement Stripe)

L'utilisateur peut entrer le montant qu'il souhaite pour le don, le minimum   tant 1  .

home > dons > mbtkK3shtP4eh.

Plus de fonctionnalit  s dans Google Cloud

(default)	dons	mbtkK3shtP4ehTndCoRO
<div>+ Commencer une collection</div> <div>auteurs</div> <div>dons</div> <div>livres</div> <div>reservations</div> <div>stocks</div> <div>utilisateurs</div>	<div>+ Ajouter un document</div> <div>F7YbF7v8okuD3IXtdpWl</div> <div>T9xs��dNoIhGHD59Bj7Un</div> <div>mbtkK3shtP4ehTndCoRO</div>	<div>+ Commencer une collection</div> <div>+ Ajouter un champ</div> <div>date_donation: 3 avril 2024 �� 00:00:00 UTC+2</div> <div>montant: 20</div> <div>utilisateur: "KUmrP6kjXnV6vMH5gF7BnhaReKF2"</div>

(Enregistrement des dons au niveau de Firebase)

Extrait de code :

```
class _DonationViewState extends State<DonationView> {  
  final TextEditingController _amountController = TextEditingController();  
  final String paymentLink = 'https://donate.stripe.com/test_00g7vY0Is0c14wM9AA';  
  
  Future<void> _saveDonationAmount() async {  
    final amount = double.tryParse(_amountController.text);  
    await DonServices().addDon(amount!);  
    _amountController.clear();  
  }  
}
```

(Fonctions d'enregistrement de don ainsi que le lien de paiement Stripe)

(Fonctions d'ajout, modification et suppression d'un don)

Nous pouvons voir que des vérifications sont faites à chaque ajout de don pour vérifier la conformité des données. (Méthode _validateDon)

```
//region CRUD  
  
Future<List<Don>> getDons() async {  
  final QuerySnapshot snapshot = await _dons.get();  
  return snapshot.docs.map((doc) => Don.fromMap(doc.data() as Map<String, dynamic>)).toList();  
}  
  
Future<void> addDon(double montant) async {  
  final user = FirebaseAuth.instance.currentUser;  
  if (user == null) {  
    return;  
  }  
  
  final Don don = Don(  
    id: '',  
    montant: montant,  
    dateDonation: Timestamp.now(),  
    utilisateur: user.uid,  
  ); // Don  
  
  _validateDon(don);  
  await _dons.add(don.toMap());  
}  
  
Future<void> updateDon(Don don) async {  
  _validateDon(don);  
  await _dons.doc(don.id).update(don.toMap());  
}  
  
Future<void> deleteDon(String id) async {  
  await _dons.doc(id).delete();  
}  
//endregion  
  
//region Validation  
  
void _validateDon(Don don) {  
  if (don.montant <= 0) {  
    throw Exception('Le montant du don doit être supérieur à zéro');  
  }  
  if (don.dateDonation.toDate().isAfter(DateTime.now())) {  
    throw Exception('La date du don ne peut pas être dans le futur');  
  }  
}  
//endregion
```


Serveur de versions

Pour sauvegarder mon travail j'ai utilisé un serveur de version. Le serveur de version que j'ai utilisé est GitHub. Le projet se nomme my_library.

Voici le lien pour accéder au projet :

https://github.com/Melanie-devv/my_library.git

Axes d'amélioration

L'application MyLibrary est fonctionnelle et répond aux cahier des charges. Cependant, il est vrai que des axes d'amélioration sont toujours possible, par exemple :

Amélioration de l'interface: Bien que l'interface utilisateur actuelle soit fonctionnelle, il y a toujours place à l'amélioration. Des tests utilisateurs pourraient être effectués pour identifier les points faibles de l'interface utilisateur actuelle et apporter des améliorations en conséquence. Par exemple, l'ajout d'animations pour une meilleure expérience utilisateur, l'optimisation de la disposition des éléments pour une meilleure lisibilité, etc.

Nouvelles fonctionnalités : On pourrait envisager de nouvelles fonctionnalités telles que la possibilité de noter et de commenter les livres, la recommandation de livres en fonction des préférences de l'utilisateur, etc.

Expansion de la plateforme : L'application pourrait être développée pour d'autres plateformes telles que iOS et Web pour atteindre un public plus large.

Internationalisation : L'application pourrait être traduite dans différentes langues pour toucher un public plus large qui n'est pas forcément en France.

