

Table des matières

[1 Glossaire 4](#_Toc73349299)

[2 Analyse préliminaire 5](#_Toc73349300)

[2.1 Introduction 5](#_Toc73349301)

[2.1.1 Titre 5](#_Toc73349302)

[2.1.2 Description 5](#_Toc73349303)

[2.1.3 Contexte 5](#_Toc73349304)

[2.2 Objectifs 6](#_Toc73349305)

[*2.3* Planification Initiale et détaillée 8](#_Toc73349306)

[3 Analyse / conception 9](#_Toc73349307)

[3.1 Concept 9](#_Toc73349308)

[3.1.1 Méthodologie de gestion de projet : En cascade 9](#_Toc73349309)

[3.1.2 Maquettes grahiques / Liason des pages : 10](#_Toc73349310)

[17](#_Toc73349311)

[3.1.3 Modèle Conceptuel des Données 19](#_Toc73349312)

[3.1.4 Besoins des utilisateurs 19](#_Toc73349313)

[3.2 Stratégie de test 20](#_Toc73349314)

[3.3 Risques techniques 20](#_Toc73349315)

[3.4 Planification 21](#_Toc73349316)

[3.5 Dossier de conception 22](#_Toc73349317)

[3.5.1 Matériel Hardware / Système d’exploitation 22](#_Toc73349318)

[3.5.2 Logiciels 22](#_Toc73349319)

[3.5.3 Suivi de projet 23](#_Toc73349320)

[3.5.4 Base de données : Firebase 23](#_Toc73349321)

[3.5.5 CocoaPods – Gestionnaire de dépendances 24](#_Toc73349322)

[3.5.6 Architecture du programme : Modèle-Vue-Contrôleur 25](#_Toc73349323)

[3.5.7 CocoaPods 26](#_Toc73349324)

[3.5.8 Cas d’utilisation 27](#_Toc73349325)

[3.5.9 Modèle Logique de Données 28](#_Toc73349326)

[4 Réalisation 29](#_Toc73349327)

[4.1 Liste des fichiers 29](#_Toc73349328)

[4.2 Firebase : Authentification 29](#_Toc73349329)

[4.3 Firebase : Firestore 29](#_Toc73349330)

[4.4 Créations des cellules d’article 29](#_Toc73349331)

[4.5 JSON 29](#_Toc73349332)

[4.6 Description des tests effectués 30](#_Toc73349333)

[4.7 Erreurs restantes 33](#_Toc73349334)

[4.8 Liste des documents fournis 33](#_Toc73349335)

[5 Conclusions 33](#_Toc73349336)

[5.1 Bilan de la planification 33](#_Toc73349337)

[5.2 Bilan des fonctionnalités demandées 33](#_Toc73349338)

[5.3 Bilan personnel 33](#_Toc73349339)

[6 Annexes 33](#_Toc73349340)

[6.1 Résumé du rapport du TPI / version succincte de la documentation 33](#_Toc73349341)

[6.2 Webographie 33](#_Toc73349342)

[6.3 Planification Initiale 33](#_Toc73349343)

[6.4 Journal de travail 33](#_Toc73349344)

[6.5 Manuel d’installation 34](#_Toc73349345)

[6.6 Manuel d’Utilisation 34](#_Toc73349346)

[6.7 Archives du projet 34](#_Toc73349347)

[Figure 1 Modèle en cascade - Page wikipédia 9](#_Toc73340677)

[Figure 2 Maquette : Page de connexion 10](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340678)

[Figure 3 Maquette : Page de création de compte 11](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340679)

[Figure 4 Maquette : Page principale de l'application 12](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340680)

[Figure 5 Maquette : Page principale de l'application (Bouton sélection activé) 13](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340681)

[Figure 6 Maquette : Page du profil utilisateur 14](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340682)

[Figure 7 Maquette : Page ajout d'un article 15](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340683)

[Figure 8 Maquette : Page détail d'un article 16](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340684)

[Figure 9 Maquette : Page modification d'un article 17](#_Toc73340685)

[Figure 10 Maquette : Navigation de l'application 18](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340686)

[Figure 11 Maquette : Modèle conceptuel des données 19](#_Toc73340687)

[Figure 12 Graphique : Bilan de la planification initiale 21](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340688)

[Figure 13 Architecture - Application : Modèle 24](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340689)

[Figure 14 Architecture – Application : Vue - Dossiers 24](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340690)

[Figure 15 Architecture - Application : Vues de l’application 24](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340691)

[Figure 16 Architecture - Application : Contrôleurs - Dossier 25](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340692)

[Figure 17 Maquette : Cas d'utilisation de l'application 26](file:///C:\Users\kilboukhlifa\Desktop\TPI_inventaireHabits_BoukhlifaK\1.Documentation\2.Rapport\R-tpi-boukhlifa-Rapport-inventaireHabits.docx#_Toc73340693)

[Figure 18 Maquette : Modèle logique des données 27](#_Toc73340694)

# Glossaire

JSON

# Analyse préliminaire

## Introduction

### Titre

Inventaire Habits

### Description

Cette application mobile sera réalisée en Swift, un langage de programmation pour les mobiles sous IOS.

ll s'agit d’une application qui fonctionnera sur un iPhone, qui permettra à un utilisateur d'inventorier et de gérer sa garde-robe d'habits/chaussures.

Les fonctionnalités attendues sont :

* Saisie des informations concernant un article, telles que, par exemples, le prix d’achat, date d'achat, taille, lieu d’achat, condition (usure), nom de l’article etc.
* Mise à jour des informations et suppression d’articles.
* Recherches d’articles à l'aide de critères, avec un résultat qui donne une liste dans laquelle on peut sélectionner un article en particulier et obtenir ainsi les informations le concernant.
* Production d'une liste d'habits selon critère qui peut être envoyée par la suite à l'aide de l'email ou sms à un magasin de ventes d'habits, ou à la Chaîne du Bonheur

L’application servira de gestionnaires pour l’utilisateur, il aura la possibilité d’ajouter de nouveaux articles, en supprimer et en modifier, cela s’affichera sous la forme d’une collection que l’ont pourra parcourir à l’aide de différentes catégories (T-shirt, Hoodie) etc.

### Contexte

Ce projet est réalisé dans le cadre de mon TPI afin de valider mon CFC, il me permettra de démontrer mes capacités acquises lors de mon apprentissage.

La thématique du projet a été choisie par moi-même avec la validation de mon chef de projet Gruaz Gilbert ayant pris en compte ma passion autours de l’univers de la mode, des vetêments et des chaussures.

## Objectifs

Le candidat est responsable de livrer, à son chef de projet et aux deux experts, les éléments de suivi de projet suivants :

* Une planification initiale (jour 1) et une planification aussi détaillée que possible (jour 3)
* Un tableau des besoins utilisateurs (clients et/ou vendeurs)
* Un rapport de projet
* Un journal de travail
* Les instructions pour la mise en œuvre de l'application, ou à une simulation de celle-ci

En plus de cela, l’application mobile sera évalué sur les 7 points spécifiques suivants (Point A14 à A20):

1. **Production d'un modèle de données pertinent pour les informations concernant les habits/Article :**

* Création d’une base de donnée
* Implémenter les données suivantes dans la base de données :
  + - * Image de *l’article\**
      * Nom \*
      * Taille \*
      * Prix \*
      * Lieu d’achat \*
      * Condition \*
      * Date d’achat

1. **Mise en œuvre d'une interface pour la saisie de ces informations :**

* Page ajout d’un article :
  + - * Toutes les informations concernant l’article
      * Liaison avec la base de données (Ecriture DB)
      * Bouton de validation
      * Bouton retour

1. **Mise en œuvre d'une interface de modifications d'informations :**

* Page modification d’un article :
  + - * Toutes les informations concernant l’article
      * Liaison avec la base de données (Lecture et écriture DB)
      * Bouton de validation
      * Bouton de suppression de l’article
      * Bouton retour

1. **Mise en œuvre d'une interface de suppressions d'articles :**

* Bouton disponible sur la page de modification, permettant la suppression d’un article.

1. **Mise en œuvre d'une interface de recherches d'habits selon critères :**

* Page principale de l’application :
  + - * Affichage des articles ajoutés sous forme de collection
      * Bouton ou Menu déroulant pour sélectionner la catégorie d’articles souhaiter : « T-shirt, Pull, Pantalon, Chaussures etc. »

1. **Mise en œuvre d'une liste d'habits selon critères, qui pourra être communiquée par email, sms ou autres :**

* Système permettant de sélectionner plusieurs articles pour les partager par e-mail, What’s App, SMS etc.

