

Projet de Développement Mobile

Ludotroc



Front-end du projet ASP.NET Core MVC pour une application d'échanges et de prêts de jeux entre particuliers

avril 2022 - mai 2022

Milo Toumine et Lucie Cayment

SOMMAIRE

Contexte de l'application	2
Choix des aspects graphiques	2
Structure de l'application	4
Page d'accueil	5
Page de recherche	6
Page d'ajout	7
Page de messagerie et notifications	7
Page du profil	9
Structure du code	10
Stockage des informations	11
Gestion de projet	12
Conclusion et retour d'expérience	13

I. Contexte de l'application

Pour donner suite au projet de Génie Logiciel nous avons développé la partie frontend de notre API. L'API repose sur le concept de d'échange et de prêt de jeux de société et de puzzles entre particuliers. Le format de cette application sera communautaire et repose donc sur l'échange.

Les enjeux de cette partie frontend étaient notamment de réussir à présenter de manière pertinente les informations de l'API. Par manque de temps, aucune méthode CCU ne sera réalisée mais on aurait pu imaginé un tri de cartes en amont et des tests utilisateurs en aval du projet afin de challenger l'application.

II. Choix des aspects graphiques

La palette de couleurs qui constitue la charte graphique a été créée avec colors.io. Nous avons conçu cette charte graphique en favorisant l'accessibilité, grâce à notamment un contraste élevé. Nous avons souhaité que la palette soit coloré afin d'apporter du dynamisme à l'application. Les couleurs sont utilisées afin de distinguer les différents éléments.

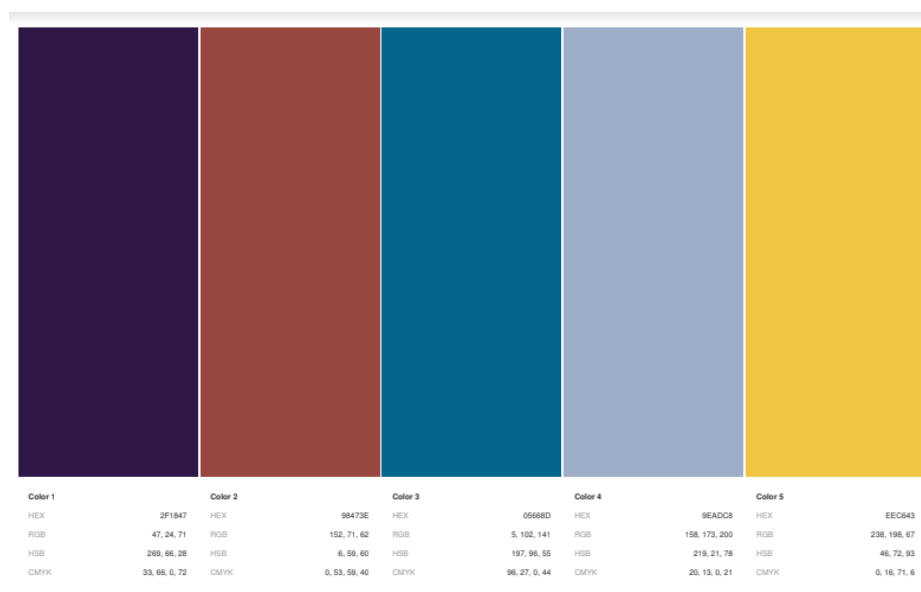
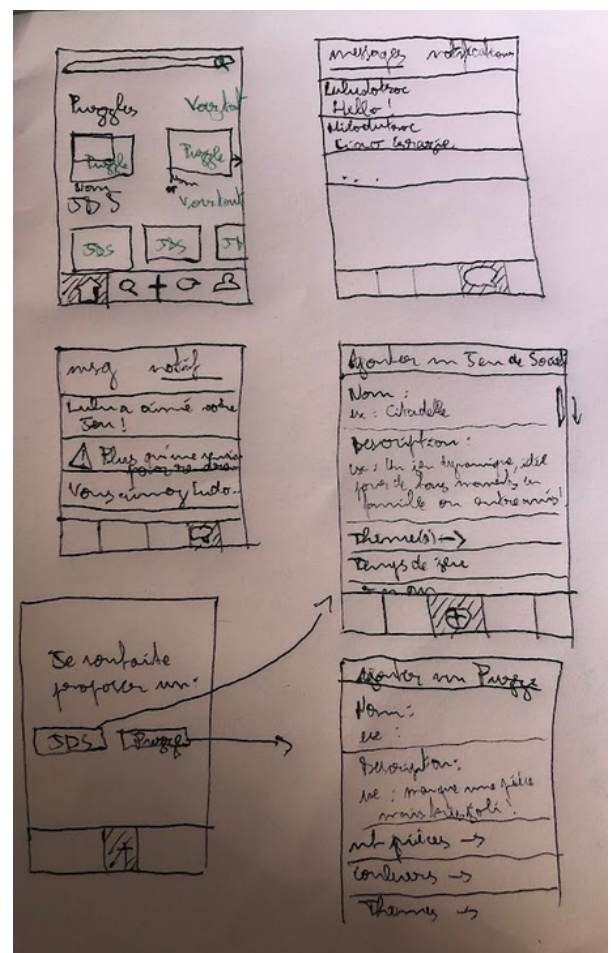
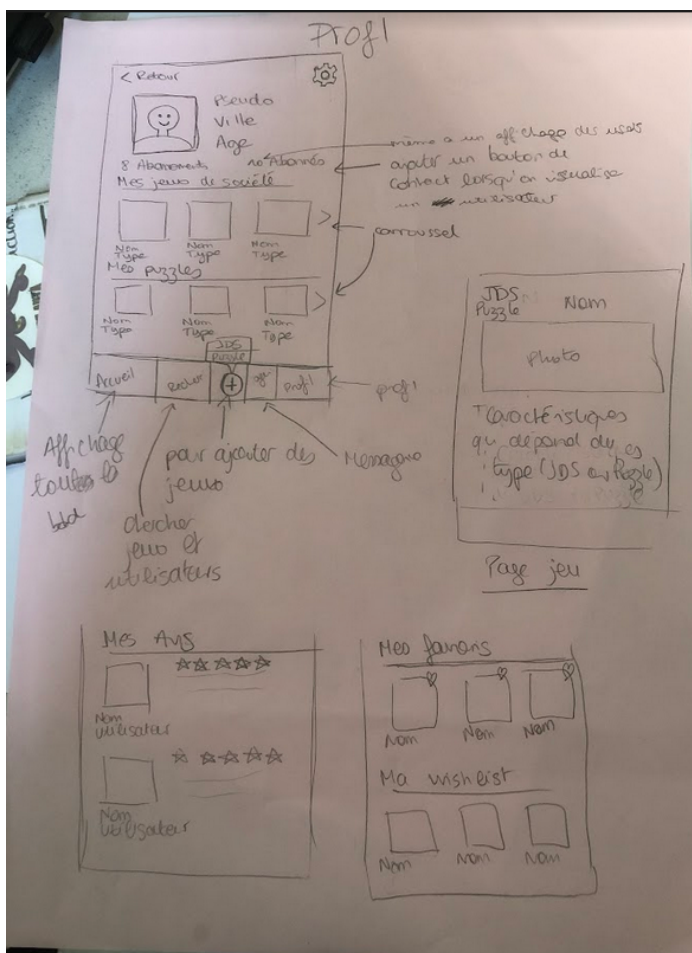


