

WPHaven

Application de fonds d'écran

Rapport du travail de semestre

Florent BENEY & Clément CHRISTENSEN

CFPT Informatique

Version	Détails	Date
1.0	Squelette	15 octobre 2020
2.0	Contenu	22 octobre au 19 novembre 2020
3.0	Style et corrections	20 au 22 novembre 2020
Finale	Finalisation	22 novembre 2020

1. Tables des matières

1. Tables des matières	3
2. Résumé	4
3. Remerciement	4
4. Introduction	4
5. Cahier des charges	5
5.1. Titre	5
5.2. Objectif	5
5.3. Matériels et logiciels	5
5.4. Prérequis	5
5.5. Descriptif complet du projet	5
6. Étude d'opportunité	7
6.1. Le besoin de l'organisation	7
6.2. Analyse de l'existant	7
6.3. Différentiation de la concurrence	7
7. Analyse fonctionnelle	8
7.1. Fonctionnalités disponibles	8
7.2. Description des interfaces	9
8. Analyse organique	13
8.1. Architecture du projet	13
8.2. Description des fonctions	15
9. Les tests	17
10. Conclusion	21
10.1. Bilans personnels	21
10.2. Bilan de la méthodologie	21
10.3. Conclusion sur l'application	22
11. Références	23
12. Annexes	23

2. Résumé

Pour ce travail de semestre, nous avons décidé de faire une application React Native utilisant l'API de Wallhaven, afin de trouver des fonds d'écran adaptés au format d'écran sur lequel l'application est installée.

Sur l'application, l'utilisateur aura le choix ou non de s'inscrire. Il est limité dans les fonctionnalités disponibles s'il n'est pas connecté : il peut afficher les 24 dernières images correspondant au format de son téléphone, et les afficher une image en détails avec la résolution ainsi que le lien vers l'image sur Wallhaven. Une fois connecté, il a également accès à une sélection d'images similaires lors de l'affichage d'une image. Ainsi, qu'une fonction de recherche et de mise en favoris afin de les retrouver plus aisément à l'avenir.

Pour réaliser ce projet, nous avons utilisé une base de données reliée à un service web qui communique avec l'application pour recevoir les résultats des requêtes en JSON. La communication avec l'API se fait directement via l'application, avec un retour des données également en JSON.

Avec l'omniprésence des smartphones aujourd'hui, la maîtrise du développement mobile et la proposition d'une application de personnalisation nous semblait un mélange idéal pour un travail de semestre.

3. Remerciement

La structure du document a grandement pu être améliorée grâce à l'aide de Monsieur Francisco GARCIA.

4. Introduction

Le travail de semestre est organisé dans le but de s'entraîner pour le travail de diplôme Technicien ES en informatique. Ce travail est une occasion pour s'entraîner avec de nouvelles technologies ou renforcer ses connaissances. Ce travail peut s'effectuer en groupe ou seul.

Pour ce travail nous avons décidé de le faire à 2 et d'approfondir nos connaissances sur le développement mobile. Pour ce faire, nous avons choisi d'utiliser le framework React Native.

Ce framework a été créé par Facebook en 2015 pour développer des applications Android et iOS en utilisant leur autre framework créé en 2013 : React.

5. Cahier des charges

5.1. Titre

WPHeaven

5.2. Objectif

Réaliser une application mobile (Android & iOS) permettant de trouver des fonds d'écrans

5.3. Matériels et logiciels

Matériels	Logiciels
<ul style="list-style-type: none"> • 2 postes standards techniciens • 1 Mac • 2 smartphones (1 Android et 1 iOS) 	<ul style="list-style-type: none"> • Visual Studio Code (IDE) • GitHub • Atom • PhpMyAdmin

Tableau 1 - Matériels et logiciels

5.4. Prérequis

- React Native → JavaScript
- API Wallhaven
- API Rest → JSON

5.5. Descriptif complet du projet

Planification

Planning en Gantt Waterfall :

teamgantt
Created with Free Edition

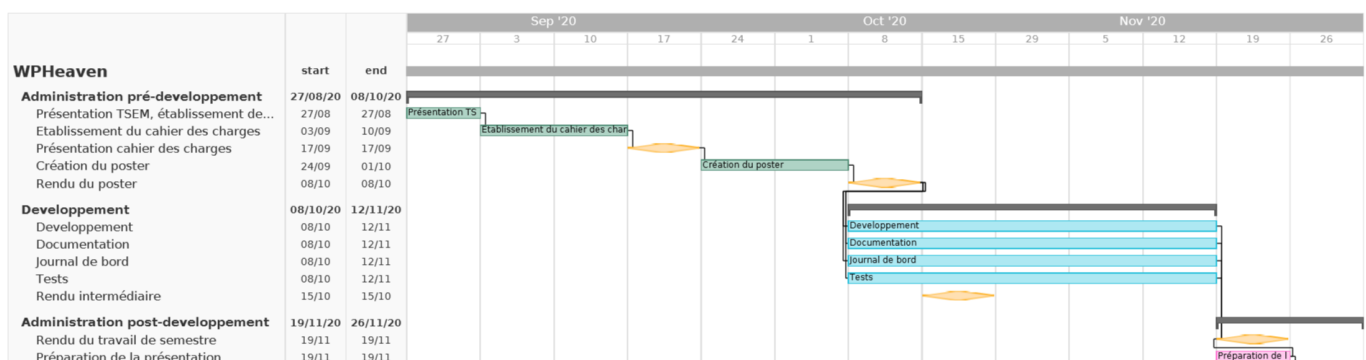


Figure 1 - Planning initial

Méthodologie

Méthode Agile (SCRUM)