1. **Les commentaires dans le code sont clairement rédigés avant de coder, de façon à permettre une réécriture dans un autre langage comme pour Android par exemple.**

## Planification Initiale et détaillée

La planification initiale a été réalisé le premier jour du projet et elle permet de répartir les tâches à effectuer lors des 5 semaines du projet TPI.

Pendant ce projet chaque journée est divisée en quart d’heure et nous avons en tout pendant ce projet 3 journées entière par semaine et 1 demi-journée.

La planification initiale est consultable en annexes.

**Annexe** : [Planification](#_Planification_Initiale)

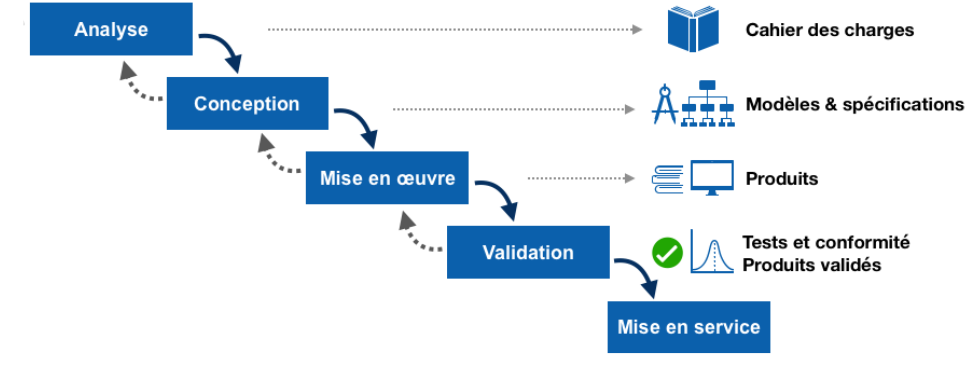
# Analyse / conception

## Concept

### Méthodologie de gestion de projet : En cascade

Lors de la réalisation de ce projet, je vais utiliser la méthode de travail dit « **en cascade** ».

Je vais donc organiser mon projet de manière séquentielle, lorsque je finirais une étape, je passerais donc à la suivante en vérifiant la précédente, c’est une méthode très adaptée au développement d’applications.

Figure 1 [Modèle en cascade - Page wikipédia](https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade)

**Analyse :**

L’analyse du projet comprend, la réflexion concernant les besoins de l’utilisateur selon le cahier des charges, ainsi que la réalisation de la planification du projet.

**Conception :**

La conception du projet comprend la réalisation des maquettes graphiques de l’application, du tableau des besoins de l’utilisateur, du MCD et du MLD de la base de données.

Ensuite, la description en détail des fonctionnalités qui devront apparaître sur les différentes pages de l’application.

**Mise en oeuvre :**

La mise en œuvre comprend le développement de l’application selon le cahier des charges ainsi que la retranscription du développement de l’application sur le rapport du projet.

**Validation :**

La validation du projet consistera à effectuer les tests attestant que les besoins de l’utilisateur sont respectés et satisfaits.

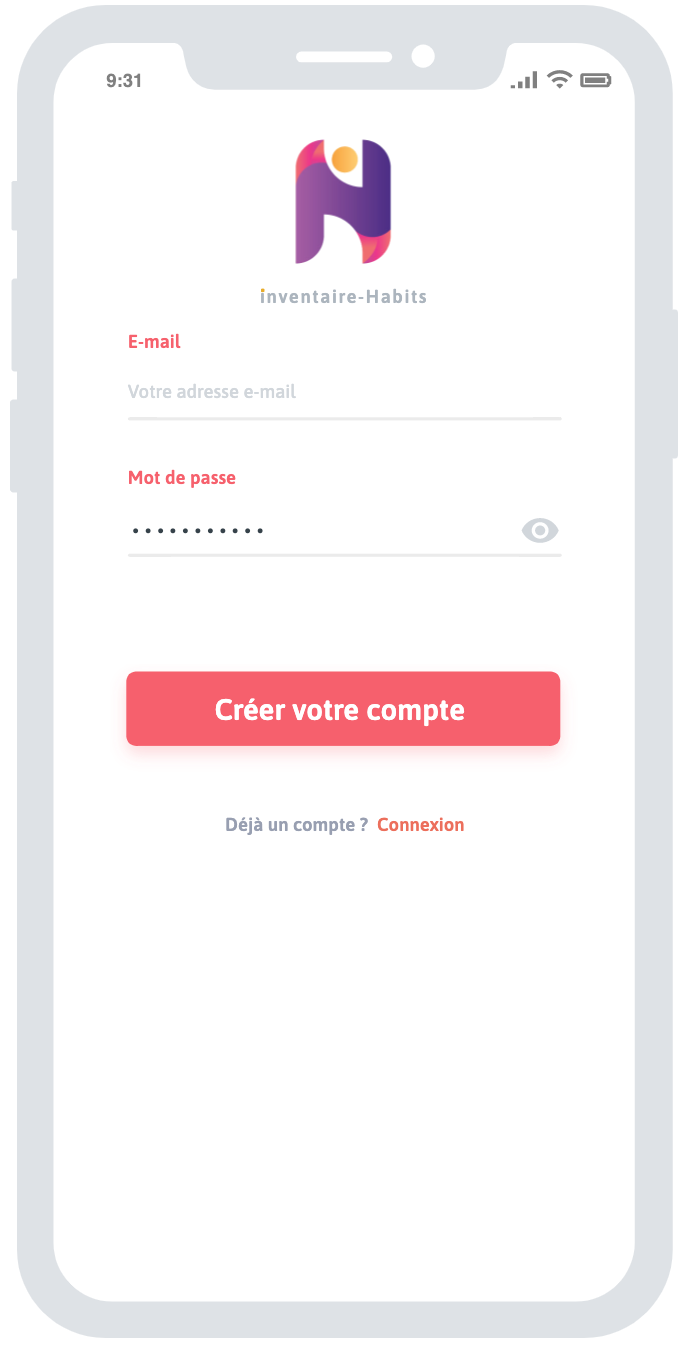
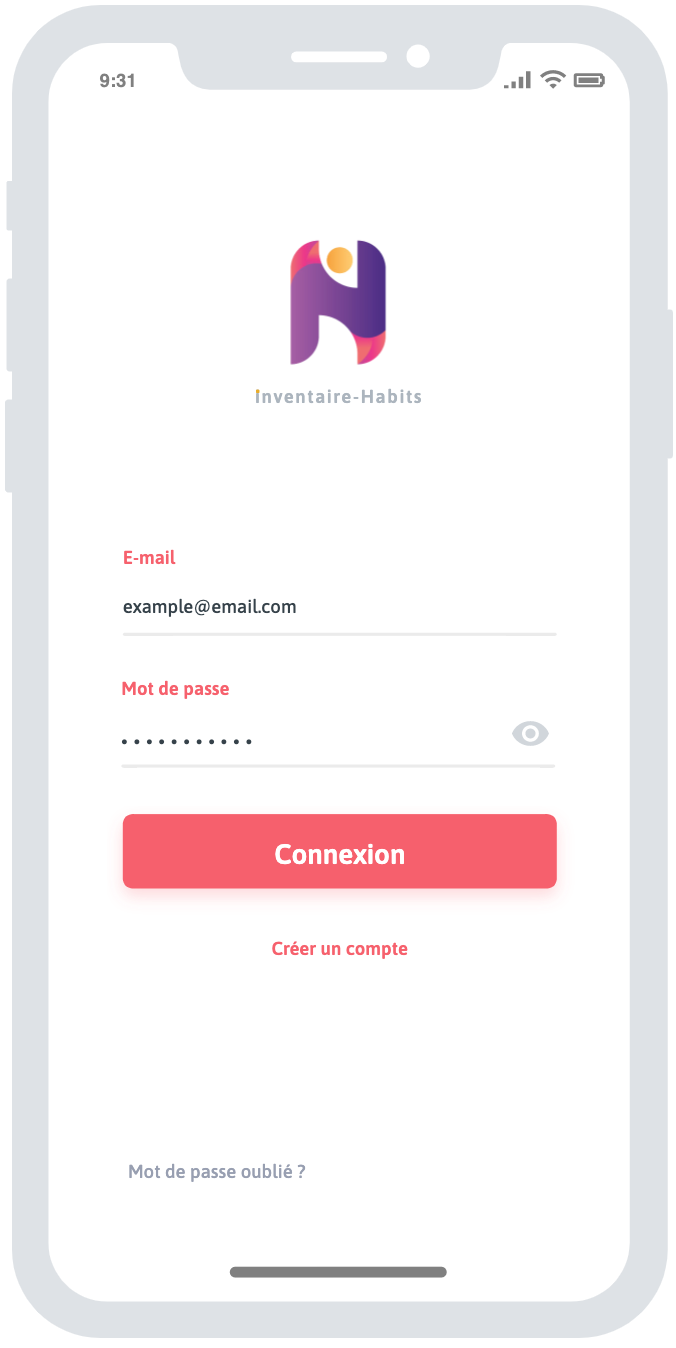
Mais en plus que l’application fonctionne correctement et comprend toutes les technologies décrites dans le cahier des charges.