Figure 1 : Palette de couleur

Des maquettes ont ensuite été réalisées sur papier pour conceptualiser le visuel de l'application. Nous nous sommes inspirés de l'application mobile Vinted pour concevoir ces frameworks.

Les maquettes n'ont volontairement pas été plus développées car le projet ne durant pas longtemps, nous ne voulions pas passer trop de temps sur cette partie de préparation. La maquette sur la gauche présente le profil utilisateur ainsi que la page de présentation d'un jeu en particulier. La maquette sur la droite présente les onglets de l'accueil, de l'ajout et des messages/transactions.



Figures 2 & 3 : Maquettes papier

III. Structure de l'application

L'application est structurée grâce à une barre de navigation en bas de l'écran qui permet d'accéder à 5 sous-menu :

- la page d'accueil : ici sont regroupés l'ensemble des informations. Nous avons choisi d'afficher tous les jeux disponibles à l'échange (jeux de société et puzzles) sur la page, à l'instar d'une page d'actualités.
- la page de recherche : cette page permet de rechercher des jeux ou des utilisateurs directement puis de pouvoir visualiser la page spécifique au jeu ou à l'utilisateur en question.
- la page d'ajout : l'application étant liée à un compte utilisateur il est possible d'ajouter des jeux sur son profil. Nous avons décidé de placer cette fonctionnalité sur la barre de navigation car c'est finalement le cœur de l'application. Si aucun jeu n'est ajouté, l'application perd son sens. Cependant nous n'avons pas réussi à développer cette fonctionnalité.
- la page de messagerie et des notifications : un autre aspect important du projet est l'échange. Il était donc indispensable d'avoir une page messagerie regroupant les différentes transactions liées au profil utilisateur.
- la page du profil utilisateur : sur cette page on peut visualiser le profil de l'utilisateur principal. On y retrouve ses jeux, ses avis, ses favoris, etc.