Versionning : GitHub

Description de l'application

- Lorsque l'utilisateur n'est pas connecté
 - Afficher les 24 derniers fonds écrans en fonction de la taille de l'appareil
 - Partage d'images
- Fonctionnalités supplémentaires pour l'utilisateur connecté
 - Recherche de fonds d'écrans par :
 - Tag
 - Couleur
 - Résolution selon le ratio d'écran
 - Utilisateur
 - Catégories
 - Affiche des fonds d'écran selon un tri
 - Aléatoire
 - Ordre ascendant / descendant
 - Nombre de likes
 - Nombre de vues
 - Date d'ajout
 - Gestion des favoris
 - Enregistrement possible d'image
 - Proposition d'images similaires

Modèle de données

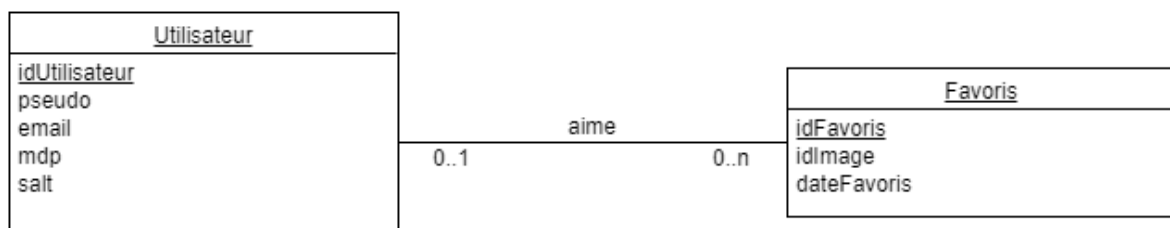


Figure 2 - Modèle de données prévisionnel

Livrables

- Mind Map
- Planning
- Rapport de projet
- Manuel utilisateur (si applicable)
- Journal de travail ou LogBook
- Poster
- Résumé / Abstract

6. Étude d'opportunité

6.1. Le besoin de l'organisation

Les raisons qui poussent à l'action

La présence du smartphone et son utilisation a explosé en quelques années. Aujourd'hui, il y a plus de smartphones actifs que d'être humains sur Terre ce qui représente un marché colossal. Toutes les personnes aiment personnaliser leurs biens pour se différencier des autres.

L'opportunité à saisir

Créer une application permettant de simplifier la personnalisation de son smartphone, utilisable sur Android et iOS pour toucher un maximum d'utilisateurs.

Identifier les parties prenantes

Développeurs : (Nous)

[Wallhaven](#) : Propriétaire de l'API

Utilisateurs : Tout public

6.2. Analyse de l'existant

Bibliothèques d'images en ligne

- [Wallhaven](#) : Bibliothèque de fonds d'écran alimenté par la communauté
- [Pixabay](#) : Bibliothèque d'images libres de droits (près de 2 millions de photos)
- [GettyImages](#) : Bibliothèque d'images

Application mobile de fond d'écran

- [Wallpaper](#) - Google LLC (plus de 100 millions de téléchargement)
- [ZEDGE](#) - Zedge (plus de 100 millions de téléchargement)
- [Wallpapers HD](#) - OGQ (plus de 100 millions de téléchargement)

6.3. Différentiation de la concurrence

L'application propose des images correspondant au format du smartphone de l'utilisateur.

La bibliothèque d'image est alimentée en continue par la communauté.

7. Analyse fonctionnelle

Cette partie de la documentation technique explique ce que le projet doit pouvoir faire, conformément au cahier des charges. Comme il y a deux façons d'utiliser l'application (lorsque l'utilisateur est connecté ou pas), les fonctionnalités seront par conséquent séparées en deux également.

7.1. Fonctionnalités disponibles

Lorsque l'utilisateur n'est pas connecté

- Connexion
- Inscription
- Recevoir les 24 dernières images
- Partager une image

Lorsque l'utilisateur est connecté

- Déconnexion
- Recherche d'image avec des paramètres
 - Tag
 - Couleurs
 - Résolution
 - Utilisateur
 - Catégories
- Recherche par tri
 - Date d'ajout
 - Nombre de vues
 - Nombre de likes
- Ajouter une image en favoris
- Retirer une image des favoris
- Enregistrer une image
- Proposition d'images similaires

7.2. Description des interfaces

Vue connexion

Cette vue permet à l'utilisateur de se connecter ou de se rediriger sur l'inscription si l'utilisateur n'a pas de compte.

Si le compte et le mot de passe concorde avec un compte présent dans la base de données.
L'utilisateur est connecté et redirigé sur la page principale.

La connexion à l'application permet de profiter de toutes les fonctionnalités dont les utilisateurs non connectés n'ont pas accès.