**Mise en service :**

Finalement, la mise en service du projet dans mon contexte, consistera à la remise du projet final à mon chef de projet ainsi qu’aux experts.

Et comprendra également la présentation du projet.

### Maquettes grahiques / Liason des pages :



Bouton pour réinitialiser un mot de passe, si l’utilisateur l’a oublié

Bouton pour la création d’un compte

Bouton pour la création d’un compte

Champ pour l’adresse mail

Champ pour le mot de passe du compte

Figure 2 Maquette : Page de connexion

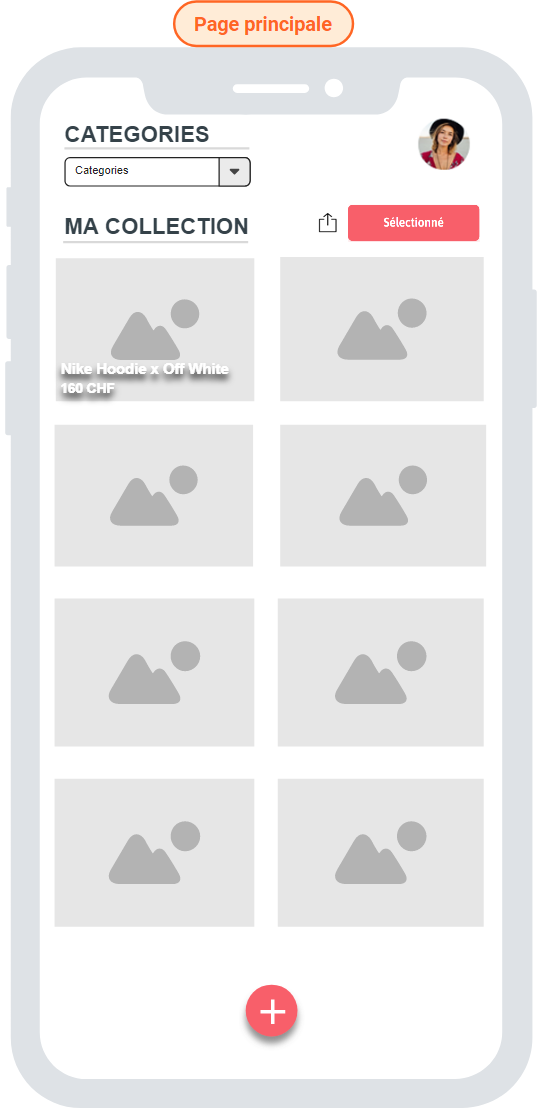
Champ pour l’adresse mail

Champ pour le mot de passe du compte

Bouton pour retourner à la page connexion (Si l’utilisateur possède déjà un compte)

Bouton pour la création d’un compte

Figure 3 Maquette : Page de création de compte



Menu déroulant permettant la sélection de la catégorie d’article que l’on souhaite afficher.

Ex (T-shirt, Pull, Pantalon etc..)

Bouton pour partager les articles sélectionnés.

Affichage d’un article, Image en fond avec le nom de l’article ainsi que son prix.

C’est un élément cliquable qui permet d’accéder à la page «  Détail d’article »

Bouton pour accéder à la page ajout d’un article.

Bouton pour accéder à la page « Profil »

Bouton qui active la possibilité de sélectionner des articles.

Figure 4 Maquette : Page principale de l'application

Affichage d’un article dans le « Feed »

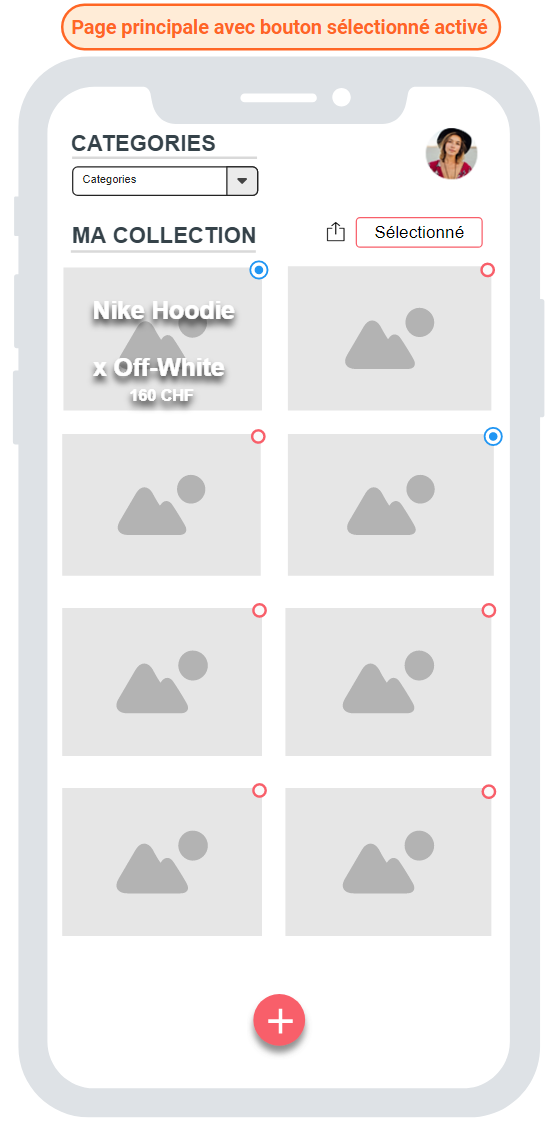


Figure 5 Maquette : Page principale de l'application (Bouton sélection activé)

Bouton radio sélectionné = Article sélectionné

Bouton « sélectionné » activé, les boutons radio sont apparus sur les articles.

Bouton radio non sélectionné = Article non sélectionné

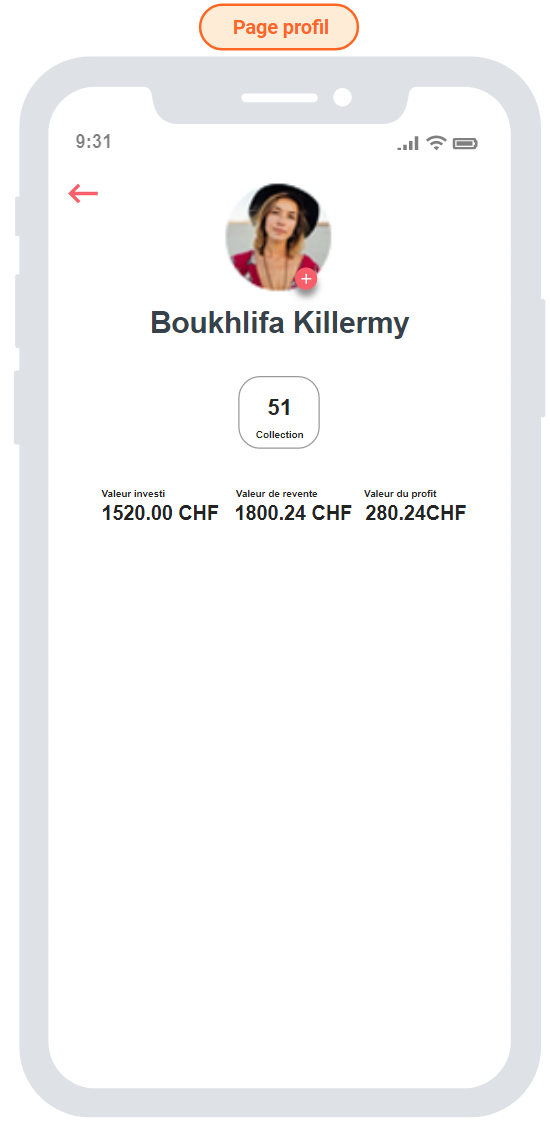


Figure 6 Maquette : Page du profil utilisateur

Différence entre le prix d’achats et le prix de revente des articles cumulés

Prix d’achats de tous les articles cumulés

Prix de reventes de tous les articles cumulés

Nombre d’articles que l’utilisateur collectionne.

