

**Document Technique**

|  |  |
| --- | --- |
| **Nom du Projet** | **DressMeUp** |
| **Type de document** | Document Technique |
| **Version** | 1.0.0 |
| **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’Application3**

**3. Côté Client3**

3.1 Architecture3

3.1.1 Les Pages3

3.1.1.1 Authentification3

3.1.1.2 Inscription3

3.1.1.3 Dashboard3

3.1.1.4 Dressing4

3.1.1.5 Profil4

3.1.1.6 Ajouter un vêtement4

3.1.2 Le Footer4

3.1.2.1 Bouton Profil4

3.1.2.2 Bouton Dressing4

3.1.2.3 Bouton Dashboard4

3.1.2.4 Bouton « Ajouter un vêtement »4

3.2 Outils utilisés 4

**4. Côté Serveur5**

4.1 Schéma de la base de données5

4.2 Outils utilisés 5

4.3 API6

**ANNEXES7**

A Les Catégories de Vêtements7

A.1 Les Hauts7

A.2 Les Bas7

A.3 Les Chaussures7

A.4 Les Accessoires8

A.5 Les Manteaux8

A.6 Les « Full-body »8

B Maquettes 9

1. **Information du document Technique**

Ce document montre les spécifications techniques du projet : le côté serveur ainsi que le côté client. Les maquettes de cette application sont aussi disponibles en fin de document (Annexes).

1. **Fonctionnement de l’Application**

Ce projet sera une app-mobile permettant à un utilisateur de rentrer ses vêtements dans l’application avec diverses informations. 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. **Côté Client**
   1. ***Architecture***
      1. ***Les Pages*** *(Voir annexe : Maquettes)*
         1. ***L’authentification***

Formulaire disposant de : Email – Mot de passe – Connexion par google – Connexion par Facebook – Bouton Inscription (si pas encore inscrit)

* + - 1. ***L’inscription***

Formulaire disposant de : Prénom – Nom – Email – Sexe (H – F) – Age – Mot de passe – confirmation mot de passe

* + - 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 – sex – 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 »***

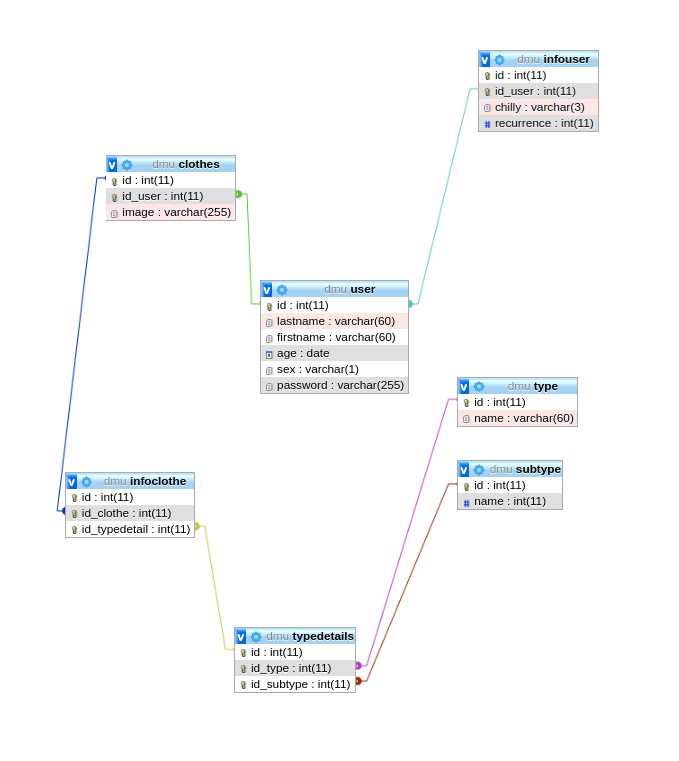
Mettre un raccourci sous la forme d’un bouton pour ajouter un vêtement.

* 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. ***Schéma de la base de données***



* 1. ***Outils utilisés***

La technologie utilisée pour réaliser l’application côté serveur sera du Javascript avec NodeJS.

* 1. ***API***

Pour la manipulation des données il y aura un API REST réalisé avec le framework Express (requêtes http et routes, les routes ne sont pas exhaustives et peuvent changer selon les besoins).