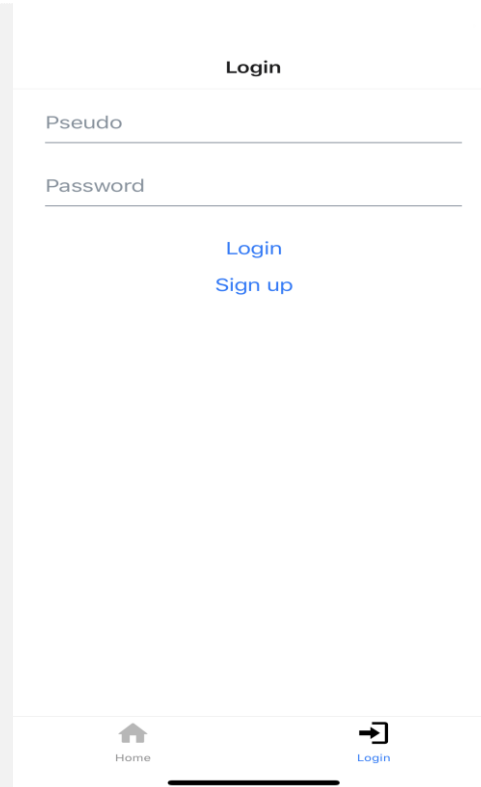
The screenshot shows a mobile application interface for the 'Login' screen. At the top, the title 'Login' is centered. Below it, there are two input fields: 'Pseudo' and 'Password'. Under the 'Password' field, there are two links: 'Login' and 'Sign up'. At the bottom, there is a navigation bar with two icons: a house icon labeled 'Home' and a right-pointing arrow icon labeled 'Login'.

Figure 3 - Vue connexion

Vue inscription

Cette vue permet l'inscription à l'application pour avoir un compte et profiter de ses avantages.

Si l'inscription est validée, l'utilisateur est connecté et redirigé sur la vue principale.

Cette page n'est accessible que depuis l'onglet de connexion.

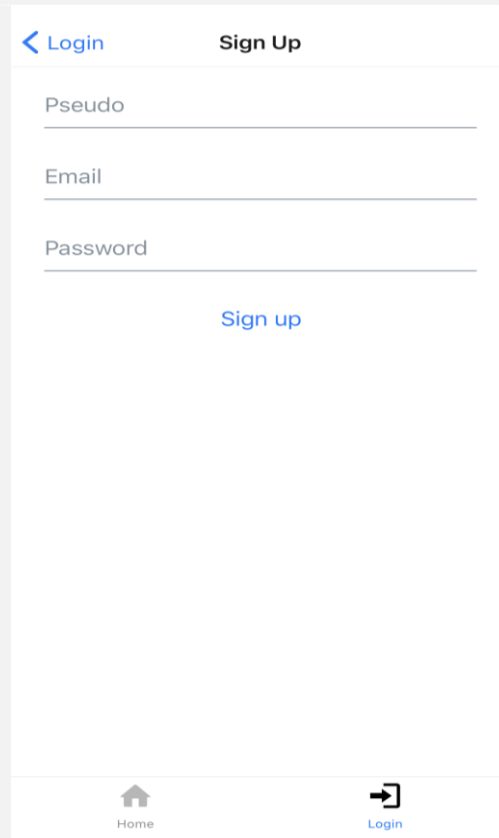
The screenshot shows a mobile application interface for the 'Sign Up' screen. At the top, there is a navigation bar with a back arrow icon and the text 'Login', and the title 'Sign Up' is centered. Below the title, there are three input fields: 'Pseudo', 'Email', and 'Password'. Under the 'Password' field, there is a link: 'Sign up'. At the bottom, there is a navigation bar with two icons: a house icon labeled 'Home' and a right-pointing arrow icon labeled 'Login'.

Figure 4 - Vue inscription

Vue déconnexion

Cette vue va déconnecter l'utilisateur, après confirmation, et le rediriger sur la page principale.

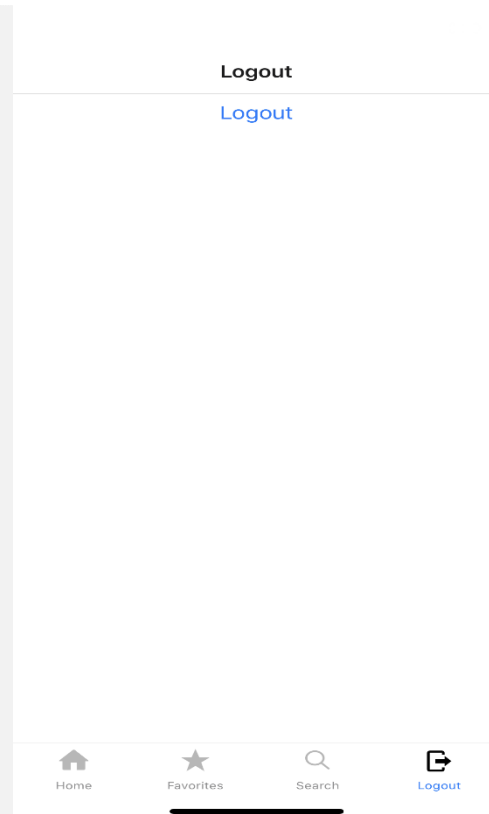


Figure 5 - Vue déconnexion

Vue favoris

Cette vue affiche toutes les images aimées par l'utilisateur.

Cette vue est accessible que par un utilisateur connecté.

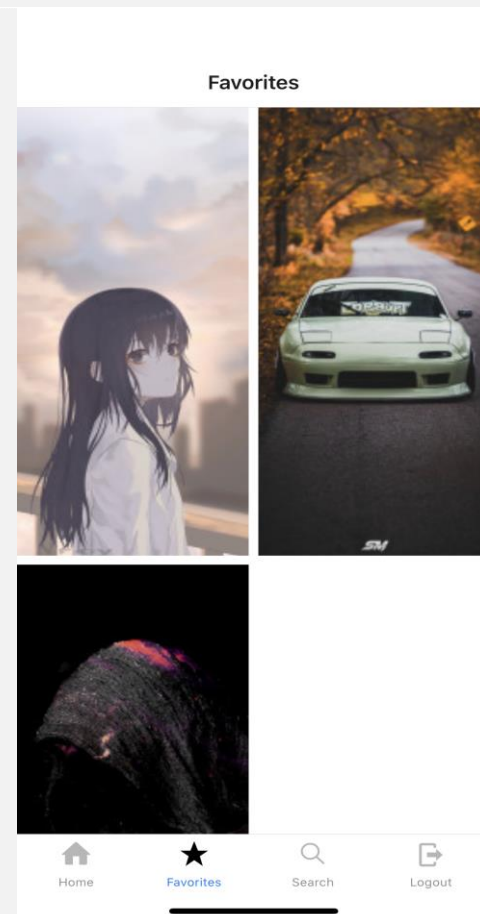


Figure 6 - Vue favoris

Vue recherche

Cette vue permet de choisir les différents paramètres afin d'effectuer une recherche de fonds d'écran.