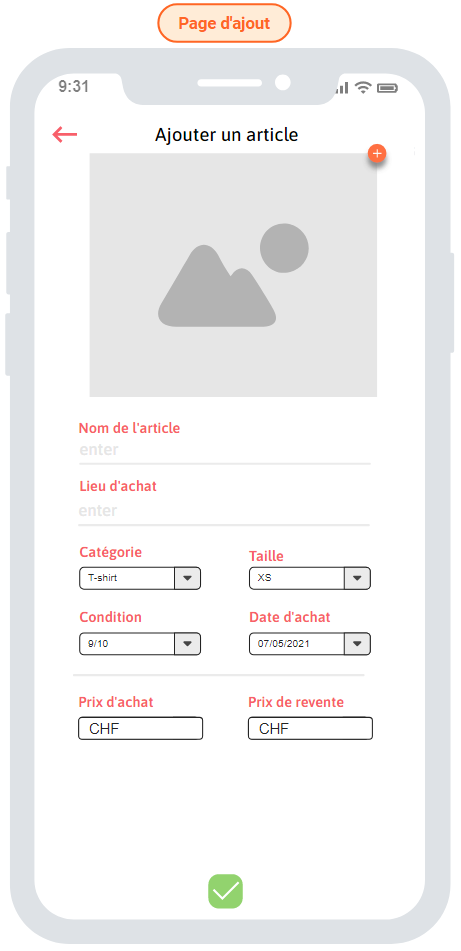


Figure 7 Maquette : Page ajout d'un article

Champ de texte pour le **nom de l’article** et son **lieu d’achat**

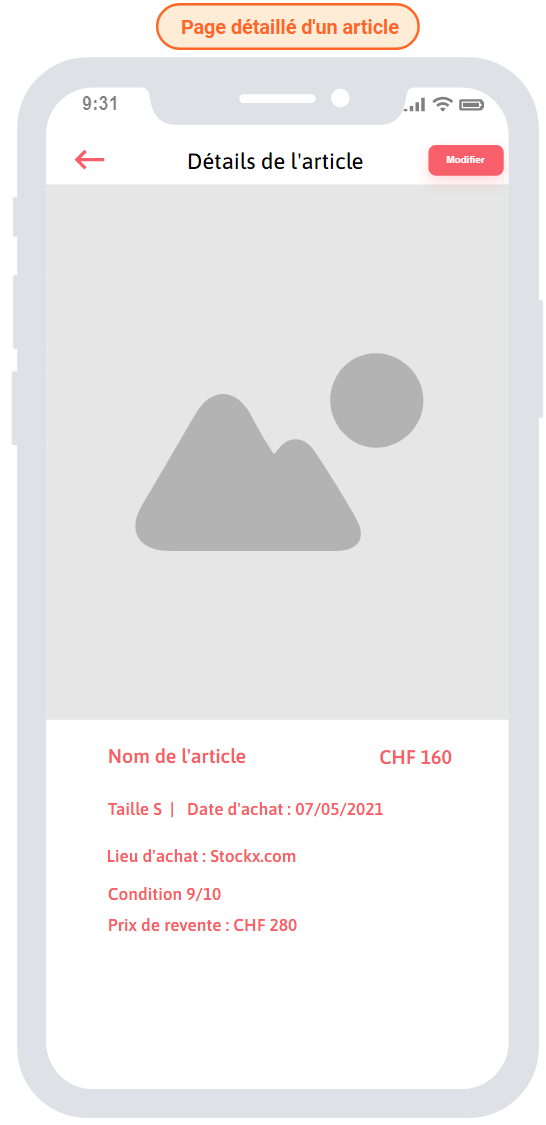
Bouton pour valider les informations saisies.

TextBox permettant la saisie du **prix d’achat** ainsi que le **prix de revente** d’un article.

Menu déroulant permettant de choisir la **condition** de l’article et un DateTimePicker pour la **date d’achat**.

Menus déroulant permettant de choisir la **catégorie** de l’article et sa **taille**.

Bouton permettant l’ajout d’une **image** dans la DB pour un article.