Thèmes : L'application pourrait proposer différents thèmes (thème sombre, thème clair, etc.).

Paielements automatisés : (à condition d'augmenter l'offre de Firebase). On pourrait automatiser, via l'API Stripe et l'extension Stripe dans Firebase, l'ajout et la gestion des données concernant les paiements sans devoir faire remplir au préalable un formulaire à l'utilisateur.

Conclusion

Dans le cadre de ce projet, nous avons développé une application mobile, MyLibrary, offrant diverses fonctionnalités pour faciliter la recherche, la réservation et la lecture de livres. L'application permet également aux utilisateurs de faire des dons à l'association caritative MyLibrary via Stripe.

Nous avons présenté en détail certaines des fonctionnalités clés de l'application, telles que l'affichage des livres et de leurs détails, la recherche de livres, la réservation de livres, le téléchargement de livres, la lecture de livres et le paiement des dons. Ainsi que toute une interface d'administration destiné au personnel de MyLibrary. Nous avons également discuté des axes d'amélioration possibles pour l'application, tels que l'amélioration de l'interface utilisateur ou l'ajout de fonctionnalités supplémentaires.

Dans l'ensemble, le développement de MyLibrary a été un projet passionnant et enrichissant. Nous avons pu mettre en pratique nos compétences en développement d'applications mobiles et en intégration de services tiers tels que Stripe. Nous sommes convaincus que MyLibrary sera une application utile et appréciée des utilisateurs passionnés de lecture. Nous prévoyons de continuer à améliorer l'application en fonction des commentaires des utilisateurs et des tendances du marché.

Épreuve E5 - Conception et développement d'applications (option SLAM)

ANNEXE 7-1-B : Fiche descriptive de réalisation professionnelle (recto)

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE		N° réalisation : 1
Nom, prénom : BULKAN Mélanie		N° candidat : 02046081832
Épreuve ponctuelle X	Contrôle en cours de formation	Date : 12 / 04 / 2024
Organisation support de la réalisation professionnelle EPSI Montpellier		
Intitulé de la réalisation professionnelle Application mobile de bibliothèque en Flutter		
Période de réalisation : Octobre 2023 à Mars 2024 Lieu : Montpellier		
Modalité : X Seul(e) En équipe		
Compétences travaillées Concevoir et développer une solution applicative Assurer la maintenance corrective ou évolutive d'une solution applicative Gérer les données		
Conditions de réalisation⁵ (ressources fournies, résultats attendus) Cahier des charge / liste de fonctionnalité MCD, UML Application fonctionnelle		
Description des ressources documentaires, matérielles et logicielles utilisées⁶ Android Studio, ordinateur personnel		
Modalités d'accès aux productions⁷ et à leur documentation⁸ OneDrive: https://tinyurl.com/ycxvtank https://drive.google.com/drive/folders/1b5zCziv6WhzlTwDavKmbWbAPkrd3cwn2?usp=sharing GIT : https://github.com/Melanie-devv/my_library Base de données : https://console.firebase.google.com/u/0/project/mylibrary-37fda/overview		

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

Voir la documentation

2.2 Les activités de l'organisation cliente s'appuient sur aux moins deux solutions applicatives opérationnelles permettant d'offrir un accès sécurisé à des données hébergées sur un site distant. Au sein des architectures de ces solutions applicatives doivent figurer l'exploitation de mécanismes d'appel à des services applicatifs distants et au moins trois des situations ci-dessous :

Éléments	Description de l'implantation dans le centre d'examen (nom du service ou de l'outil et caractéristiques techniques)	Remarques de la commission D'interrogation
Du code exécuté sur le système d'exploitation d'une solution technique d'accès fixe (type client lourd)		
Du code exécuté dans un navigateur Web (type client léger ou riche)	Visual studio code - Framework Symfony(HTML, CSS, Bootstrap, Twig, PHP)	
Du code exécuté sur le système d'exploitation d'une solution technique d'accès mobile	Android Studio – Framework Flutter (dart)	
Du code exécuté sur le système d'exploitation d'un serveur	Base de données Firebase et MySQL	

2.3 Une solution applicative peut être issue d'un développement spécifique ou de la modification du code d'un logiciel notamment open source.

2.4 Les solutions applicatives présentes dans le contexte sont opérationnelles et leur code source est accessible dans un environnement de développement opérationnel au moment de l'épreuve.