On peut retrouver ces informations sur la mind map suivante.

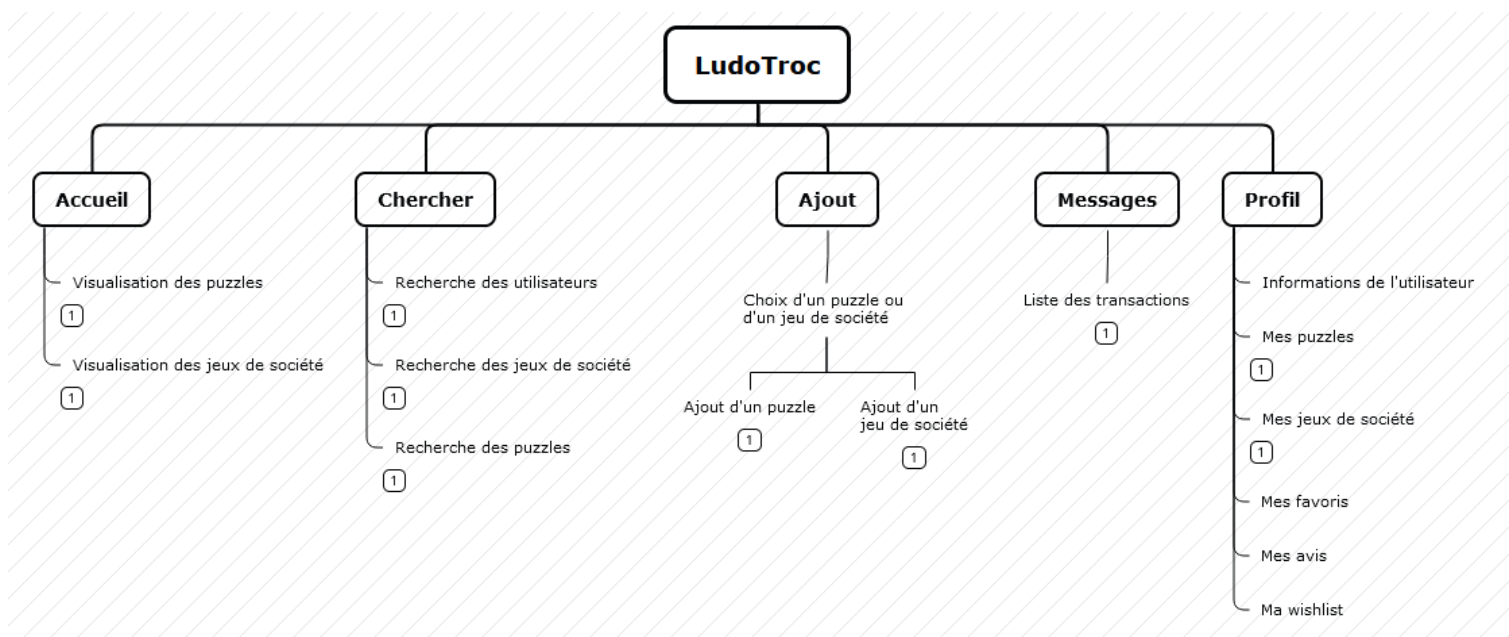


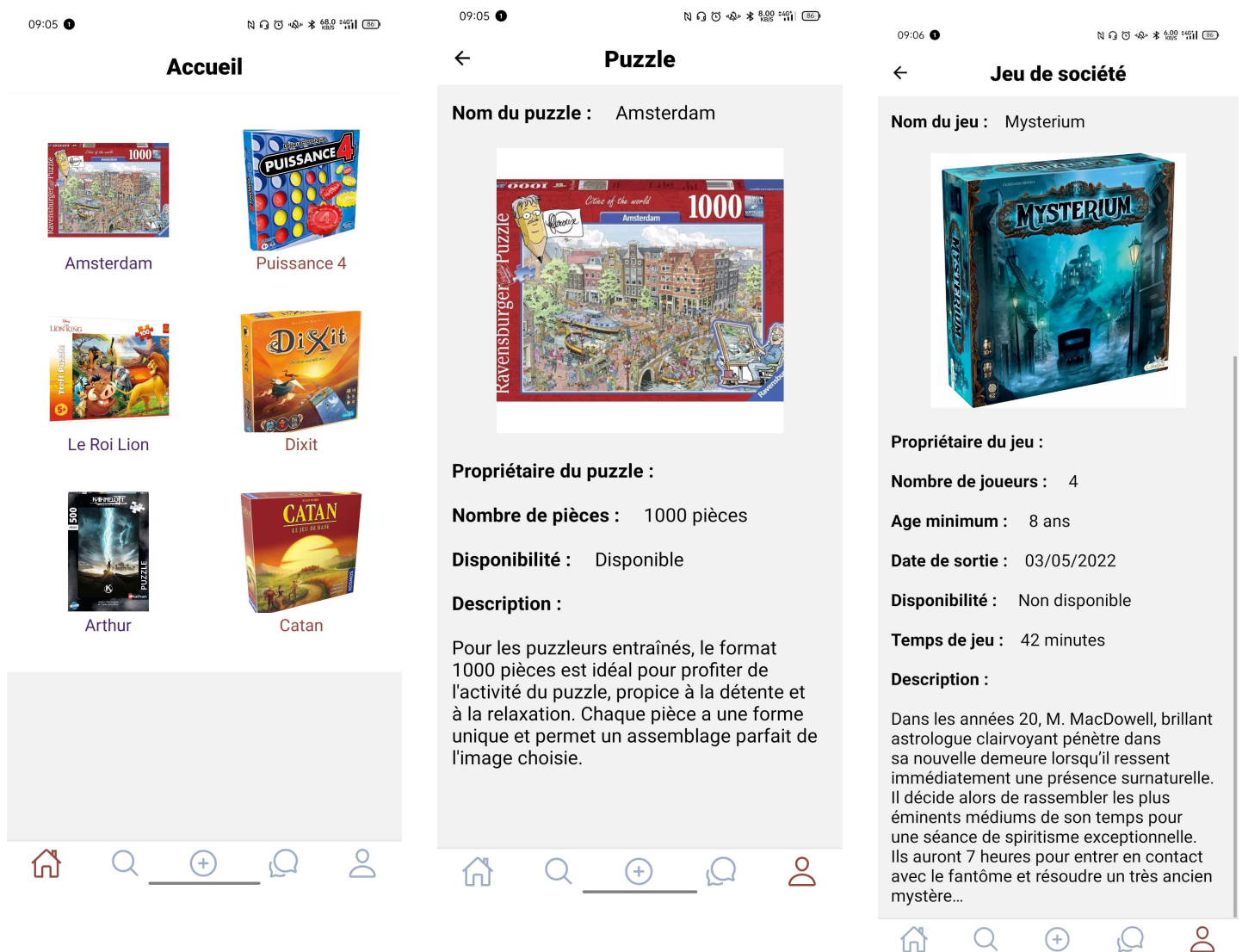
Figure 4 : Architecture de l'application

A. Page d'accueil

Cette page rassemble l'ensemble des jeux disponibles à la location. A l'instar d'une page d'actualités on pourrait y retrouver l'ensemble des informations concernant le jeu.

Un code couleur permet de différencier les puzzles des jeux de société en plus de leur emplacement et des photos. En effet les puzzles sont regroupés sur la colonne de gauche et les jeux de société sur la colonne de droite.

L'utilisateur a la possibilité de cliquer sur les images afin d'en savoir plus sur le jeu. Selon le type de jeu (puzzle ou jeu de société), différentes informations sont affichées, comme on peut le voir sur les captures d'écrans suivantes.



Figures 5, 6 & 7 : Accueil et détails