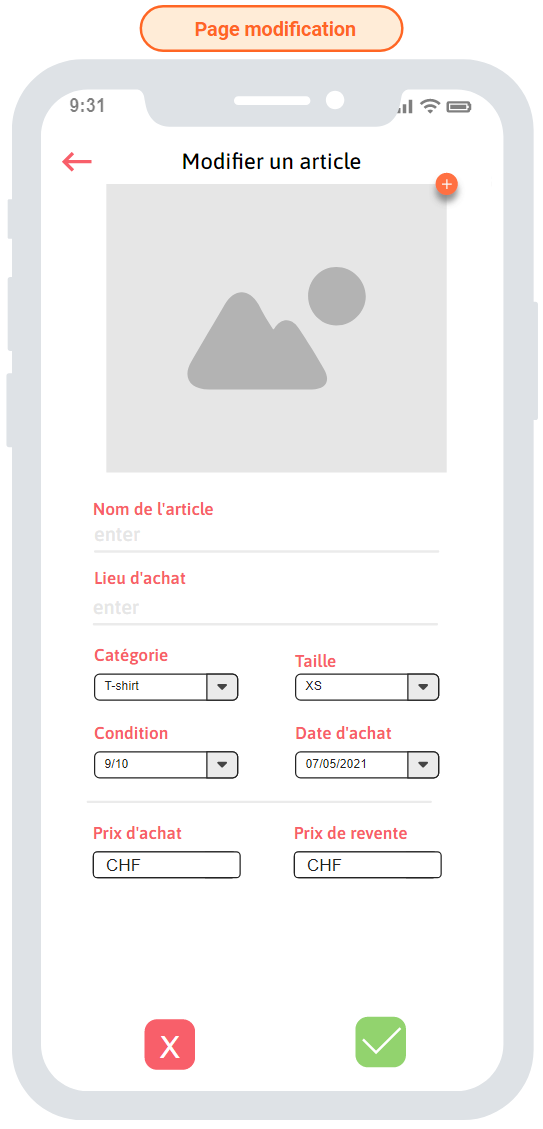
Affichage des informations de l’article

Affichage de l’image de l’article

Bouton retour à la page précédente

Bouton pour modifier les informations d’un article

Figure 8 Maquette : Page détail d'un article

****

Bouton permettant la suppression de l’article

Modification des informations de l’article

### 

Bouton permettant la validation des modifications des informations

Figure 9 Maquette : Page modification d'un article

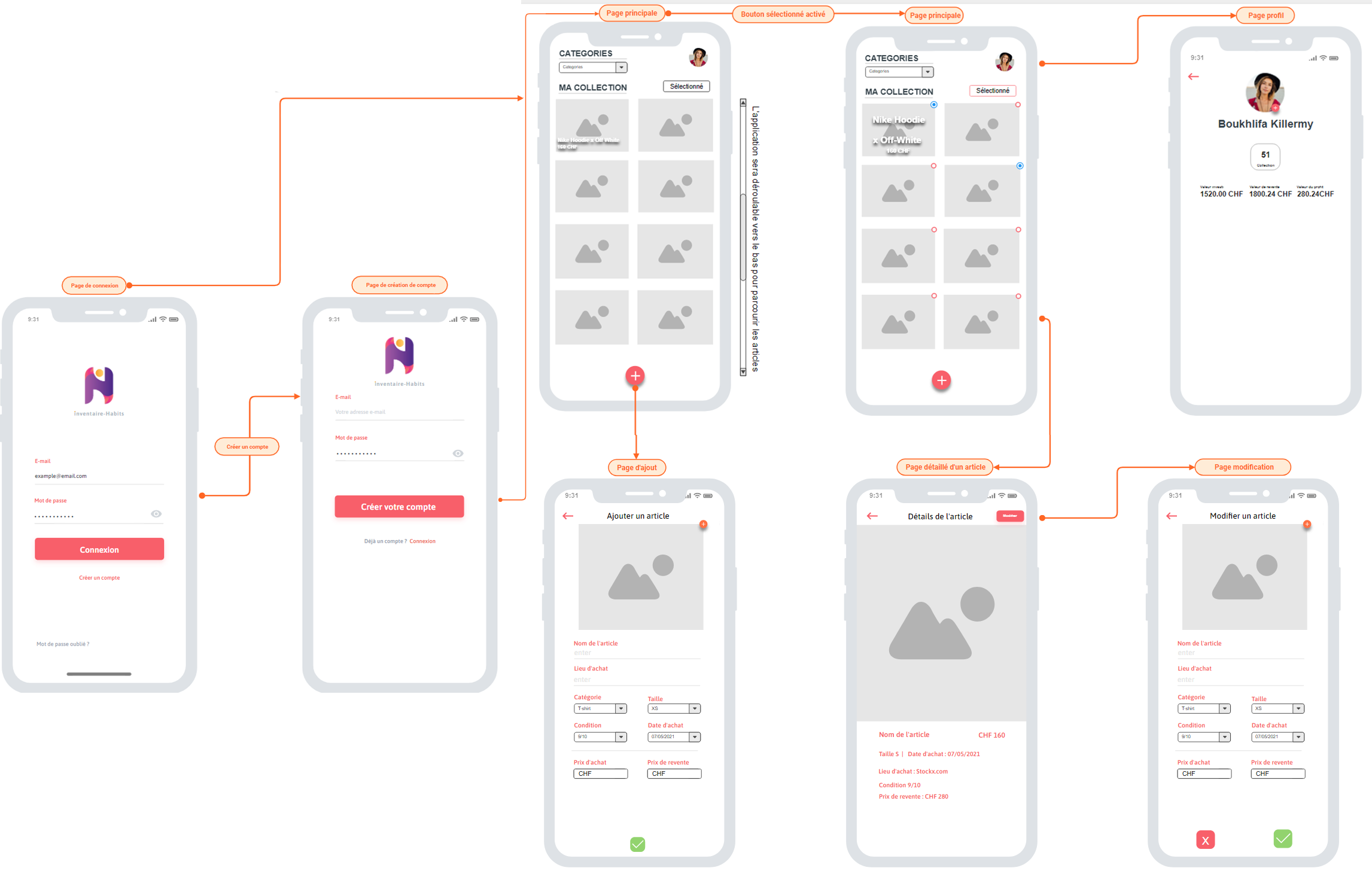
****

Figure 10 Maquette : Navigation de l'application

### Modèle Conceptuel des Données

**MCD**

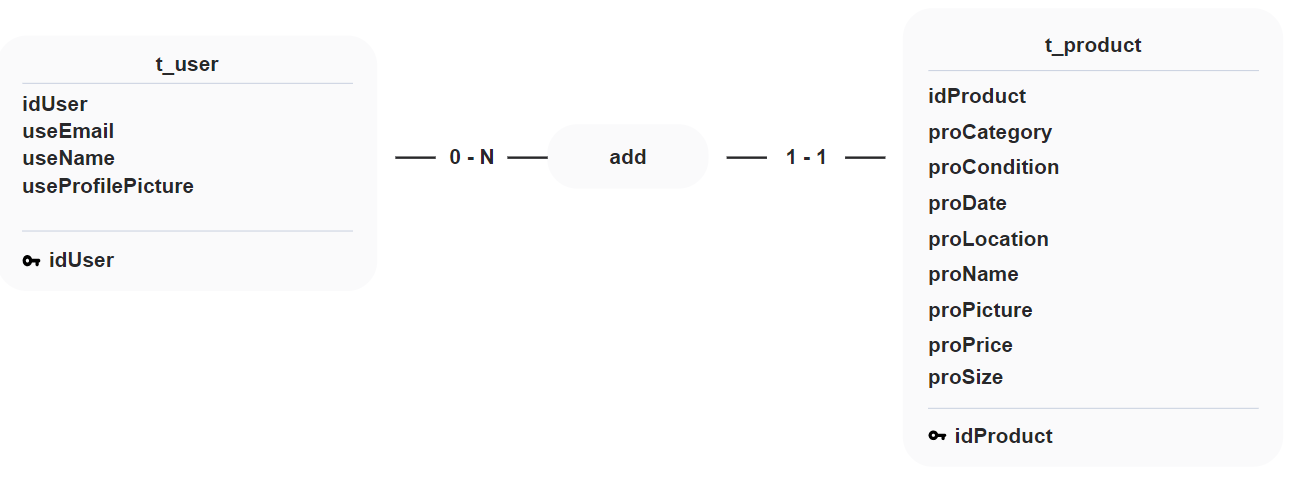


Figure 11 Maquette : Modèle conceptuel des données

### Besoins des utilisateurs

|  |  |
| --- | --- |
| ***Besoins des utilisateurs*** | **Page répondant à ces besoins.** |
| Enregistrement de ses informations. | Page création d’un compte/Base de donnée Firestore |
| Ajouts d’articles dans la collection de l’utilisateur | Page principale « Gallerie » pour parcourir et trier tous les articles de l’utilisateurs à l’aide du bouton se trouvant sur la page principale.  Page détail d’un article, pour consulter en détail les informations concernant un article. |
| Modifier les informations d’un article | Page « Modification » permettant de modifier les informations d’un article, changer le nom, description, l’image etc.. d’un article précédemment ajouter. |
| Supprimer un article de sa collection | Page « Modification » bouton permettatn de supprimer un article. |
| Obtenir un bilan de la collection de l’utilisateur | Page « Profil » permettant à l’utilisateur d’obtenir des informations concernant sa collection complète. |
| Partager les articles ce trouvant dans la collection de l’utilisateur | Page « Principale – Galerie » Bouton sélection pour sélectionner plusieurs articles et partager leurs informations via E-mail, SMS, What’s App etc… |

*Le concept complet avec toutes ses annexes:*

*Par exemple :*

* *Bases de données: interfaces graphiques, modèle conceptuel.*
* *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle…*

## Stratégie de test

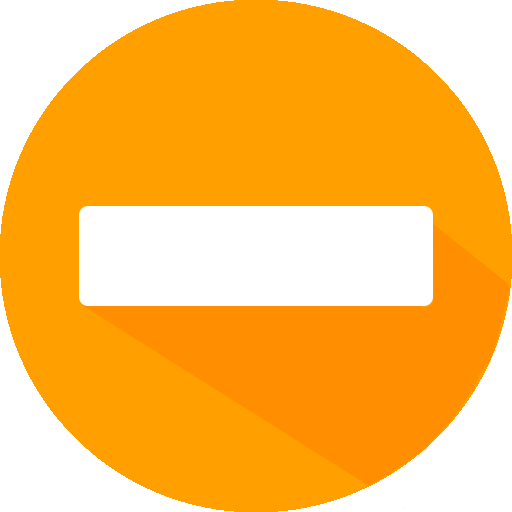
Les tests me permettront de savoir si mon projet se déroule correctement et de détecter d'éventuels problèmes survenus lors de la réalisation du projet.

J’effectuerais vers la fin de mon projet des tests afin de corriger de potentiels problèmes.

Les tests seront notés de la manière suivante :



* Test validé



* Test partiellement validé



* Test non validé

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Conséquence | Résultat |
| La date de réalisation du test | Description de l’élément testé sur le projet. | Que s’est-il passé durant le test. | Est-ce que le test est validé ou non. |

**Environnement – Méthode – Résultat attendu – résultat obtenue**

Les tests seront effectués manuellement sur l’application, ils seront réalisés dans l’ordre d’apparition des pages.

## Risques techniques

Concernant la complexité de ce projet, je risque d’avoir des difficultés quant aux gestions de données étant donné que c’est un élément que je n'ai jamais exploré dans une application IOS.

Lors de ce projet, je vais également devoir me documenter durant mes périodes de projet, mais aussi lors de mon temps libre afin de gagner du temps et me consacrer uniquement à la réalisation lorsque je me trouverais en période de projet.

Afin de faciliter le stockage des données, je compte réaliser une gestion des utilisateurs avec un système d’authentification à l’aide de Firebase, ce qui me permettrait d’utiliser Firebase pour stocker/modifier/supprimer mes données.

Le système d’authentification était le sujet sur lequel je me suis formé lors du pré-TPI, ce qui me faciliterait sa mise en place et me permettrai un gain de temps considérable.

## Planification

**Total Heures planifiées : 89h50 min**

**Projet du 05.05.2021 jusqu’au 04.06.2021**

Il y aura lors de ce projet **50%** de temps consacrer à l’implémentation/réalisation de l’application

**23%** de documentation, **20%** d’analyse des technologies liée au projet et sa conception.

Enfin pour finir, je consacrerai **7%** de mon temps lors de ce projet à effectuer des tests sur mon application.