POST /clothes (ajout de vêtement avec tout ses informations)  
PUT /clothes/ :id\_clothe (mise à jour d’un vêtement)  
DELETE /clothes/ :id\_clothe (suppression d’un vêtement)  
GET /clothes/ :id\_user (récupération des informations des vêtements d’un utilisateur)  
POST /signup (ajout d’un utilisateur)  
POST /signing (connexion)  
PUT /user/ :id\_user (mise à jour des informations d’un utilisateur)  
GET /user/ :id\_user (récupération des informations d’un utilisateur)

**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 :

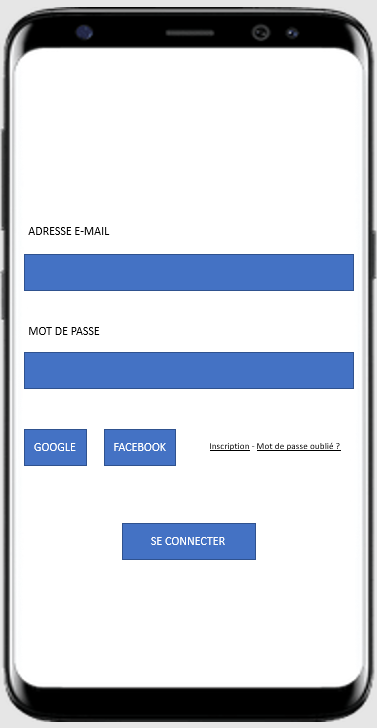
* Veste
* Manteau
* Doudoune
* Trench
* Blazer
* Gilet
* Anorak
  1. Full-body

Possèdent des sous-catégories :

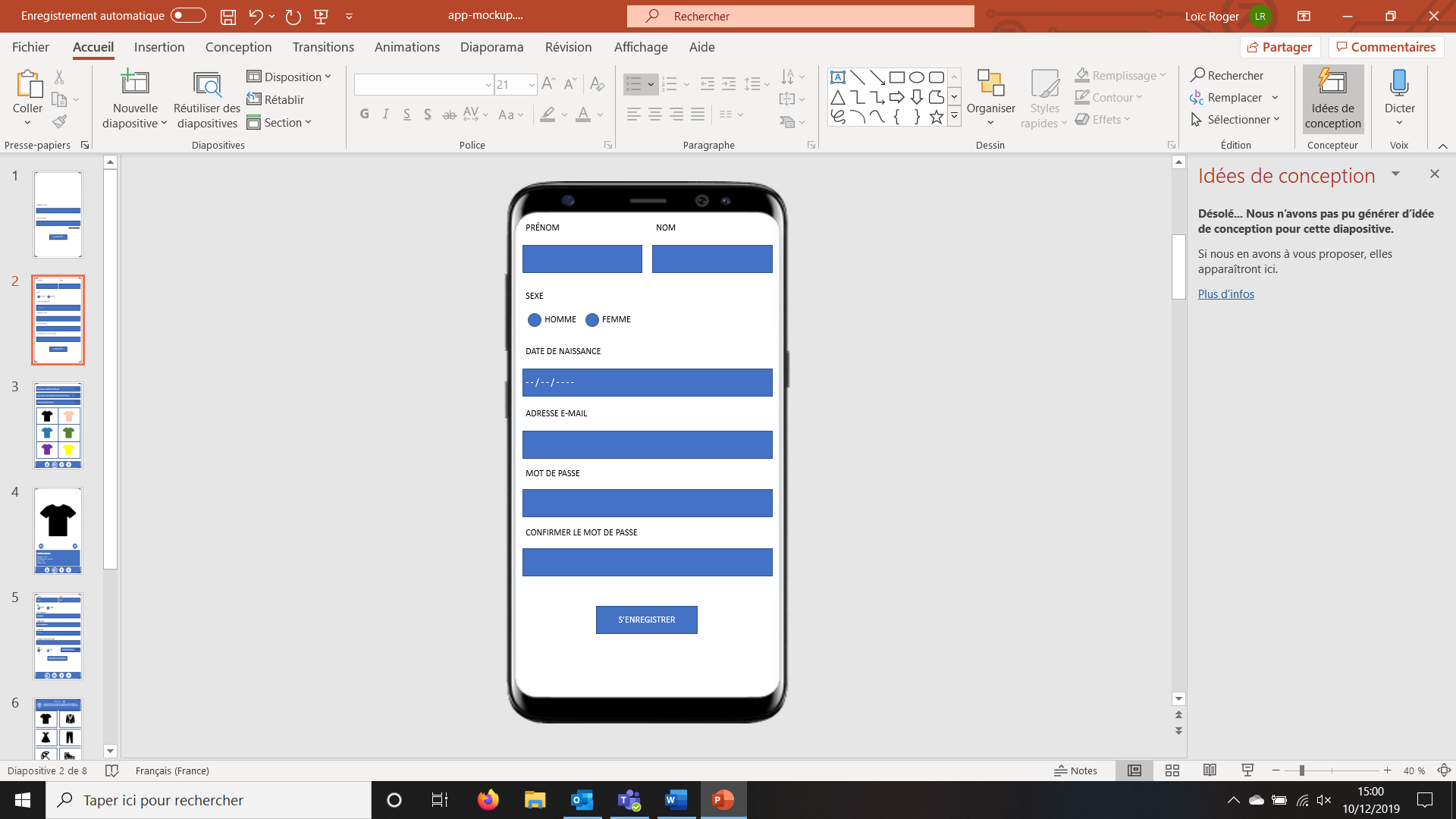
* Robe
* Combinaison
* Combinaison-short
* Combinaison-pantalon
* Robe manche courte