B. Page de recherche

Cette interface permet de rechercher plusieurs éléments :

- les utilisateurs
- les puzzles
- les jeux de société

L'ensemble de ces éléments a été affiché sur la page avec des carrousels pour chaque catégorie. Cette option n'est cependant viable qu'avec peu d'informations. Dans notre cas nous n'avons que 6 utilisateurs et une dizaine de jeux. Cette interface devra être repensée dans le cas où la densité de l'API augmente.

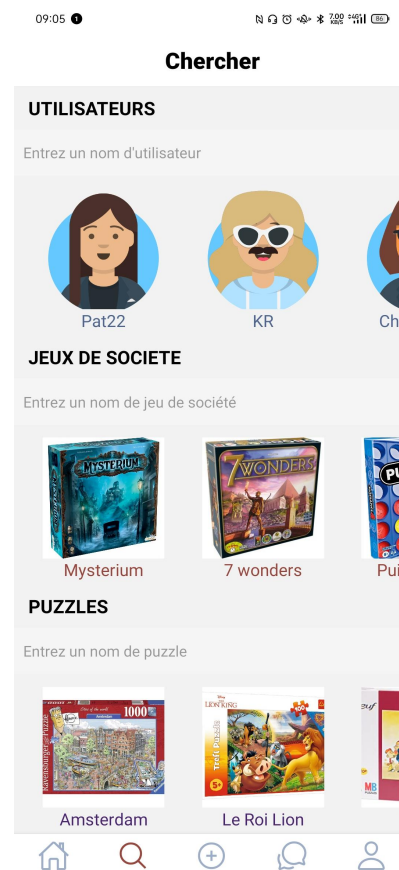


Figure 8 : Recherche

C. Page d'ajout

La page d'ajout est centrale car elle permet à l'utilisateur de proposer un jeu ou un puzzle à la communauté. Il s'agit du cœur de l'application car c'est en ajoutant du

contenu que les utilisateurs font vivre notre projet. Cependant la fonctionnalité d'ajout d'un jeu n'a pas pu être réalisée par manque de temps. Les pages ont cependant été développées.