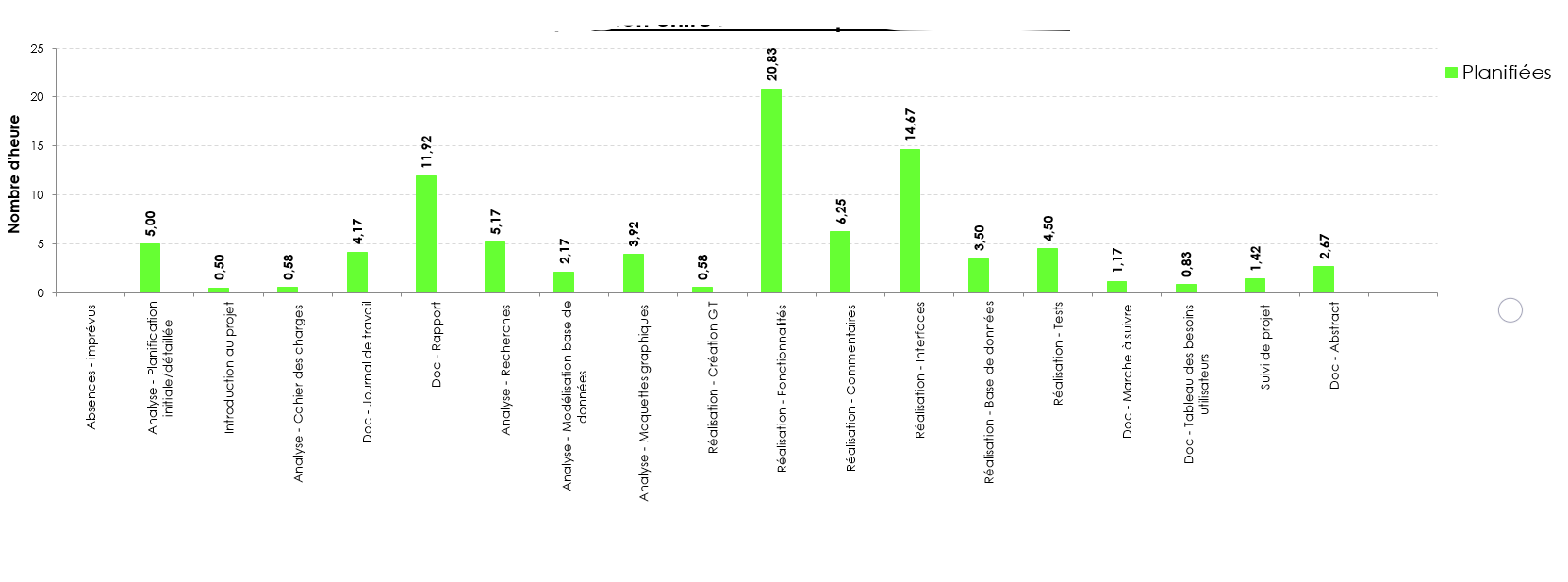
**

Figure 12 Graphique : Bilan de la planification initiale

## Dossier de conception

### Matériel Hardware / Système d’exploitation

1. **Ordinateur ETML - Windows 10 Education – Version 1909**

3,40 GHz Intel Core i7-6700 CPU

32 Go de mémoire RAM

1. **MacBook Pro - MacOS Big Sur – Version 11.3**

2,2 GHz Intel Core i7 quatre cœurs

16 Go de mémoire RAM / 1600 MHZ DDR3

Intel Iris Pro 1536 Mo

1. **Iphone 11 - IOS – Version 14.5.1**

Téléphone sur lequel l’application sera testée et devra être fonctionnelle.

### Logiciels

1. **Moqups.com / OverFlow.io / Adobe Photoshop**

Outils utilisés pour la réalisation des maquettes graphiques de l’application, des modèles relationnels ainsi que les éléments visuels qui composeront l’application (Logo, Icône, Bouton etc.)

1. **Xcode – Version 12.5**

Outil utilisé pour le developpement de mon application IOS

1. **Swift 5**

Langage de programmation avec lequel a été développé mon application.

1. **CocoaPods – Version 1.10.7**

Gestionnaire de dépendances, il m’a permis d’installer des API ou des éléments de différentes librairies pour réaliser mon application.

1. **Firebase – SDK (Software Development Kit) 8.0.0 IOS**

Service utilisé pour l’’hébergement et l’authentification au sein de mon application.

1. **GitHub / GitHub Desktop – Version 2.8.1**

Service d’hébergement de mon projet en ligne

(Répertoire : <https://github.com/KillermyB/TPI_inventaireHabits_BoukhlifaK>)

Outil permettant d’effectuer mes commits directement depuis mon bureau.

1. **FireBase – Firestore**

Service d’hébergement Cloud des données de mon application.

1. **Suite Office 365 – Version 2016**

Suites de logiciels utilisés pour la réalisation de ma documentation.(Microsoft Word, Excel, Outlook et Powerpoint)

1. **Scribens – Reverso Correcteur**

Outils de correction orthographique utilisés lors de la rédaction de la documentation de mon projet.

### Suivi de projet

Lors de ce projet, je vais permettre à mon chef de projet ainsi qu’aux experts de suivre mon projet en temps réel via un [répertoire Git](https://github.com/KillermyB/TPI_inventaireHabits_BoukhlifaK.git)[[1]](#footnote-1) mis à jour quotidiennement.

Je remplirai également de manière quotidienne un journal de travail attestant de mon avancement dans le projet qui permettra de suivre les différentes tâches effectuées et finalement comparer ma planification de projet avec sa réalisation.

Je notifierai également mon chef de projet et mes experts chaque mercredi et vendredi en fin de journée afin de leur mettre à disposition la dernière version de mon rapport et de mon journal de travail quotidien.

### Base de données : Firebase

1. **Firebase Authentification**

Pour réaliser ce projet, j’ai utilisé le service d’authentification de Google : Firebase Authentification

Il fournit des services de backend, des SDK simples d’implémentation et une multitude de bibliothèques d’interface d’authentification. (Google, Facebook, Twitter, etc.)

J’ai implémenté pour mon projet l’authentification via une adresse-mail et un mot de passe, l’utilisateur peut donc créer son compte, il obtient alors un User-ID définit par Firebase.

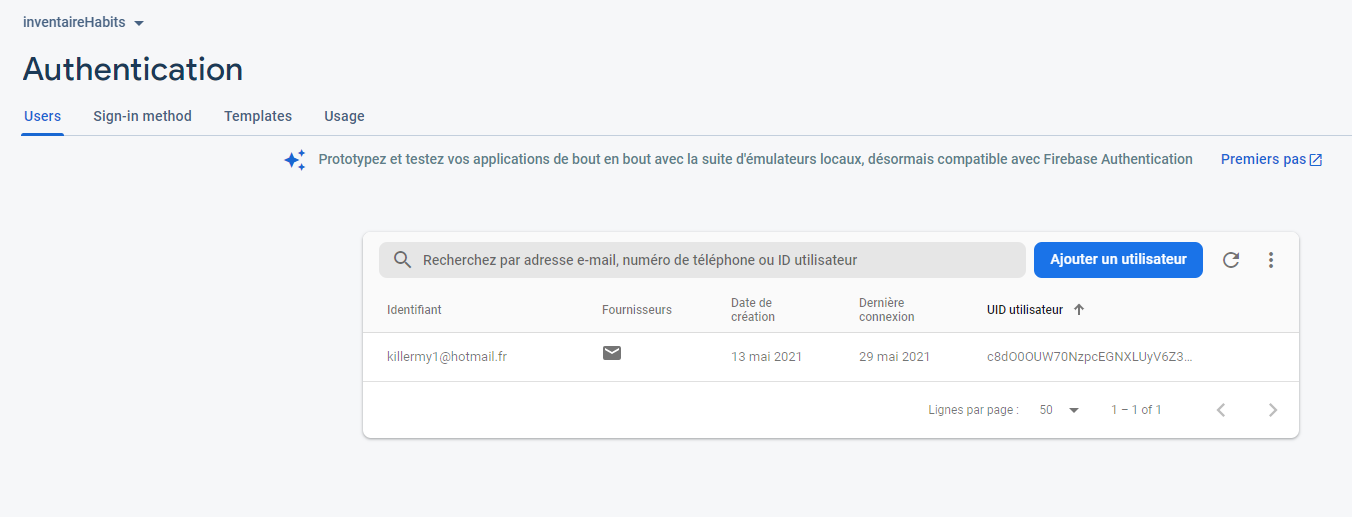
Cet User-ID sera ensuite vérifié avec les informations de connexion par Firebase et elles seront renvoyées une fois vérifiées vers mon application pour confirmer la connexion ou refuser une connexion, ce qui me permet d’avoir une application sécurisée et fiable.

Figure 13 Firebase Authentification : Page contenant les Users

1. **Firebase – Firestore**

Concernant le stockage des données de mon application, j’ai utilisé un autre service de Firebase : Firestore Database

C’est une base de données cloud NoSQL qui a la particularité d’être flexible et évolutive.

J’ai utilisé un système de stockage orienté documents qui consistent à identifier un élément via une clé lié à une valeur, qui dans ce cas, est un document, dans le document, nous y stockons une liste de champs qui a une valeur associée.

Ces documents sont de type JSON.

Collection : t\_product

Une collection peut contenir de multiples documents.

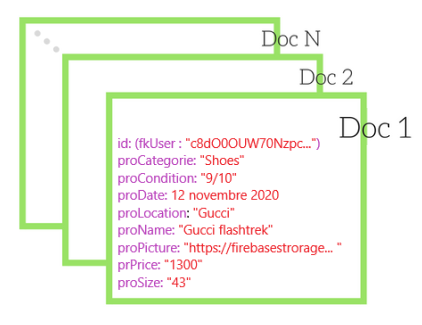


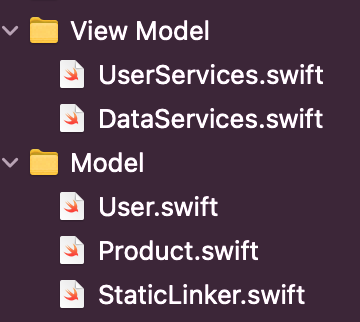
Figure 14 Firebase Firestore : Explication du stockage orienté "documents"

### Architecture du programme : Modèle-Vue-Contrôleur

Pour la réalisation de ce projet, j’ai utilisé l’architecture MVC qui consiste à diviser le développement d’une application en trois types de modules différents.

