



Projet développement mobile

L'application mobile Astroshare

Clément Weinreich & Angeline Rondeau
ENSC - 2ème année
Groupe 3

—

11/05/2022

Table des matières

Table des matières	2
Contexte	3
Organisation du travail	3
Modification apportée à l'API Astroshare	5
Hébergement des données	5
Organisation générale de l'application	6
AstroHome	6
Add	7
Discover	8
Architecture du code source	9
Assets	9
Components	10
Navigation	11
Screens	11
Services	11
Fonctionnalités disponibles	12
Consultation d'astrophotos	12
Recherche d'astrophotos	12
Afficher les détails d'une astrophoto	12
Poster une astrophoto	12
Découvrir une astrophoto aléatoirement	13
Supprimer une astrophoto	13
Conclusion	13

Contexte

Dans le cadre du module développement mobile, nous avons créé une application capable d'utiliser les données d'une API. Ce projet s'inscrit dans la continuité du projet de génie logiciel du semestre 7 qui avait pour but de créer une application Web selon le modèle MVC ainsi qu'une API. C'est cette API qui constitue la base d'informations de notre application mobile.

L'attrait pour l'astronomie d'un des membres du groupe a dirigé notre projet vers ce domaine. Nous avons décidé de créer une application de partage de photos d'astrophotographie. Dans un premier temps, nous voulions créer un réseau social centré sur le contenu (comme Reddit par exemple) et non sur les utilisateurs. Toutefois les difficultés rencontrées pour mettre en place une authentification au niveau de l'API a réorienté notre projet. Voici ce que nous souhaitons mettre en place après avoir abandonné l'authentification d'utilisateurs :

L'application mobile ne dispose donc pas d'authentification, mais permet tout de même de visualiser, rechercher, poster et supprimer des photos d'astronomie, ce qui reste le cœur de notre projet.

Organisation du travail

Dans un premier temps, nous nous sommes réunis à plusieurs reprises pour définir les objectifs et l'apparence de l'application. Nous avons réalisé des maquettes pour différents écrans sur papier, et réalisé une liste de tâches à réaliser pour atteindre nos objectifs.

Ensuite, nous avons fait du pair programming pour déployer l'API, la mettre en lien avec l'application et développer la base de chaque écran. Nous nous sommes ensuite réparti les tâches relatives aux différents écrans de l'application :

- Home Screen :
 - Barre de recherche : Clément
 - Requêtage de l'API : Angeline & Clément

- Affichage des éléments : Angéline
- Front : Angeline
- Details Screen:
 - Front : Angeline
 - Affichage des informations dans le bon format : Angéline
 - Requêtage de l'API : Angeline
 - Suppression d'une image : Clément
- Add Screen:
 - Formulaire : Clément
 - Requêtage de l'API : Clément
 - Front : Clément
- Let's Discover:
 - Bouton de recherche aléatoire : Clément
 - Requêtage de l'API : Clément
 - Front : Similaire au Details Screen

Apparence générale de l'application (header, navigations, etc.) : Angeline et Clément

Nous avons utilisé github pour versionner notre code et travailler en simultané avec une gestion des conflits efficace. Les premières séances de travail ont été réalisées en présentiel, mais la suite a été réalisée principalement à distance.

Voici le planning effectif du projet :

SEMAINE DU PROJET :			AVR.			MAI	
			11	18	25	2	9
1	REFLEXION ET PRÉPARATION	- Définition du projet et des tâches à réaliser - Maquettage					
2	DEVELOPPEMENT	- Amélioration de l'API puis déploiement - Services - Écrans					
	RAPPORT						

Modification apportée à l'API Astroshare

A la fin du projet de génie logiciel, l'API n'était pas prête à être utilisée telle quelle pour le projet de développement mobile. Initialement nous avons fait le choix de stocker les photos dans un répertoire, avec le nom de l'image dans la base de données. Cela n'était pas vraiment compatible avec l'envoi de photos par l'API. C'est pourquoi nous avons modifié l'API afin de stocker l'image encodée au format base64. De plus, nous avons complété l'API en ajoutant et en corrigeant différents points d'accès que nous souhaitions implémenter en décembre, mais qui n'ont pas pu être réalisés par manque de temps :