Cette vue est accessible que par un utilisateur connecté.

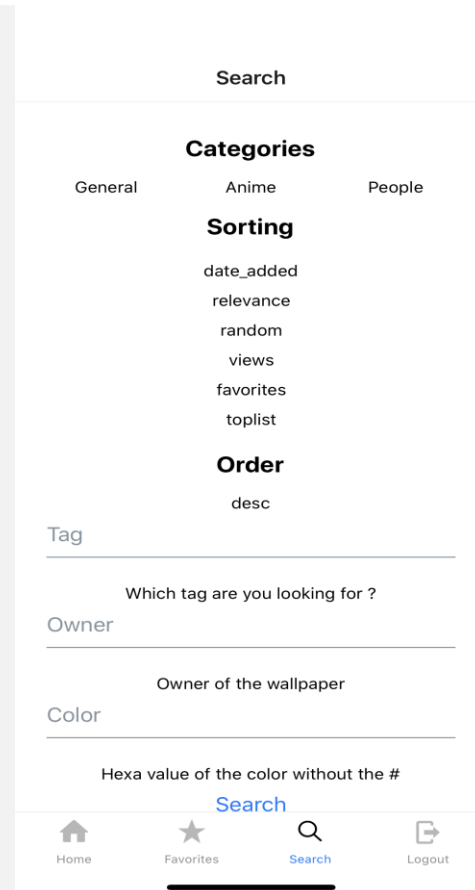


Figure 7 - Vue recherche

Vue résultats de la recherche

Cette vue affiche les résultats de la recherche, par page de 24 images chacune.

La barre inférieure affiche la page actuelle et le nombre de pages totales. Avec les deux flèches à gauche et à droite, l'utilisateur peut naviguer entre les différentes pages.

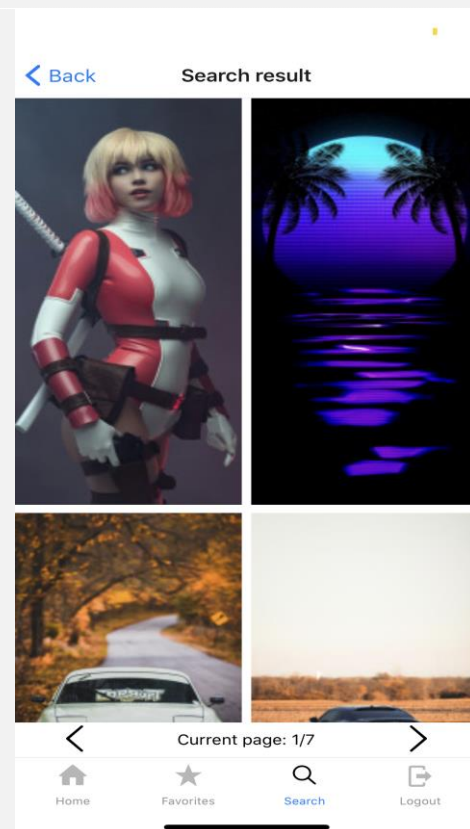


Figure 8 - Résultat de la recherche

Vue principale

Cette vue est le point d'entrée dans l'application pour l'utilisateur. Elle affiche les 24 dernières images qui correspondent aux ratio de l'écran.

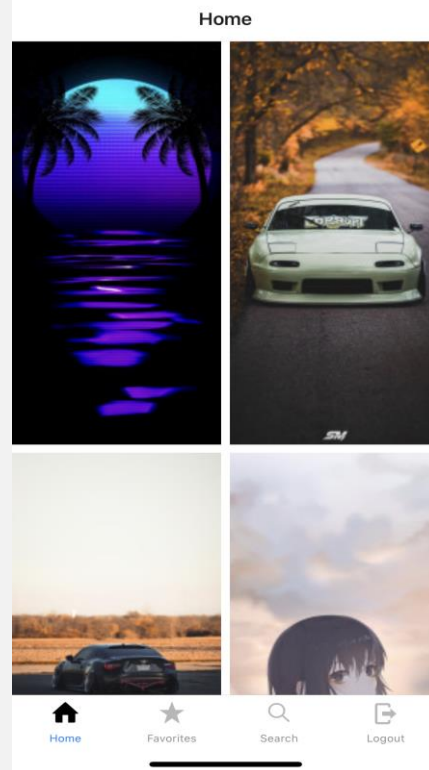


Figure 9 - Vue principale

Vue détaillée d'une image

Cette vue affiche une image en grand avec sa résolution et propose des images similaires. Il est possible de l'ouvrir dans le site web Wallhaven, de la partager ou encore de la mettre en favoris.