Les **Modèles** contiennent les données à afficher :

Figure 15 Architecture - Application : Modèle



UsersServices.swift : C’est le modèle qui permet de récupérer les informations des utilisateurs dans la base de données Firebase Auth, c’est cette Class qui fait le lien entre l’application et Firebase.

DataServices.swift : c’est le modèle qui permet de récupérer les informations des articles dans la base de données Firestore.

C’est cette Class qui fait le lien entre l’application et Firebase.

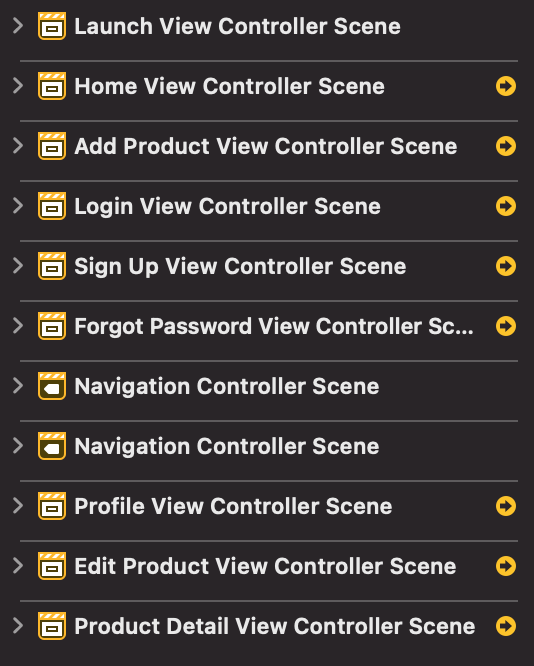
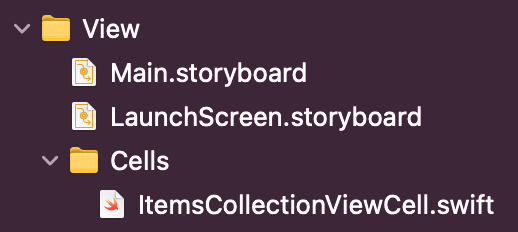


Figure 16 Architecture – Application : Vue - Dossiers

Figure 17 Architecture - Application : Vues de l’application

Les **Vues** contiennent la présentation de l’interface graphique, c’est ce que l’utilisateur verra lors de l’utilisation de l’application.

Ces différentes vues font partie de la « Main.storyboard » qui correspond à une page contenant tous les écrans de l’application, la gestion de la navigation entre les pages.

Il y a également la « LaunchScreen.storyboard » qui correspond à l’écran de chargement afficher à l’ouverture de l’application.

Et un dossier « Cells » contenant les éléments graphiques contenus dans une case d’article, elle est utilisée comme template d’affichage.

StaticLinker.swift, c’est la mise en place d’un lien statique permettant est l’appel depuis d’autres fonctions ou classes et qui permet de récupérer toutes les informations précédemment initialisées dans les pages Model : User et Product.

User.swift et Product.swift permet la structuration des données qui seront communiquées par l’utilisateur et l’ajout d’articles,etc.

C’est sur ces pages que sont définis les types (String, Double, Timestamp etc.) de chaque information a stocké dans les fichiers JSON.

Les Contrôleurs contiennent la logique concernant les actions effectuées par l’utilisateur.

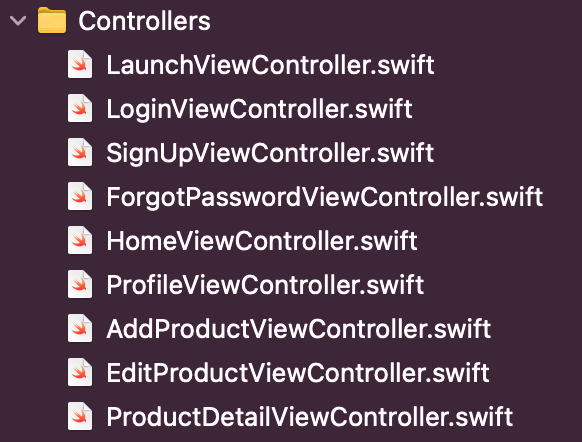


Figure 18 Architecture - Application : Contrôleurs - Dossier

Chacune des vues de mon application possède un contrôleur qui définit les actions effectué sur celui-ci.

Lorsqu’une action est effectuée sur un contrôleur, il donne l’information au modèle qui modifie les données selon l’action.

La vue lit ensuite les données du modèle pour les afficher.

### CocoaPods – Gestionnaire de dépendances

C’est un gestionnaire de dépendances au niveau de l’application pour Objective-C, Swift et tout autre langage s’exécutant sur le moteur d’exécution d’Objective-C.

Il permet de simplifier l’importation de différentes bibliothèques externes (frameworks), permettant d’ajouter des fonctionnalités ou des éléments à une application, et permet de mettre à jour simultanément l’ensemble des bibliothèques installés avec la commandes suivante : « pod update »

|  |  |
| --- | --- |
| **Pods installées** | **Description** |
| Firebase[[2]](#footnote-2) |  |
| FirebaseAuth |  |
| Firebase Core |  |
| Firebase Firestore |  |
| Firebase Storage |  |
| IQKeyboardManagerSwift |  |
| MBProgessHUD – 1.2.0 |  |
| SDWebImage – 5.0 |  |

### Cas d’utilisation

Figure 19 Maquette : Cas d'utilisation de l'application

### Modèle Logique de Données

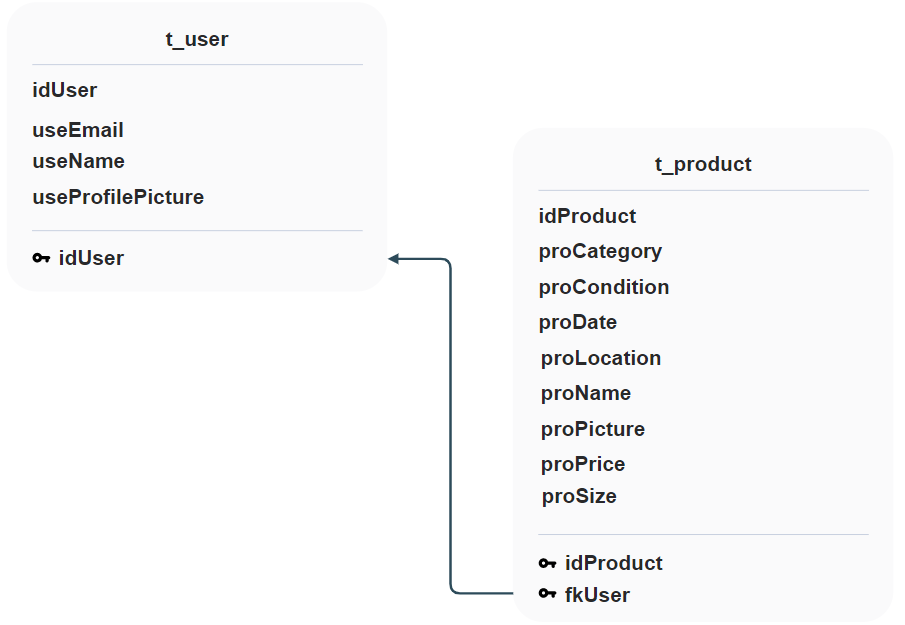
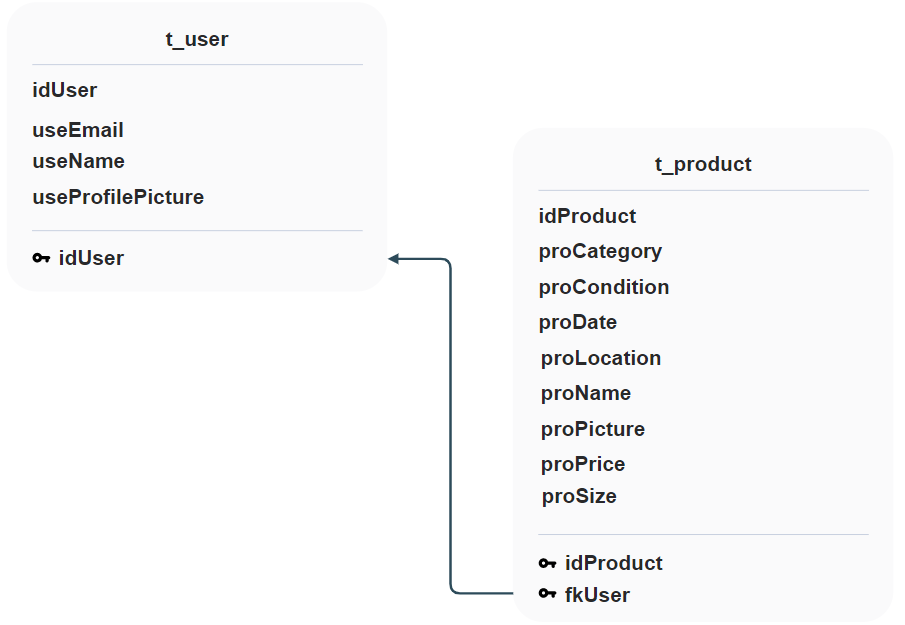