- **PhotosApi/search/{searchString}** : Permet de rechercher un élément en filtrant parmi tous les champs d'une astrophoto
- **PhotosApi/random/{n}** : Permet de récupérer aléatoirement un nombre n de d'astrophotos
- **PhotosApi/recent/{n}** : Permet de récupérer les n photos les plus récentes
- **PhotosApi/categories** : Retourne un tableau contenant le nom de chaque catégorie d'astrophoto (avec l'énumération Category)
- **PhotosApi/categories/{index}** : Retourne les photos appartenant à une catégorie. L'index correspond au numéro permettant d'identifier la catégorie dans l'énumération Category.

Hébergement des données

L'application web MVC et l'API a été développée en .NET 5.0 avec une base de données SQLite.

L'application web et l'API sont hébergées sur Azure à l'adresse :

<https://enscastrosharev2.azurewebsites.net/>.

Concernant les données métiers de l'application qui sont stockées en base, on a une classe Photo et une énumération Category.

+ Photo	
+ Id	INT
+ Title	VARCHAR(50)
+ Description	TEXT
+ Category	Category
+ OpticalInstrument	VARCHAR(200)
+ EyePiece	VARCHAR(50)
+ Mount	VARCHAR(50)
+ CameraSettings	TEXT
+ PostProcessing	TEXT
+ AdditionalDetails	TEXT
+ ShootingDate	DATE
+ ShootingPlace	VARCHAR(50)
+ Photographer	VARCHAR(50)
+ ImageName	VARCHAR(50)
+ ImageBase64	VARCHAR(50)

+ enum : Category
+ PLANET
+ STAR
+ GALAXY
+ NEBULA
+ DUSTCLOUD
+ NATURALSATELLITE
+ ARTIFICIALSATELLITE
+ STARCLUSTER
+ ASTEROID
+ COMET
+ METEOR
+ OTHER

Diagramme de classe des données

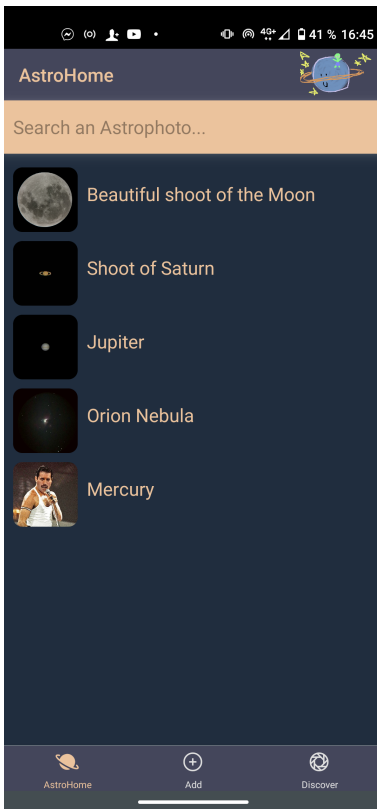
Organisation générale de l'application

L'application contient 3 onglets : AstroHome, Add et Discover.