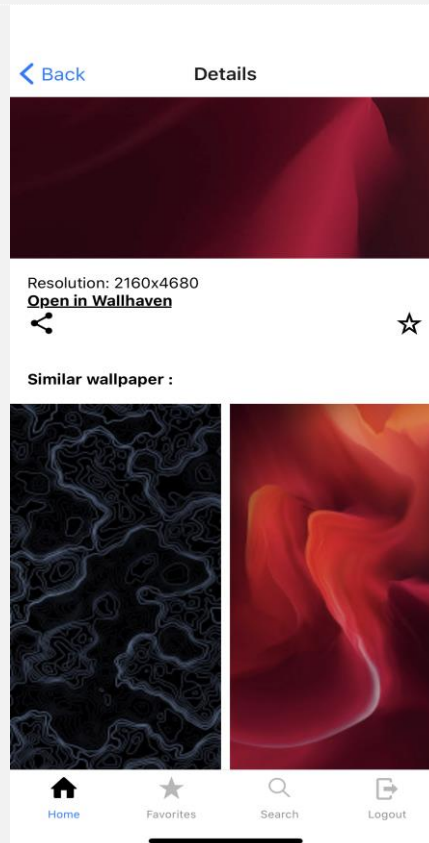


Figure 10 - Vue d'une image détaillée

8. Analyse organique

Le projet est découpé en 4 parties

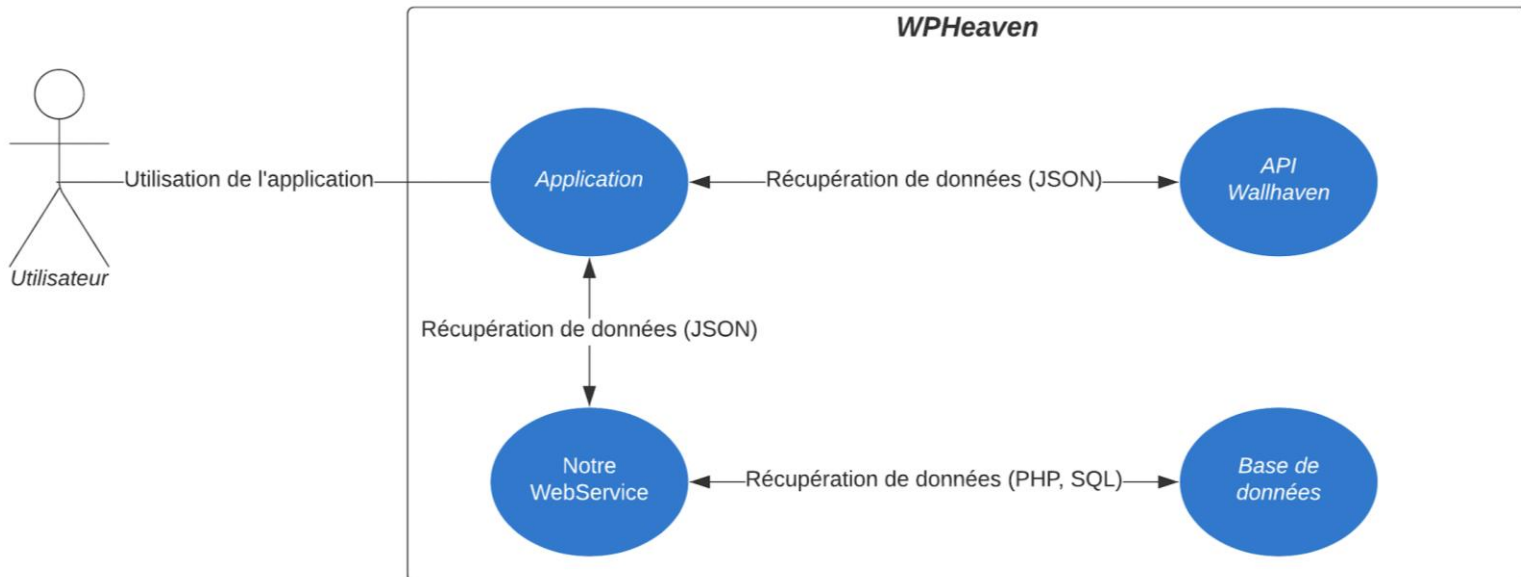


Figure 11 - Diagramme de la structure de l'application

- **L'application** va utiliser les différents composants d'informations et les traiter pour l'utilisateur
- **L'API Wallhaven** va nous fournir les fonds d'écran
- **La base de données** qui va stocker les utilisateurs et leurs favoris
- **Notre Web Service** qui va servir d'intermédiaire entre la base de données et l'application.

8.1. Architecture du projet

Application

- **assets** : contient différents icônes pour l'écran de démarrage de l'application ou l'icône sur le smartphone, par exemple
- **Components** : différentes pages de l'application
- **Navigation** : fichiers pour organiser la navigation en onglets (BottomTabNavigator) et pages qui s'empilent (StackNavigator)
- **node_modules** : modules installés avec npm pour ajouter des fonctionnalités à l'application
- **Store** : contient les fichiers pour la configuration et le stockage des données en local (Redux, State global)
- **WS** : contient les constantes et fonctions pour faire appel au WebService
- **App.js** : point d'entrée de l'application
- **package.json** : liste des modules à installer avec npm

WebService

- **php** : contient la connexion à la base de données ainsi que les fonctions PHP nécessaires
- **ws** : pages auxquelles l'application se connecte, pour récupérer les informations nécessaires en JSON

La base de données

Cette base de données est composée de 2 tables liées entre elle pour obtenir le lien entre une image mise en favoris et celui qui aimerait la sauvegarder.