Le même code couleur que celui présent sur la page d'accueil a été utilisé pour différencier un puzzle d'un jeu de société.

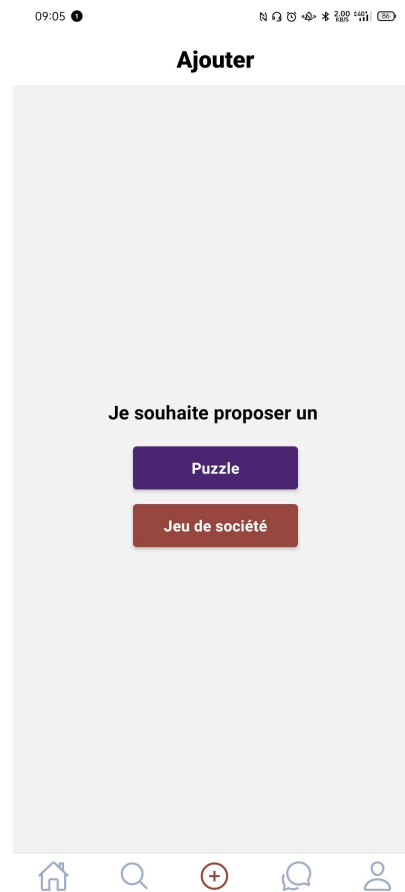


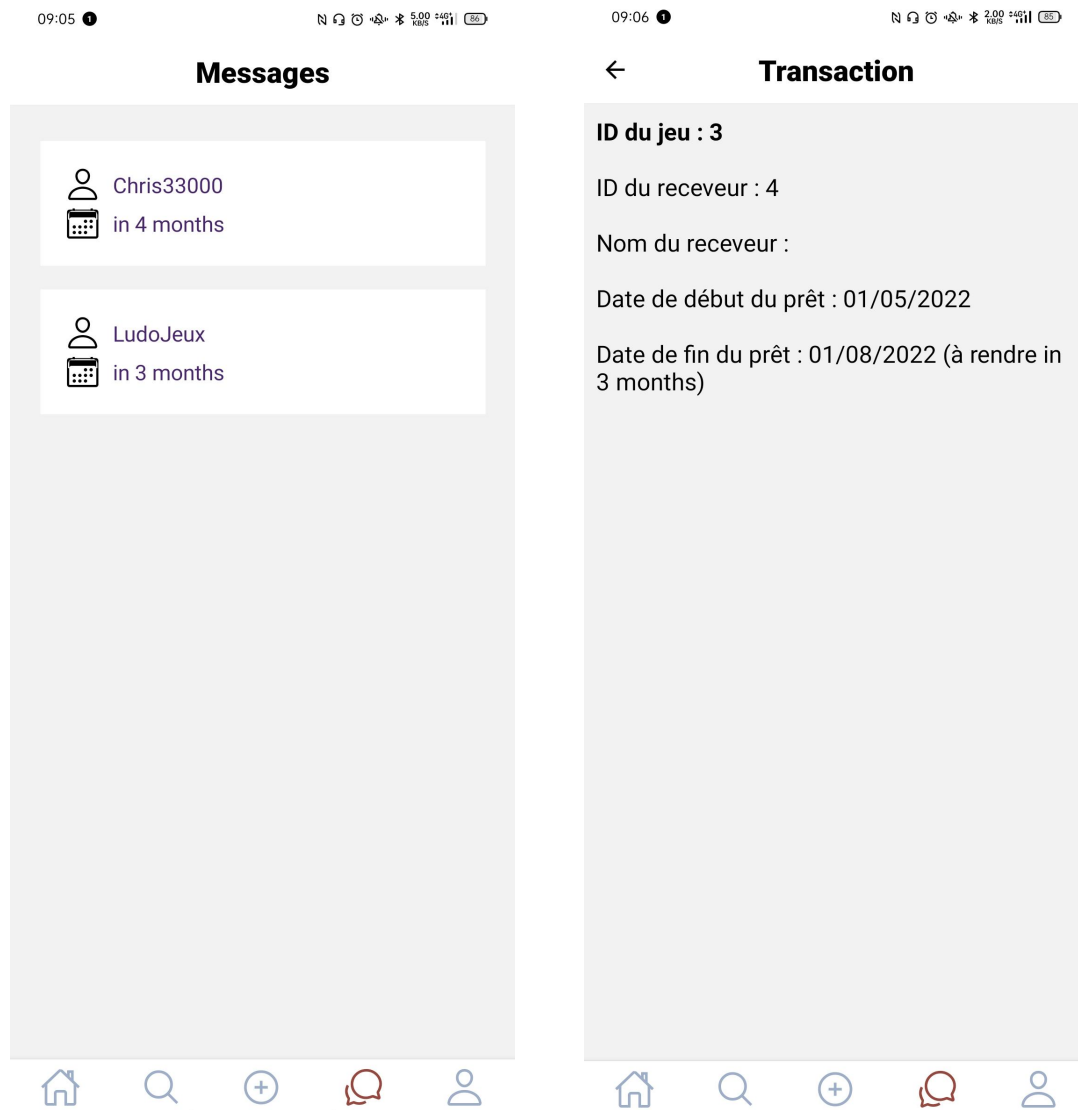
Figure 9 : Ajout

D. Page de messagerie et notifications

Cette page permet la visualisation des transactions entre les membres de LudoTroc. Idéalement cette page permet d'échanger et un onglet notifications les informe des activités concernant leurs jeux, leurs emprunts, leurs échanges, les jeux qu'ils convoitent...

Pour le moment, seulement la liste des transactions est affichée. Il est possible d'obtenir des détails en plus en cliquant sur une transaction donnée. De nouveau, cette sous-page n'as pas été trop poussée par manque de temps mais on peut

imaginer un affichage de plus d'informations, un lien vers le jeu emprunté et un système de messagerie.



Figures 10 & 11 : Messages et transactions

E. Page du profil