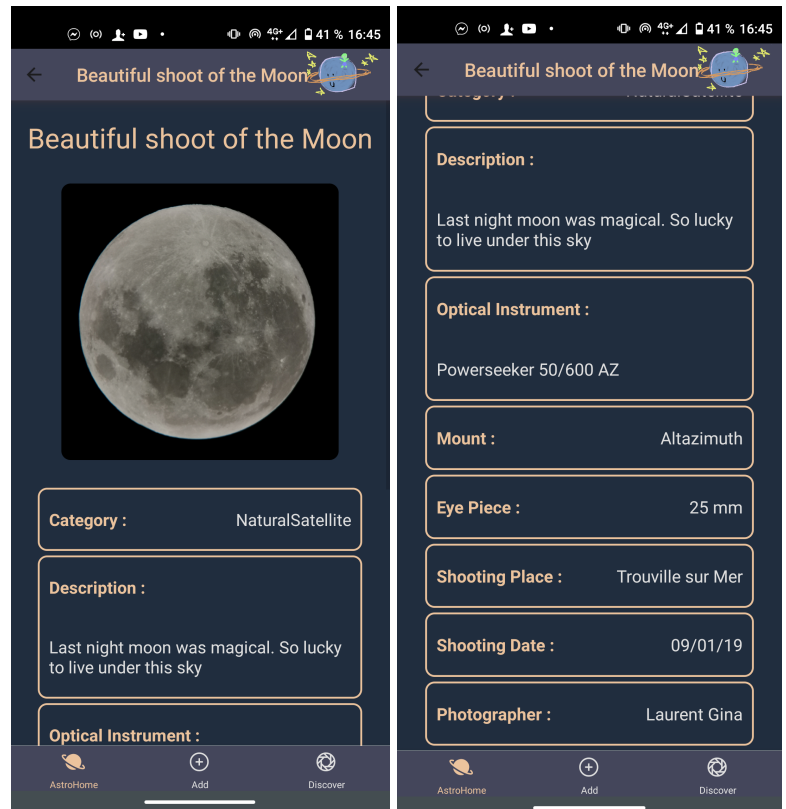
AstroHome

Cet onglet contient une pile de 2 écrans :

- Home avec une barre de recherche et l'affichage des photos postées sur l'application.
- Details qui permet d'afficher une photo ainsi que des informations plus précises sur celle-ci