La table User est composée de :

- Son id en Auto Increment
- Un pseudo
- Un email (utilisé pour la connexion)
- Un mot de passe qui sera encodé en sha256 avec le salt
- Un salt aléatoire encodé en sha256

La table Favoris est composée de :

- Son Id en Auto Increment
- L'id de l'image donné par l'API
- La date du favori pour trier les résultats
- Et l'id de l'utilisateur

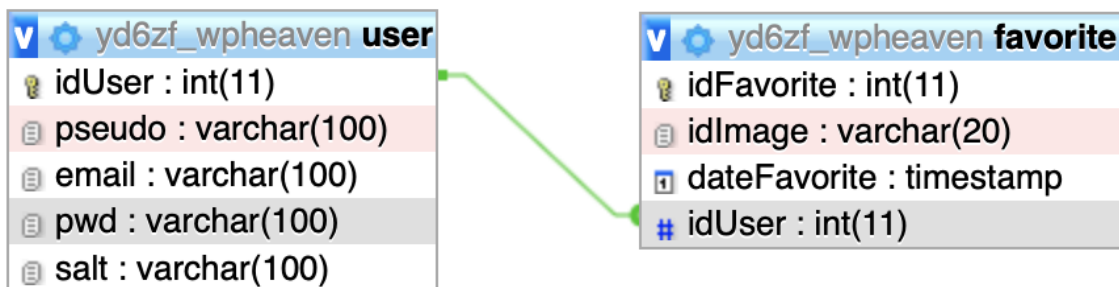


Figure 12 - Modèle de données réalisé

8.2. Description des fonctions

Application

Traitement en JS

- **pgcd** : prend deux nombres en paramètre et calcul le plus grand commun diviseur de ces deux nombres
- **nothingEmpty** : prend un objet en tant que paramètre, et vérifie qu'aucune de ses valeurs ne soit vide
- **trimValues** : prend un objet en tant que paramètre, et "trim" (enlève les espaces au début et à la fin) chaque valeur

Récupération d'informations depuis l'API Wallhaven

- **getLastWP** : récupère les derniers fonds d'écran qui correspondent au ratio d'écran de l'appareil de l'utilisateur
- **getSimilarWP** : prend l'id d'un fond d'écran comme paramètre et récupère ceux qui sont similaires et qui correspondent au ratio d'écran de l'appareil de l'utilisateur
- **searchWP** : prend tous les paramètres de recherche et de tri possible. Le tag, la couleur, le propriétaire, la catégorie, la page de recherche, l'ordre d'affichage et le tri. Si le paramètre n'est pas recherché, il faut attribuer la valeur null. Ces paramètres seront utilisés pour appeler setAPISearchLink qui va retourner le lien de l'appel à faire à l'API que cette fonction va utiliser pour récupérer tous les fonds d'écran.
- **setAPISearchLink** : Cette fonction permet de créer le lien de recherche d'images via l'API. Elle est utilisée uniquement dans searchWP et prend les même paramètres.

Récupération d'informations depuis notre WebService

- **login** : prend un objet qui contient les informations de connexion d'un utilisateur en tant que paramètre, et l'envoie au WebService pour vérifier que ces informations de connexion sont correctes
- **signUp** : prend un objet qui contient les informations d'inscription d'un utilisateur en tant que paramètre, l'envoie au WebService pour vérifier que ces informations d'inscriptions sont correctes, et l'inscrit le cas échéant
- **getFav** : cette fonction prend en paramètre l'id d'un utilisateur et récupère ses favoris
- **checkFav** : comme paramètre, cette fonction prend un objet contenant l'id d'un utilisateur ainsi que l'id d'un fond d'écran Wallhaven, et vérifie si le fond d'écran est dans les favoris de l'utilisateur
- **toggleFav** : comme paramètre, cette fonction prend un objet contenant l'id d'un utilisateur ainsi que l'id d'un fond d'écran Wallhaven, et ajoute le fond d'écran dans les favoris s'il n'y est pas déjà; si c'est le cas, elle le supprime des favoris

WebService

- **connecteur** : cette fonction sert à établir la connexion à la base de données. Elle ne reçoit pas de paramètre, mais utilise quatre constantes, qui sont « DB_HOST » pour l'hôte du serveur, « DB_NAME » pour le nom de la base de données, « DB_USER » pour l'utilisateur qui se connecte et « DB_PASSWORD » pour le mot de passe utilisé lors de la connexion. Cette fonction retourne une instance de la classe PDO
- **Fonctions SELECT**
 - **getSaltUser** : prend en paramètre le pseudo d'un utilisateur, et récupère le "salt" de cet utilisateur
 - **login** : prend en paramètre un tableau associatif qui contient les informations de connexion d'un utilisateur et vérifie ces informations de connexion
 - **checkEmailNotExist** : prend en paramètre un email et vérifie si le mail est déjà utilisé ou pas
 - **checkPseudoNotExist** : prend en paramètre un pseudo et vérifie si le pseudo est déjà utilisé ou pas
 - **checkFavExist** : prend en paramètre un tableau associatif contenant l'id d'un fond d'écran Wallhaven ainsi que l'id d'un utilisateur, et vérifie si un enregistrement avec les valeurs données existe dans la table "favorite"
 - **getFavUser** : prend en paramètre l'id d'un utilisateur et récupère ses favoris
- **Fonctions INSERT INTO**
 - **createUser** : prend en paramètre un tableau associatif qui contient les informations d'inscription d'un utilisateur, puis insère une nouvelle ligne dans la table "user"
 - **addFav** : prend en paramètre un tableau associatif contenant l'id d'un fond d'écran Wallhaven ainsi que l'id d'un utilisateur, et ajoute un enregistrement avec ces valeurs dans la table "favorite"
- **Fonction DELETE**
 - **delFav** : prend en paramètre un tableau associatif contenant l'id d'un fond d'écran Wallhaven ainsi que l'id d'un utilisateur, et supprime l'enregistrement avec ces valeurs de la table "favorite"