Comme précisé, la page profil présente le profil de l'utilisateur principal. En théorie, cet utilisateur se serait connecté via une authentification mais nous n'avons pas implémenté cette fonctionnalité dans notre application.

Le profil présenté est un profil choisi de manière aléatoire parmi l'ensemble des utilisateurs :

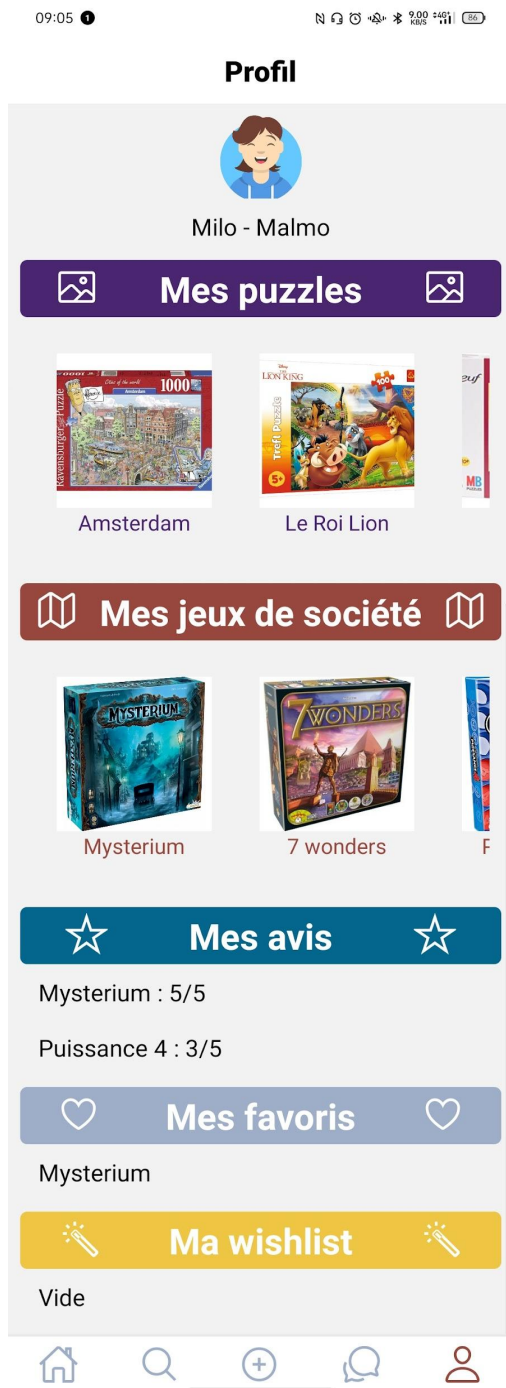


Figure 13 : Affichage du profil dans l'application

```
id: 6
pseudo: "Milo"
tel: "0909090909"
email: "milo@mail.fr"
localisation: "Malmo"
wishlist: null
```

Figure 12 : Données du profil dans l'API

Elle se structure en plusieurs catégories :

- les informations de l'utilisateur avec son pseudo et sa localisation
- les puzzles possédés par l'utilisateur
- les jeux de société possédés par l'utilisateur
- les avis laissés par l'utilisateur
- les favoris de l'utilisateur
- la liste de souhait de l'utilisateur

Tout comme depuis la page d'accueil et la page de recherche, il est possible de consulter les détails d'un jeu.