Home Screen

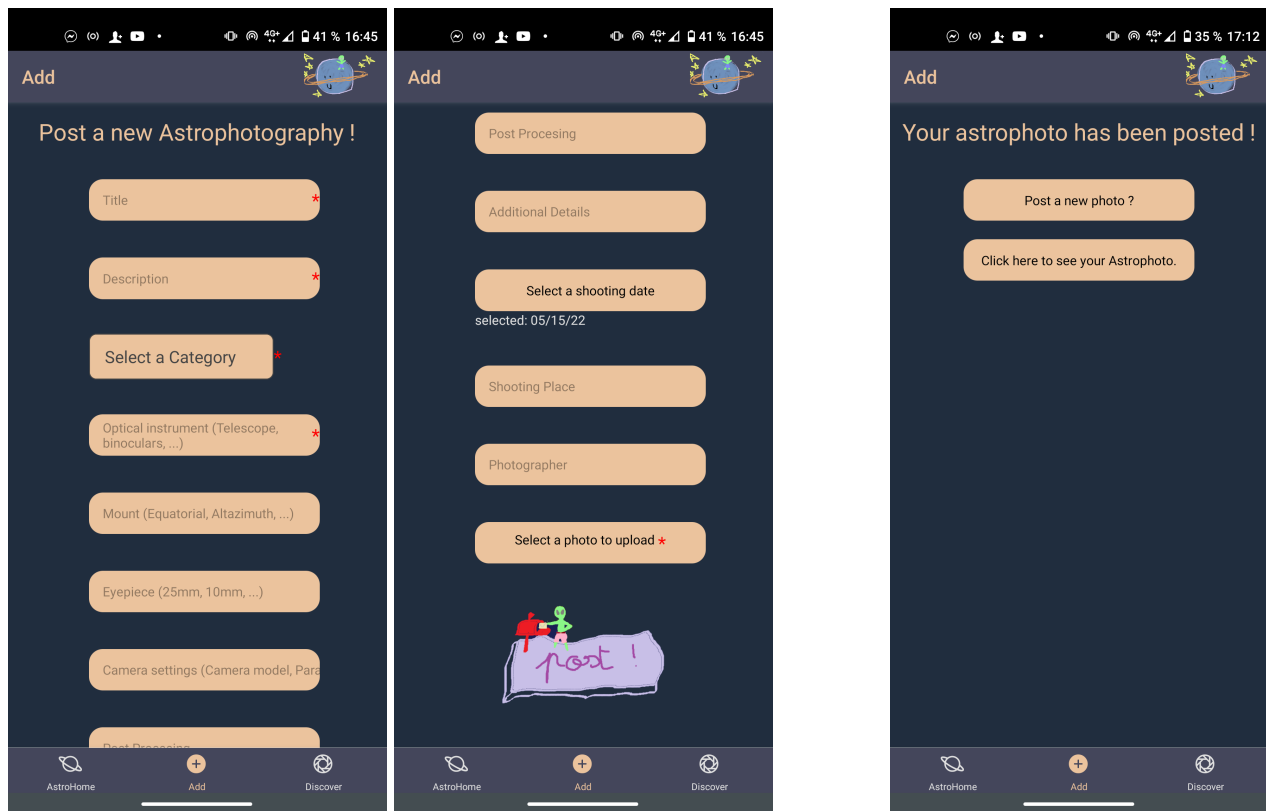


Details Screen

Add

L'onglet Add permet de poster une astrophoto, il contient une pile de 2 écrans :

- Add avec un formulaire permettant de poster une astrophoto
- Details qui permet d'afficher les détails de la photo qu'on vient de poster



Add Screen

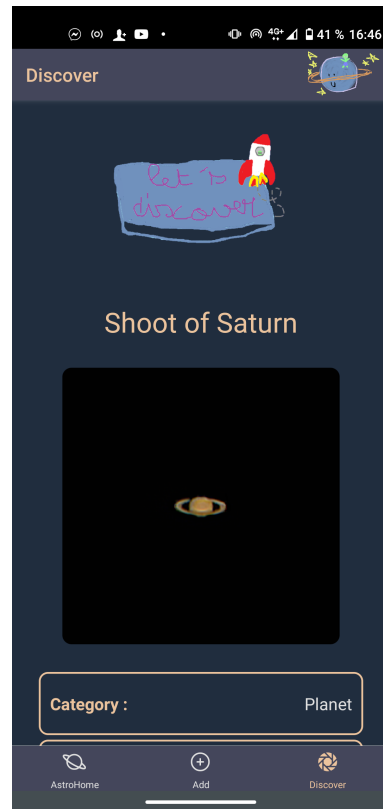
Discover

L'onglet Discover permet d'afficher une astrophoto aléatoirement en appelant le composant d'affichage de détail. Cet onglet contient une pile d'un seul écran :

- Discover qui contient le bouton d'affichage aléatoire d'une photo



Discover Screen
avant clique sur "Let's Discover"



Discover Screen
après clique sur "Let's Discover"

Architecture du code source

La base du code source a été générée avec expo. Nous avons choisi de détailler certaines parties de l'architecture du code source, c'est-à-dire les répertoires assets, components, navigation, screens, services. Le contenu de l'application est lancé depuis le fichier App.tsx.

Assets

Le répertoire assets contient les images utilisées dans notre application. Ici ce sont principalement des logos, ou des images utilisées comme boutons :



Ces images ont été réalisées par des artistes peintres renommés (nous) sur un outil mondialement reconnu (paint).

Components

Le répertoire composants contient tous les composants appelés dans les différents écrans :

- **AstrophotoDetails** : affiche une astrophoto avec ses informations
- **AstrophotoInformation** : affiche une information d'une astrophoto pour la page details
- **AstrophotoList** : affiche une liste d'astrophoto.
- **AstrophotoItem** : est l'item affiché par AstrophotoList. Un item représente une astrophoto avec son image en miniature et son titre.
- **SearchBar** : est la barre de recherche affichée dans le Home Screen.
- **DateInput**, **DropdownCategories**, **PhotoUploader** et **Input** sont des éléments du formulaire d'ajout d'une astrophoto dans l'application qui permettent respectivement de sélectionner la date de prise de la photo, sélectionner la catégorie de l'objet céleste observé, téléverser une photo depuis la catégorie, et enfin de rentrer des informations textuelles.
- **LogoHeader** est l'affichage du logo de l'application en haut à droite du header sur chaque page