9. Les tests

Les tests sont découpés par fonctionnalités de l'application. Voici les résultats des tests à la fin du développement :

Inscription

N°Test	A tester	Commentaire	Résultat Attendu	Résultat	Bug
1	Inscription	Valider l'inscription sans avoir rempli tous les champs	Message d'erreur signalant qu'il faut remplir tous les champs	ok	
2	Inscription	Valider l'inscription avec un email d'un utilisateur connu	Message d'erreur signalant l'existence de son adresse email	ok	
3	Inscription	Valider l'inscription avec un pseudo d'un utilisateur connu	Message d'erreur signalant l'existence de ce pseudo	ok	
4	Inscription	Valider l'inscription avec un email et un pseudo unique et un mot de passe valide	Création de l'utilisateur	ok	
5	Inscription	Valider l'inscription avec un email et un pseudo unique et un mot de passe valide	Connexion de l'utilisateur créé	ok	

Tableau 2 - Tests de l'inscription

Connexion

10	Connexion	Valider la connexion sans avoir rempli tous les champs	Message d'erreur signalant qu'il faut remplir tous les champs	ok	
11	Connexion	Valider la connexion avec un identifiant ou mot de passe incorrect	Message d'erreur signalant que l'identifiant ou le mot de passe ne concorde pas	ok	

12	Connexion	Valider la connexion avec un identifiant et un mot de passe connu	L'utilisateur est connecté et redirigé sur la vue principale	ok	
----	-----------	---	--	----	--

Tableau 3 - Tests de la connexion

Déconnexion

20	Déconnexion	Clic sur déconnecter	Affichage d'un message confirmant la déconnexion	ok	
21	Déconnexion	Confirmation de la déconnexion	Déconnexion et redirection sur la page principale en étant déconnecté	ok	
22	Déconnexion	Annulation de la déconnexion	Ne déconnecte pas l'utilisateur	ok	

Tableau 4 - Tests de la déconnexion

Ouverture de l'application

30	Ouverture de l'application	Ouverture de l'application en étant déconnecté auparavant	Ouverture de la page principale en étant déconnecté	ok	
31	Ouverture de l'application	Ouverture de l'application en étant connecté auparavant	Ouverture sur la page principale en étant connecté	ok	

Tableau 5 - Tests de l'ouverture de l'application

Droit non connecté

40	Droit non connecté	Accès à la page inscription	Ouverture de l'inscription	ok	
41	Droit non connecté	Accès à la connexion	Ouverture de la connexion	ok	
42	Droit non connecté	Accès à la page principale	Affiche les 24 dernières images	ok	
43	Droit non connecté	Ouverture d'une image	Affichage de l'image mais pas de proposition d'images similaires	ok	
44	Droit non connecté	Accès à la déconnexion	Refus de l'accès	ok	
45	Droit non connecté	Accès à la page de recherche / tri	Refus de l'accès	ok	
46	Droit non connecté	Accès à la page favoris	Refus de l'accès	ok	

Tableau 6 - Tests des droits des utilisateurs non connectés

Droit connecté

50	Droit utilisateur connecté	Accès à la page inscription	Refus de l'accès	ok	
51	Droit utilisateur connecté	Accès à la connexion	Refus de l'accès	ok	

52	Droit utilisateur connecté	Accès à la page principale	Affiche les 24 dernières images	ok	
53	Droit utilisateur connecté	Ouverture d'une image	Affichage de l'image avec des propositions d'images similaires	ok	
54	Droit utilisateur connecté	Accès à la déconnexion	Ouverture de la déconnexion	ok	
55	Droit utilisateur connecté	Accès à la page de recherche / tri	Ouverture de la recherche	ok	
56	Droit utilisateur connecté	Accès à la page favoris	Ouverture des favoris	ok	

Tableau 7 - Tests des droits des utilisateurs connectés

Favoris

60	Favoris	Accès à la vue Favoris	Affichages des images mise en favoris	ok	
61	Favoris	Clic sur une image	Accès à l'affichage détaillé	ok	
62	Favoris	Ajouter en favoris une image	L'étoile des favoris est mise à jour et l'image de trouve dans l'onglet des favoris	ok	
63	Favoris	Suppression d'une image des favoris	L'étoile des favoris est mise à jour et l'image ne se trouve plus dans l'onglet des favoris	ok	

Tableau 8 - Tests des favoris

Affichage d'une image

70	Affichage détaillé d'une image	Clic sur une image	Affichage de l'image	ok	
71	Affichage détaillé d'une image	Clic sur une image	Affichage de sa résolution	ok	
72	Affichage détaillé d'une image	Clic sur une image	Affichage du lien de l'image sur Wallhaven	ok	
73	Affichage détaillé d'une image	Clic sur une image	Affichage de l'icône pour partager l'image	ok	
74	Affichage détaillé d'une image	Clic sur une image en étant connecté	Affichage de l'étoile des favoris avec son état	ok	
75	Affichage détaillé d'une image	Clic sur une image en étant connecté	Proposition d'images similaires	ok	

76	Affichage détaillé d'une image	Clic sur une image en étant déconnecté	Affichage Uniquement de l'image, sa résolution, le lien sur Wallhaven et l'icône de partage	ok	
----	--------------------------------	--	---	----	--