IV. Structure du code

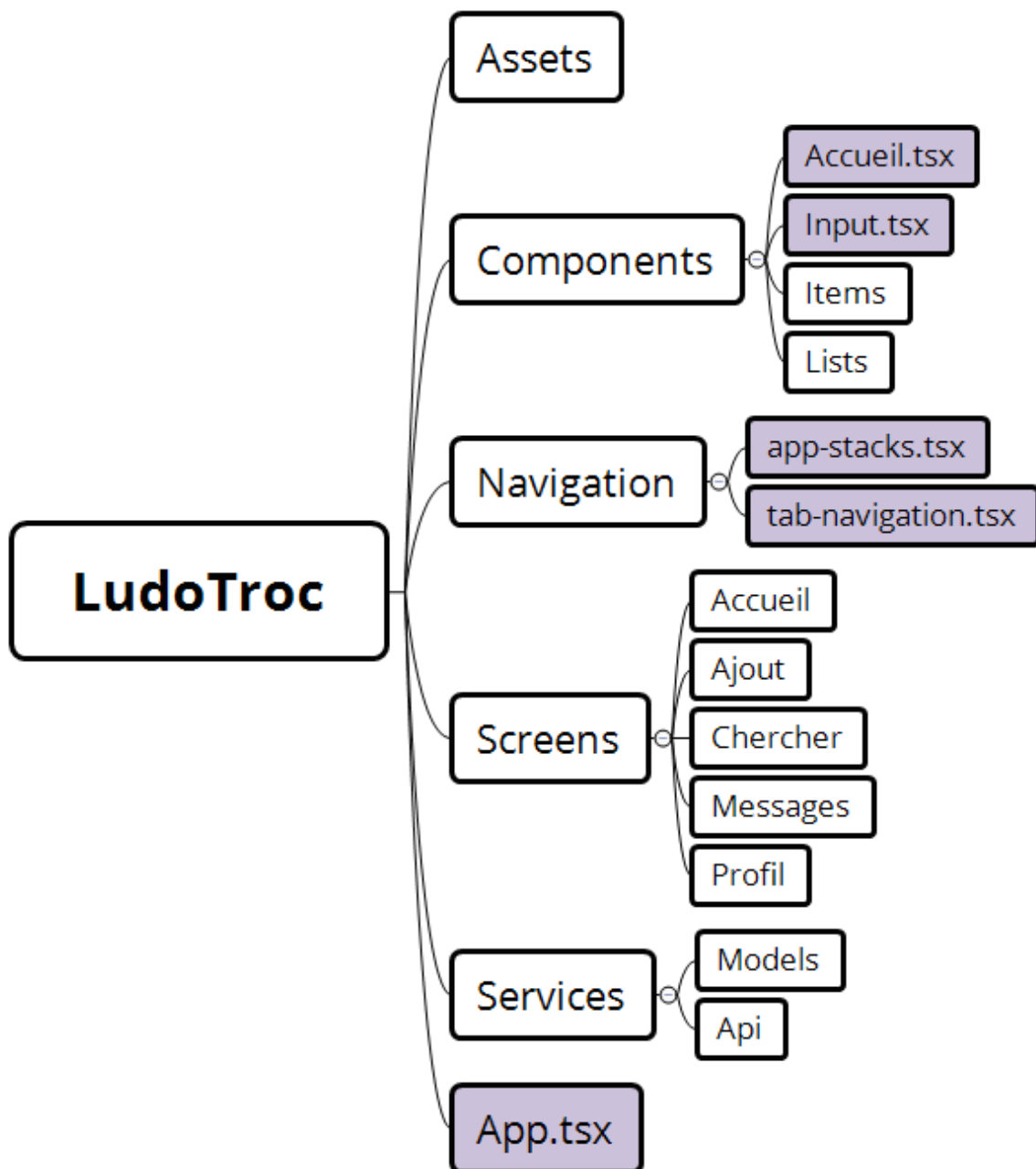


Figure 14 : Architecture du code

La carte organisationnelle ci-dessus décrit le squelette global de la solution codée pour ce projet. Les dossiers sont sur fond blanc et les fichiers sur fond violet.