**B) Les Maquettes**

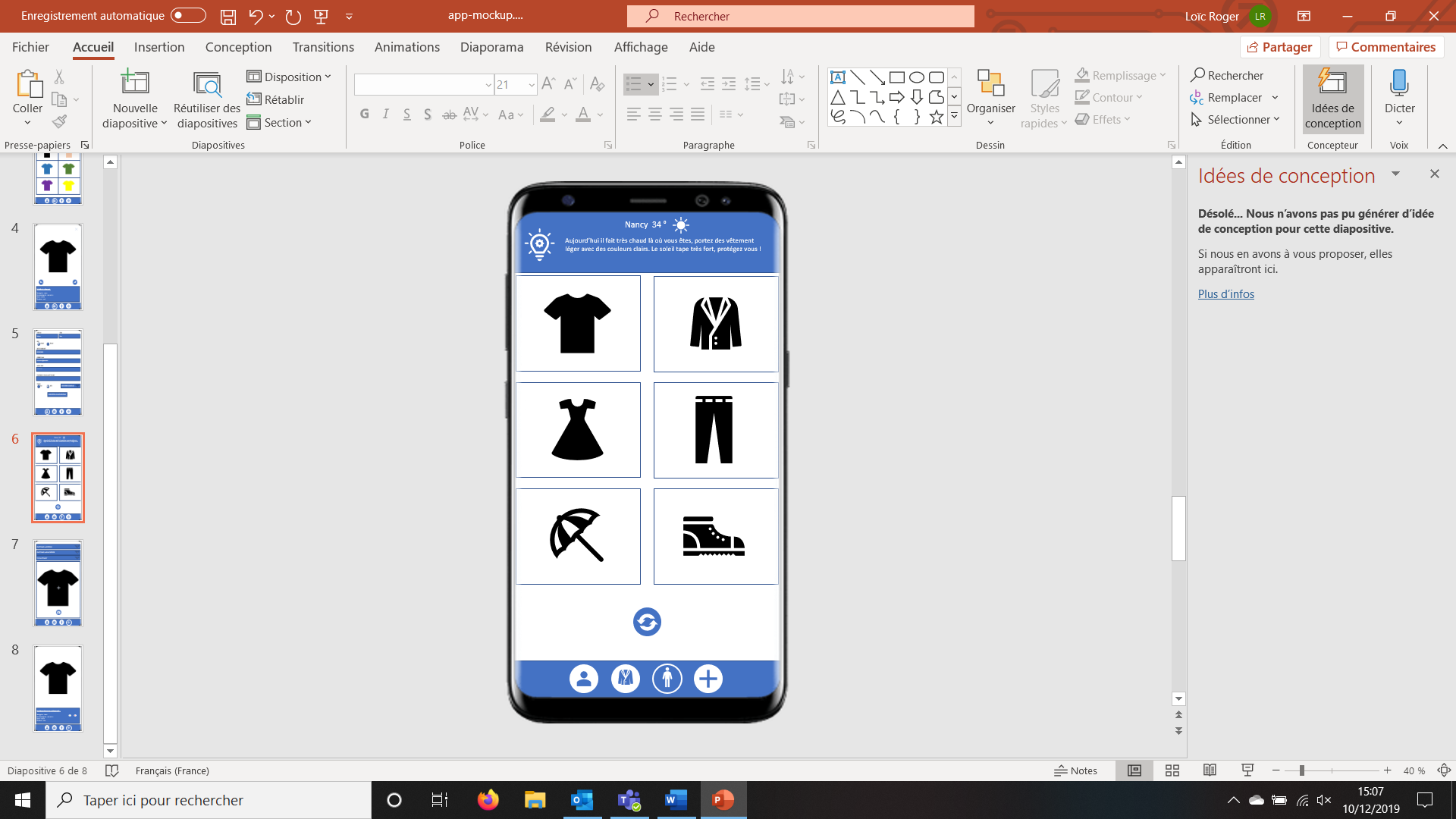
**Authentification :**



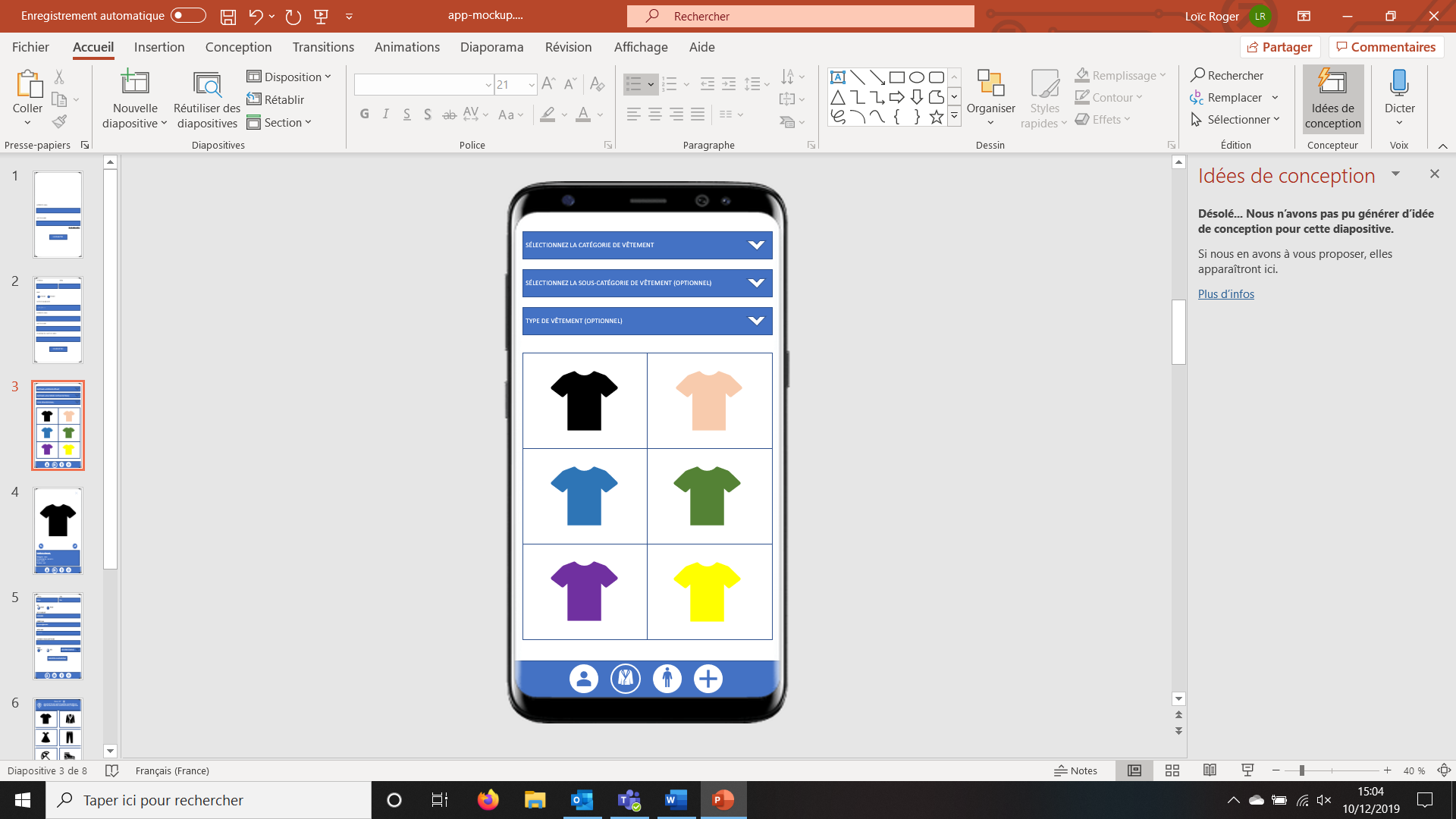
**Inscription :**