Tableau 9 - Tests lors de l'affichage d'une image

Recherche

80	Recherche	Sélection d'une catégorie	Change son état et l'affiche en gras ou en simple	ok	
81	Recherche	Sélection d'un type de tri	Affichage du tri en gras	ok	
82	Recherche	Sélection de l'ordre	Change entre asc ou desc	ok	
83	Recherche	Recherche d'un tag	Images correspondant au tag et à l'écran	ok	
84	Recherche	Recherche par pseudo	Images correspondant au pseudo et à l'écran	ok	
85	Recherche	Recherche par couleur	Images correspondant à la couleur et à l'écran	ok	
86	Recherche	Recherche multiple (plusieurs paramètres)	Images correspondant aux paramètres	ok	

Tableau 10 - Tests de la recherche

10. Conclusion

10.1. Bilans personnels

Florent :

Ayant déjà eu l'occasion de travailler avec le framework React Native, ce projet de semestre m'a permis d'à nouveau m'entraîner à utiliser ce framework afin de produire une application fonctionnelle sur plusieurs plateformes (Android et iOS). J'ai pu aider Clément a débuté avec ce framework et j'ai apprécié ce rôle. Ce projet m'a confirmé que React Native est un framework que j'apprécie et je suis sûr que je vais continuer à l'utiliser dans le futur si j'ai besoin de développer une application mobile.

Clément :

Durant ce travail j'ai appris et utilisé un nouveau framework avec un langage dont je ne suis pas encore bien à l'aise. Dans un premier temps je me suis amélioré en JavaScript ce qui m'a permis d'avoir de meilleures bases pour la suite.

Après cette étape franchie, je me suis concentré sur le framework React Native avec l'aide de Florent qui était plus expérimenté dans ce domaine. Aujourd'hui je ne me sentirai pas encore prêt à utiliser ce framework dans un travail plus conséquent mais j'ai pu apprécier de sortir de ma zone de confort en réalisant un projet sans avoir toutes les connaissances.

10.2. Bilan de la méthodologie

Pour ce travail, nous avons découpé toutes les tâches dans une logique de sprints. Pour ce faire, nous avons utilisé l'onglet "Projects" de notre dépôt GitHub, qui nous a permis de garder une trace des tâches effectuées, en cours et en attente.

Nous n'avons pas vraiment eu d'éléments imprévus ou de points bloquants.

Lors de la mise en place du télétravail, nous avons décidé de continuer à venir en classe pour les périodes prévues au travail de semestre. Nous avons remarqué que nous étions plus productifs et qu'on collaborait mieux en étant en physique comparé au distanciel.

10.3. Conclusion sur l'application

L'application actuelle est dans un état final proche de la publication si on prend en référence le cahier des charges. Tous les tests imaginés sont passés.

Il manque malheureusement une fonctionnalité qui est l'enregistrement d'images via l'application. L'implémentation de cette fonctionnalité mériterait plus de temps pour analyser les solutions de réalisations possibles. En ce qui concerne la fonctionnalité, elle est partiellement accessible car nous mettons à disposition un lien de l'image vers le site Wallhaven et l'utilisateur peut l'enregistrer par ce biais.

Avant de pouvoir publier l'application, il manque plusieurs éléments :

- Une image pour l'icône de l'application
- Une image utilisée à l'ouverture de l'application, lorsque cette dernière charge
- Un compte développeur chez Apple pour publier sur l'App Store, et un compte développeur chez Google afin de publier sur le Play Store
 - Pour cette dernière étape, il faut enregistrer un compte pour chaque plateforme, qui est payant.
- Le dernier critère dépend des plateformes mentionnées ci-dessus, c'est le fait que les plateformes acceptent de publier l'application sur leurs stores respectifs

Dans l'ensemble, nous sommes très contents du déroulement du projet ainsi que de son résultat. Ce projet nous a tous les deux amenés des choses différentes et renforcé dans le travail de groupe.

11. Références

Liens

Wallhaven site : <https://wallhaven.cc/>

Wallhaven API : <https://wallhaven.cc/help/api>

Pixabay : <https://pixabay.com/fr/>

GettyImages : <https://www.gettyimages.ch>

Wallpaper (Google) :

<https://play.google.com/store/apps/details?id=com.google.android.apps.wallpaper>

ZEDGE : <https://www.zedge.net/>

Wallpapers HD (OGQ) : <https://www.ogq.me/policy/about>

Figure 1 - Planning initial.....	5
Figure 2 - Modèle de données prévisionnel.....	6
Figure 3 - Vue connexion	9
Figure 4 - Vue inscription	9
Figure 5 - Vue déconnexion	10
Figure 6 - Vue favoris	10
Figure 7 - Vue recherche	11
Figure 8 - Résultat de la recherche	11
Figure 9 - Vue principale	12
Figure 10 - Vue d'une image détaillée	12
Figure 11 - Diagramme des interfaces	13
Figure 12 - Modèle de données réalisé	14
Tableau 1 - Matériels et logiciels	5
Tableau 2 - Tests de l'inscription	17
Tableau 3 - Tests de la connexion	18
Tableau 4 - Tests de la déconnexion	18
Tableau 5 - Tests de l'ouverture de l'application.....	18
Tableau 6 - Tests des droits des utilisateurs non connectés	18
Tableau 7 - Tests des droits des utilisateurs connectés	19
Tableau 8 - Tests des favoris	19
Tableau 9 - Tests lors de l'affichage d'une image.....	20
Tableau 10 - Tests de la recherche.....	20

12. Annexes

- Manuel d'installation
- Journal de bord
- Codes sources