Navigation

Ce répertoire permet de définir les composants de navigations, les piles et les propriétés du header et du footer. Il contient deux fichiers :

- **app-stack.tsx** : Définition des différentes piles de navigation et de leurs propriétés
- **tab-navigation.tsx** : Définition des propriétés de navigation depuis le tab-navigator du footer

Screens

Ce répertoire screen contient la définition des écrans évoqués précédemment. On y trouve alors 4 fichiers :

- **AddScreen.tsx** : Définit le composant affiché sur l'écran de post d'une astrophoto.
- **AstroHomeScreen.tsx** : Définit le composant affiché sur l'écran d'accueil, avec la barre de recherche et l'affichage d'une liste de d'astrophotos
- **DetailsScreen.tsx** : Définit le composant qui affiche les détails d'une astrophoto
- **DiscoverScreen.tsx** : Définit le composant qui permet d'afficher des astrophotos aléatoirement.

Services

Le répertoire service permet de faire le lien avec la partie métier de l'application, et avec l'API. On y retrouve 2 fichiers :

- **astrophoto.service.ts** : Définit l'interface d'une Astrophoto. Cette interface correspond à la définition d'une Astrophoto dans le modèle de l'API. La seule différence réside dans la catégorie, qui est une énumération dans l'API, et un string dans la définition de l'interface.

- **Astrophotodbapi.service.ts** : Ce fichier contient les méthodes qui requêtent l'API, et qui sont par la suite utilisées par les différents composants.

Fonctionnalités disponibles

Les fonctionnalités disponibles ont été évoquées précédemment. Nous allons les détailler plus précisément dans cette partie.

Consultation d'astrophotos

Il est possible de consulter les photos postées depuis la page d'accueil AstroHome. Par défaut, elles sont toutes affichées grâce au composant AstrophotoList.

Recherche d'astrophotos

Il est possible de rechercher des photos grâce à des mots clés dans une barre de recherche. La recherche par mots clefs concerne tous les champs descriptifs des photos.

Afficher les détails d'une astrophoto

Il est possible de consulter les détails d'une astrophoto qui met en forme chacune des informations correctement. En effet, pour poster une photo, certains champs sont facultatifs. Un affichage conditionnel permet donc d'afficher les éléments et leurs libellés uniquement s'ils ne sont pas vides.

Poster une astrophoto

Il est possible de poster des astrophotos depuis l'application. Un formulaire permet de rentrer toutes les informations nécessaires (champs textes, date, liste déroulante de catégories, image à poster, etc.) Une fois que l'astrophoto est postée, on peut en poster une nouvelle ou aller directement voir les détails de la photo.

Découvrir une astrophoto aléatoirement

Il est possible de découvrir une astrophoto aléatoire parmi toutes celles postées dans l'application. En appuyant sur un bouton, les détails d'une astrophoto choisie aléatoirement seront affichés en dessous.

Supprimer une astrophoto

Il est possible de supprimer une astrophoto. Dans une page de détails d'astrophoto, un bouton présent en dessous des informations permet de supprimer la photo. Une fois la photo supprimée, il faut recharger la page principale (en faisant une recherche vide par exemple) pour la voir disparaître. De plus, si l'utilisateur réussit à accéder à la page de détail de la photo supprimée, "Not found" sera affiché sur les informations.

Conclusion

Pour conclure, nous avons réellement apprécié la réalisation de ce projet. Nous n'avons pas réellement rencontré de difficultés mais nous avons appris à mieux nous servir du framework react native en implémentant des fonctionnalités que nous n'avions pas vu en cours. Pour rendre complète l'application, nous aurions aimé ajouter une gestion d'utilisateurs, afin que ceux-ci puissent modifier et supprimer leurs photos. C'est pourquoi, nous avons créé d'autres fonctionnalités annexes comme le Discover qui permet de rendre l'application plus complète.