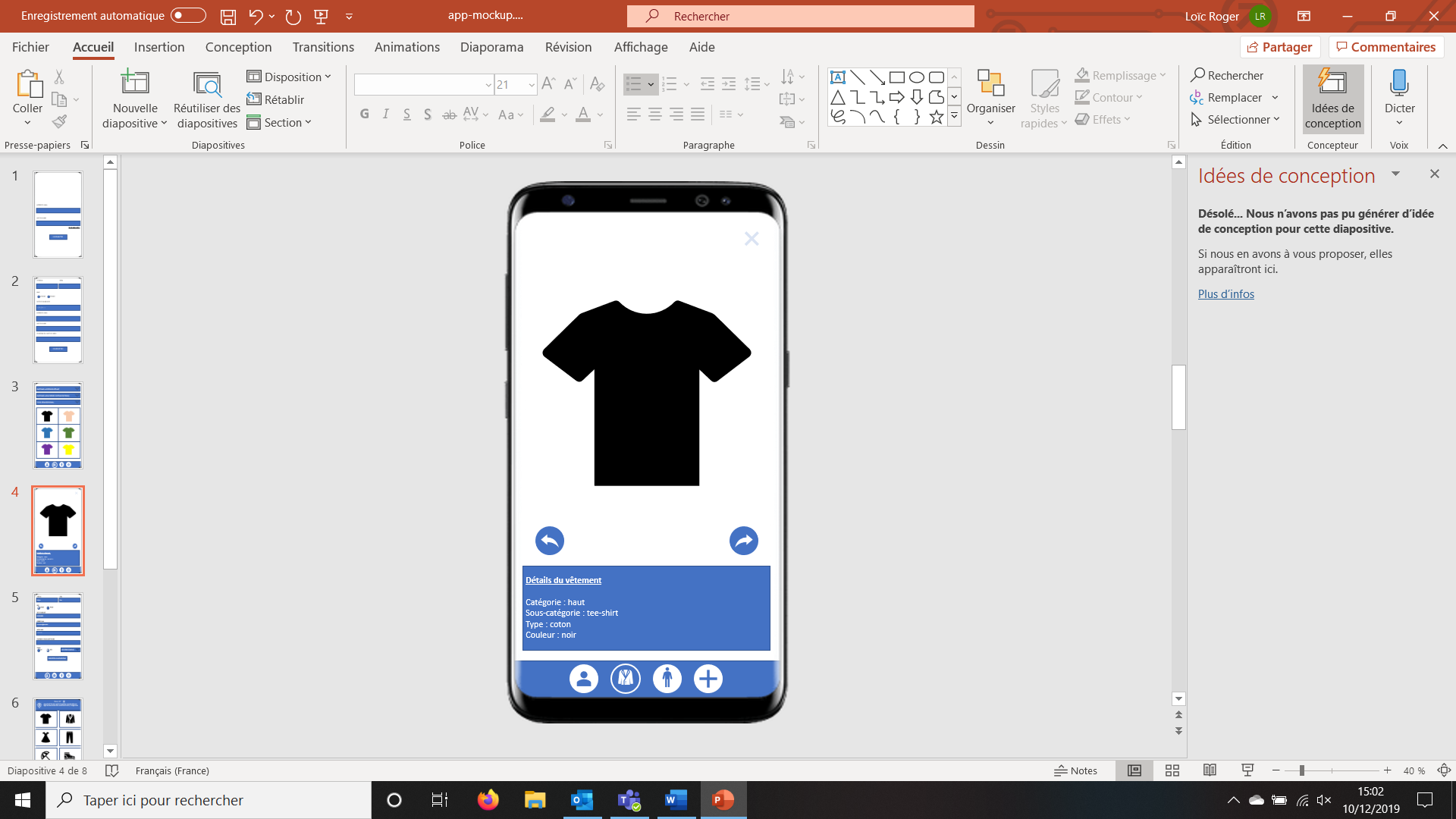


**Dashboard :**

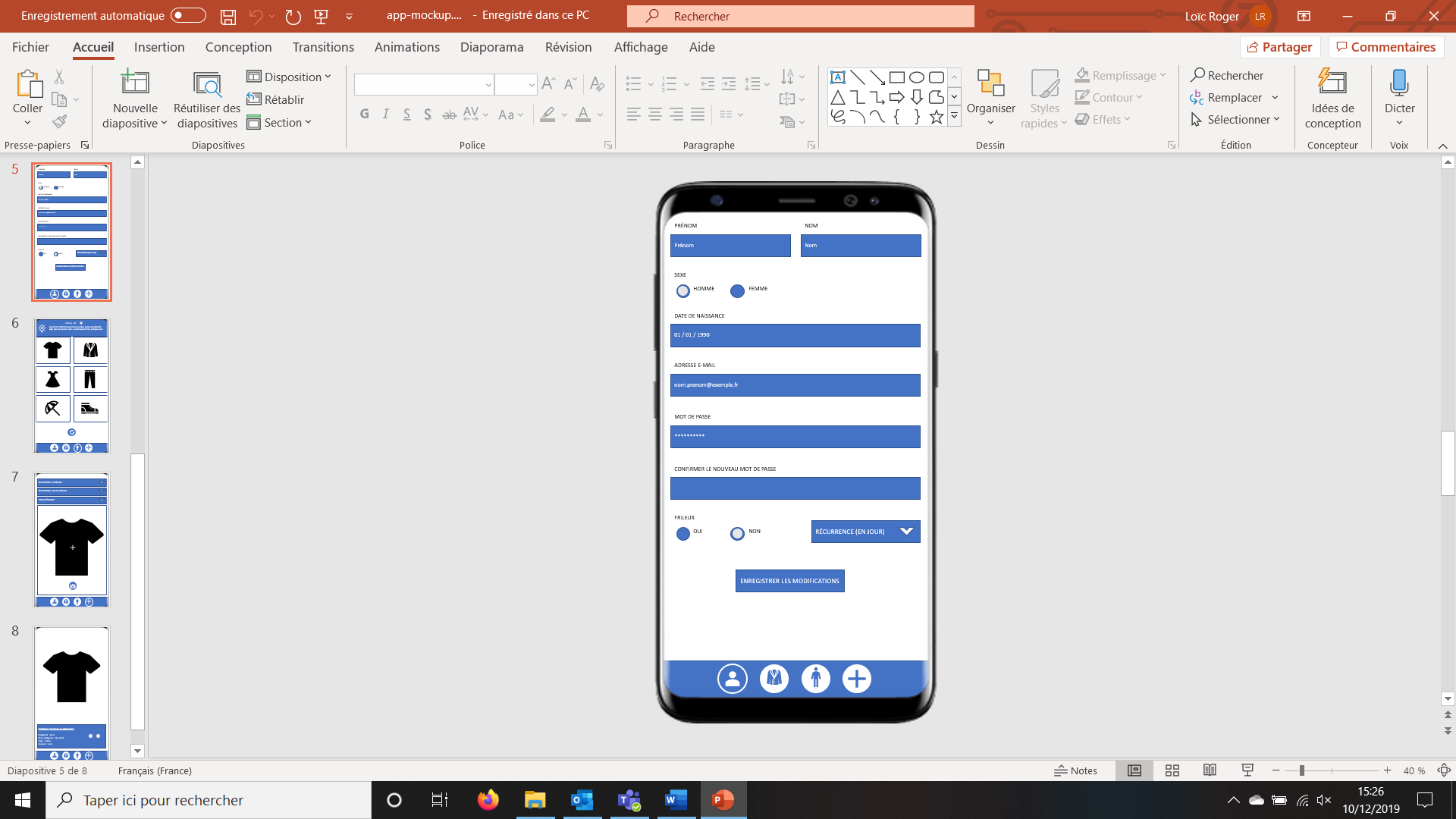


**Dressing :**





**Profil :**



**Ajouter un vêtement :**

