

**Document Technique**

|  |  |
| --- | --- |
| **Nom du Projet** |  |
| **Type de document** | Document Technique |
| **Version** | 1.2.3 |
| **Date** | 10/12/2019 |
| **Mots-clés** | Architecture – Application Mobile – IA |
| **Auteurs** | ROGER Loïc  RAZAFINDRAZAKA Miranto |

Table des matières

**1. Information du document Technique**3

**2. Fonctionnement de l’Application4**

2.1 Description de l'application4

2.2 Décomposition du projet5

2.3 Architecture globale5

**3. Côté Client4**

3.1 Architecture5

3.1.1 Les Pages5

3.1.1.1 Authentification5

3.1.1.2 Inscription5

3.1.1.3 Dashboard5

3.1.1.4 Dressing5

3.1.1.5 Profil5

3.1.1.6 Ajouter un vêtement5

3.1.2 Le Footer5

3.1.2.1 Bouton Profil5

3.1.2.2 Bouton Dressing5

3.1.2.3 Bouton Dashboard5

3.1.2.4 Bouton « Ajouter un vêtement »5

3.2 Outils utilisés 5

**4. Côté Serveur4**

4.1 Architecture5

4.2 Outils utilisés 5

4.3 API5

**ANNEXES4**

A Les Catégories de Vêtements5

A.1 Les Hauts5

A.2 Les Bas5

A.3 Les Chaussures5

A.4 Les Accessoires5

A.5 Les Manteaux5

A.6 Les « Full-body »5

B Maquettes 5

1. **Information du document Technique**
2. **Fonctionnement de l’Application**
   1. ***Description de l’Application***

Ce projet sera une app-mobile permettant à un utilisateur de rentrer ses vêtements dans l’application. Une fois son dressing intégré, en fonction de la localisation et de la météo, un bot proposera à l’utilisateur plusieurs vêtements adéquats en fonction de la température et de la météo (des bottes et un manteau s’il pleut ou un t-shirt et des baskets s’il fait beau).

Le téléchargement de l’application sera gratuit mais comportera des achats intégrés sous la forme d’abonnement : Par défaut, un utilisateur aura un abonnement « starter », cet abonnement permettra à l’utilisateur d’intégrer un nombre de vêtements limité dans l’application. En revanche, si l’utilisateur décide de souscrire à l’abonnement « premium », il pourra intégrer un nombre de vêtement illimité dans l’application.

* 1. ***Décomposition du projet***
  2. ***Architecture globale***

1. **Côté Client**
   1. ***Architecture***
      1. ***Les Pages*** *(Voir annexe : Maquettes)*
         1. ***L’authentification***

Email – password – google- facebook – register

* + - 1. ***L’inscription***

Firstname – lastname – email – sexe (h – f) – sexe – age – password-confirm

* + - 1. ***Le Dashboard***

Informations météo actuelle (sur la position)

Proposition de tenue (vue avec détails des habits)

Changement de tenue (autre proposition)

* + - 1. ***Le Dressing***

Choix (hauts – bas – chaussures – accessoires)

Listes

* + - 1. ***Le Profil***

Modification des infos (firstname – lastname -password – sexe – age - photo de profil)

Récurrence du look (l’utilisateur peut choisir la récurrence du look proposé par l’application : par exemple ne pas proposer le même look pendant 5 jours)

L’utilisateur peut dire à l’application s’il est plus frileux ou non. Dans ce cas, l’application choisira des vêtements plus chauds que la normale en fonction de la température.

* + - 1. ***Ajouter un vêtement***

Prendre une photo du vêtement (mobile)

Validation de la photo

Choix de catégorie

Choix de sous-catégorie

* + 1. ***Le Footer***
       1. ***Bouton Profil***

Mettre un raccourci sous la forme d’un bouton pour revenir au Profil.

* + - 1. ***Bouton Dressing***

Mettre un raccourci sous la forme d’un bouton pour revenir au Dressing de l’utlisateur.

* + - 1. ***Bouton Dashboard***

Mettre un raccourci sous la forme d’un bouton pour revenir au Dashboard.

* + - 1. ***Bouton « Ajouter un vêtement »***

Mettre un raccourci sous la forme d’un bouton pour ajouter un vêtement.

* 1. ***Outils Utilisés***

Le framework utilisé pour réaliser l’application côté client (Front-end) est Angular.

Pour pouvoir réaliser cette interface, l’utilisation de librairies est recommandée comme Angular Material.

1. **Côté Serveur**
   1. ***Architecture***

Graphique UML :

1. Vêtements

* ID (entier)
* ID\_user (entier)
* Image

1. Info-vêtement

* ID (entier)
* ID\_vêtement (entier)
* ID\_details-type (entier)

1. Type
   * ID (entier)
   * Nom (string)
2. Sous-type
   * ID (entier)
   * Nom (string)
3. Détails\_type

* ID (entier)
* ID\_type (entier)
* ID\_sous-type (entier)

1. User

* ID (entier)
* Nom (string)
* Prénom (string)
* Age (entier)
* Sexe (string)
* MDP (string)

1. Info-user

* ID (entier)
* Frileux (booléen)
* Récurrence\_look (entier)
  1. ***Outils utilisés***

Le framework utilisé pour réaliser l’application côté serveur (Back-end) est NodeJS (ExpressJS).

* 1. ***API***

**ANNEXES**

**A) Les catégories de vêtements**

*Catégories générales*

1. *Hauts*
2. *Bas*
3. *Chaussures*
4. *Accessoires*
5. *Manteaux*
6. *Full-body*
   1. Les hauts

Possèdent des sous-catégories :

* Débardeur
* T-shirt
* Pull
* Manches longues
* Chemises
* Sweat
  1. Les bas

Possèdent des sous-catégories :

* Short
* Jogging
* Jeans
* Pantalon
* Jupe
* Legging
* Collant + jupe
  1. Les chaussures

Possèdent des sous-catégories :

* Tongues
* Baskets
* Chaussures de villes
* Escarpins
* Bottes
* Sandales
* Ballerines
* Mocassins
* Derby
  1. Les accessoires

Possèdent des sous-catégories :

* Casquette
* Chapeau
* Parapluie
* Foulard / écharpe
* Gants
* Bonnets
* Sac à main
  1. Les manteaux

Possèdent des sous-catégories :

* Veste
* Manteau
* Doudoune
* Trench
* Blazer
* Gilet
* Anorak
  1. Full-body

Possèdent des sous-catégories :

* Robe
* Combinaison
* Combinaison-short
* Combinaison-pantalon
* Robe manche courte

**B) Les Maquettes**

**Authentification :**

**Inscription :**

**Dashboard :**

**Dressing :**

**Profil :**

**Ajouter un vêtement :**