Le dossier **assets** contient essentiellement des fichiers .png qui sont utilisés pour alimenter le visuel de l'application.

Le fichier **app-stacks** du dossier **navigation** définit les stacks permettant la navigation entre chaque page, sous page, les screens associés et les composants qui s'y trouvent. Le fichier **tab-navigation** définit la barre de navigation

(visuellement et fonctionnellement) qui permet d'accéder à chacune des 5 pages de l'application.

Le dossier **services** contient les fichiers .ts qui définissent les modèles des classes utilisées, ainsi que les fichiers qui récupèrent les données de notre API. Ces derniers décrivent également un certain nombre de méthodes qui servent à utiliser les données de manière pertinente dans l'application.

Les **screens** du dossier éponyme définissent les vues des pages et des sous-pages de l'application. Les méthodes des fichiers de services liés à l'API sont utilisées pour afficher les éléments de la base de données. Les différents composants du dossier **components** sont disposés dans ces vues.

V. Stockage des informations

Les informations sont disponibles via l'API que nous avons conçue au premier semestre. Cette API est disponible à l'url suivante :

<https://enscludotroc.azurewebsites.net/>

Il est possible via cette interface d'ajouter, modifier ou supprimer les données.

Les données brutes sont obtenues via différents liens en lien avec l'API. Par exemple pour accéder aux jeux de société on utilisera la lien suivant :

<https://enscludotroc.azurewebsites.net/api/JDSApi/>

Il en va de même pour les autres modèles.

L'API a été modifiée pour y ajouter une méthode de recherche par nom dans plusieurs Controllers. Nous l'avons donc redéployé pour profiter de ces nouvelles fonctionnalités.

Les photos présentes sur l'application ne proviennent en revanche pas de l'API, et sont simplement stockées dans des assets.

VI. Gestion de projet

Ce projet a duré du 15 avril au 15 mai. La plupart du travail s'est fait à distance et a donc un peu complexifié son déroulement.

Les premières séances en TP ont été dédiées à mettre en place le squelette de l'application, la barre de navigation, et la navigation en elle-même. Il nous semblait également nécessaire de réussir à déployer notre API en ligne afin de pouvoir travailler plus facilement par la suite.

Le planning initial est présenté ci-dessous. Aucune date n'a été donnée pour la réalisation des différents aspects de l'application car nous voulions pouvoir nous adapter à nos emplois du temps particuliers et rester flexibles. Nous nous sommes répartis le travail entre les éléments en vert et en jaune afin d'éviter le plus possible des possibles conflits dans le code.

	Début 15/04/2022	Milo
	Fin 15/05/2022	Lucie
Dates	Tâches	Etat
15/04/2022	Déployer bdd avec Azure	<input checked="" type="checkbox"/>
	Maquettes papier	<input checked="" type="checkbox"/>
	Copier TP4 dans le projet	<input checked="" type="checkbox"/>
22/04/2022	Dernière séance de TD	<input checked="" type="checkbox"/>
vers le 27/28 avril	Point d'avancement	<input checked="" type="checkbox"/>
TODO	Navigation Home, Search, Messages, Profil	<input checked="" type="checkbox"/>
	Barre Nav (avec le bouton + et JDS/Puzzle)	<input checked="" type="checkbox"/>
	Page Profil	<input checked="" type="checkbox"/>
	Page Jeu Zoom	<input checked="" type="checkbox"/>
	Page Recherche Puzzle	<input checked="" type="checkbox"/>
	Page Recherche JDS	<input checked="" type="checkbox"/>
	Page Ajout	<input checked="" type="checkbox"/>
	Page Accueil	<input checked="" type="checkbox"/>
	Page Messagerie	<input checked="" type="checkbox"/>
15/05/2022	Rendu	<input checked="" type="checkbox"/>

Figure 15 : Planning prévisionnel

La répartition indiquée a finalement été bousculée par des problèmes techniques sur l'ordinateur personnel de Milo. La grande majorité du code (celle écrite hors des TPs, donc après la fin des partiels) a donc été écrite par Lucie.

VII. Conclusion et retour d'expérience

Nous sommes globalement satisfaits du rendu final de l'application. Travailler sur la partie frontend nous a cependant fait réaliser qu'il manquait des éléments dans notre API.

Il est cependant dommage que la période de projet soit si courte et positionnée en partie sur une période de révision et des partiels. Nous aurions souhaité pouvoir rendre l'application encore plus complexe, en y ajoutant par exemple la possibilité d'ajouter un jeu ou de demander à en emprunter un. Les jeux ne sont également pas liés à leur propriétaire, ce qui aurait été préférable.