Figure 20 Maquette : Modèle logique des données



idUser : **Identifiant** de l’utilisateur dans la table

*Type String :*

useEmail : enregistrement du **mail** de l’utilisateur -

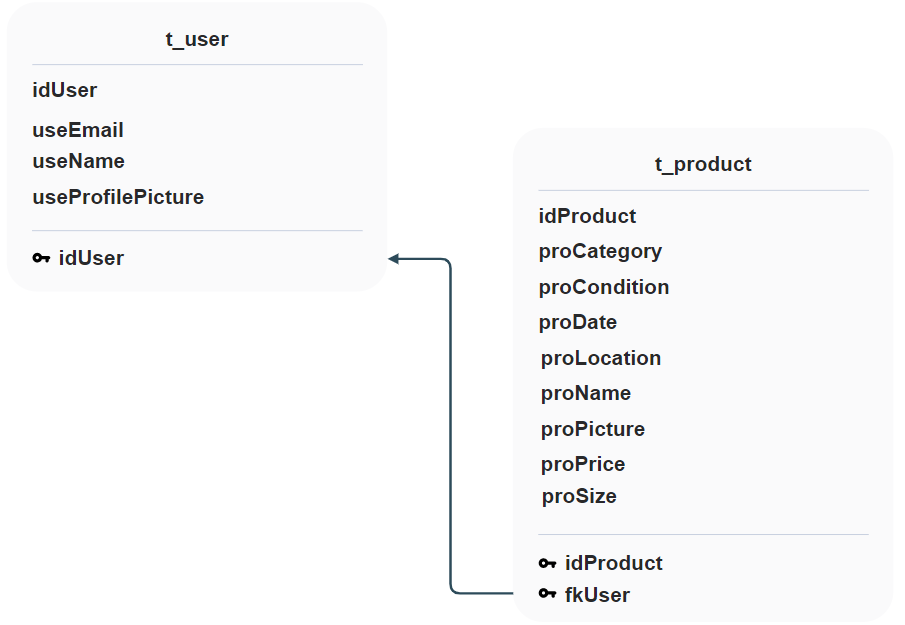
*Type String :*

useName : enregistrement du **nom** de l’utilisateur

*Type String :*

useProfilePicture : enregistrement **de l’image de profil** de l’utilisateur - String

idUser : Clé primaire de la **table t\_user**



*Type number :*

*Type timestamp :*

*Type String :*

*Type String :*

*Type String :*

*Type String :*

*Type String :*

*Type String :*

proCatergory : enregistrement de la **catégorie de l’article** (T-shirt, etc.)

fkUser: **Clé étrangère** de la **table t\_user**

idProduct: **Clé primaire** de la table **t\_product**

proSize: enregistrement de la **taille de l’article** (XS, S, M etc.)

proPrice: enregistrement du **prix de l’article**

proPicture: enregistrement de **l’image de l’article**

proName: enregistrement du **nom de l’article** (Hoodie Dior« CD », Vest Louis Vuitton)

proLocation: enregistrement du **lieu d’achat de l’article**

proDate : enregistrement de la **date d’achat** **de l’article**

proCondition : enregistrement de la **condition de l’article** (10/10 Neuf, 5/10 Usée etc.)

idProduct : **Identifiant** du produit (Article) dans la table

* *les mots-clés, choisir une formule*

*d'hébergement, définir la méthode de mise à jour, …*

* *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
* *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme…*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

# Réalisation

## Liste des fichiers

## Firebase : Authentification

## Firebase : Firestore

## Créations des cellules d’article

## JSON

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

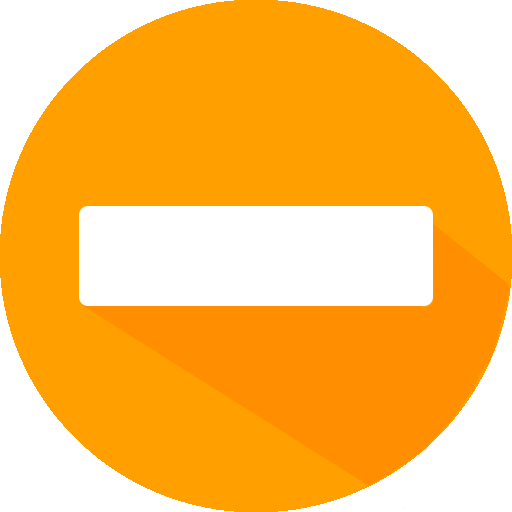
* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

Ces tests seront effectués afin de vérifier que le projet fonctionne correctement.

Les tests seront notés de la manière suivante :



* Test validé



* Test partiellement validé



* Test non validé

|  |  |  |
| --- | --- | --- |
| Description | Conséquence | Résultat |
| **Page connexion**   * Bouton mot de passe oublié et créer un compte |  |  |
| **Page connexion**   * Connexion d’un utilisateur |  |  |
| **Page connexion**   * Erreur indiquée lors de la saisie des informations de connexion. |  |  |
| **Page création d’un compte**   * Ajout d’une image de profil |  |  |
| **Page création d’un compte**   * Saisies des informations enregistrées et création d’un compte possible. |  |  |
| **Page création d’un compte**   * Bouton retour |  |  |

|  |  |  |
| --- | --- | --- |
| **Page mot de passe oublié**   * Saisi de l’email |  |  |
| **Page mot de passe oublié**   * Possibilité de modifier son mot de passe |  |  |
| **Page principale**   * Bouton profil / Affichage de la photo de profil |  |  |
| **Page principale**   * Liste des articles affichés |  |  |
| **Page principale**   * Articles cliquables pour accéder à la page détaillée de l’article |  |  |
| **Page principale**   * Bouton d’ajout d’un article |  |  |
| **Page principale**   * Bouton sélection d’un article * Possibilité de partager un article |  |  |
| **Page principale**   * Possibilité de trier les articles par catégorie |  |  |
| **Page détail d’un article**   * Bouton retour |  |  |
| **Page détail d’un article**   * Image de l’article * Nom de l’article * Lieu d’achat * Catégorie de l’article * Taille * Condition de l’article * Date d’achat * Prix d’achat |  |  |
| **Page détail d’un article**   * Bouton modification de l’article |  |  |

|  |  |  |
| --- | --- | --- |
| **Page ajout d’un article**  Champs d’informations :   * Image de l’article * Nom de l’article * Lieu d’achat * Catégorie de l’article * Taille * Condition de l’article * Date d’achat * Prix d’achat |  |  |
| **Page ajout d’un article**   * Bouton ajout |  |  |
| **Page ajout d’un article**   * Bouton retour |  |  |
| **Page modification d’un article**  Champs d’informations modifiables :   * Image de l’article * Nom de l’article * Lieu d’achat * Catégorie de l’article * Taille * Condition de l’article * Date d’achat * Prix d’achat |  |  |
| **Page modification d’un article**   * Bouton enregistré les modifications |  |  |
| **Page modification d’un article**   * Bouton suppression de l’article |  |  |
| **Page modification d’un article**   * Bouton retour |  |  |
| **Page profil**   * Affichage des informations de l’utilisateur |  |  |
| * Modification de l’image de profil de l’utilisateur |  |  |
| * Statistiques détaillés des articles disponible dans la collection de l’utilisateur |  |  |
|  |  |  |

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les* numéros de versions

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

## Bilan de la planification

## Bilan des fonctionnalités demandées

## Bilan personnel

# Annexes

## Résumé du rapport du TPI / version succincte de la documentation

## Webographie

## Planification Initiale

Dossier en annexe du rapport.

## Journal de travail

Dossier en annexe du rapport.

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## Manuel d’installation

## Manuel d’Utilisation

## Archives du projet

1. <https://github.com/KillermyB/TPI_inventaireHabits_BoukhlifaK.git> [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)