**Cahier des charges – Application de réalisation de devis d'assurance**

L'objectif de l'application est de réaliser et de suivre des devis d'assurance pour des véhicules ou des habitations.

La conduite de développement se fera au travers un Kanban (ce Kanban incluant l’homologation).

A la fin du développement, une présentation sera faite, un support visuel doit servir de base à la présentation. Une démo de l’application peut également être intéressante.

**Architecture technique**

|  |  |
| --- | --- |
| Base de données | MySql |
| Framework UI | JSP |
| Application frontal | Tomcat (ou FatJar) / Spring (MVC/IOC/Data) |
| Micro service | Vert.X (qui sera lié à un MySql) |

Une couche de sécurité devra être implémentée, côté Application frontal on utilisera Spring Sécurity et côté micro service, Vert.X utilisera un token JWT (<http://vertx.io/docs/vertx-auth-jwt/js/>)

Cinématique d’authentification :

* L’utilisateur accès à l’application via une page de login.
* Submit vers Spring Sécurity
* Authentification en utilisant un microservice et récupération de l’utilisateur ainsi qu’un token JWT généré par le microservice.
* Persistence en session du token généré pour l’utilisateur

Vert.X refusera tout appels hormis l’authentification si aucun token valide n’est passé à m’appels.

Les appels entre Spring et Vert.X se feront via des appels Rest (ON reste ouvert à d’autres propositions).

**Description des écrans**

|  |  |
| --- | --- |
| Ecran 1 | Une page de login |
| Ecran 2 | Une page de synthèse (Devis en cours) |
| Ecran 3 | Un wizard pour les devis voitures |
| Ecran 4 | Un wizard pour les devis habitation |
| Ecran 5 | Une page de profil pour les utilisateurs |
| Ecran 6 (Bonus 1) | Les écrans d’admin (CRUD des listes) |
| Ecran 7 (Bonus 2) | Un écran d’administration qui liste en temps réel les utilisateurs qui sont en train de réaliser un devis avec mention de l’étape en cours |

**Règles de gestions générales :**

* Il doit être possible de reprendre un devis en cours (on enregistre l’état entre chaque changement d’écran)
* Il est possible d’annuler un devis en cours de rédaction (donc suppression des étapes enregistrées)
* Les différentes listes (véhicules, marques, etc ..) seront fournies par des microservices
* A la fin d’un devis, on l’enregistre dans la base de l’application et si tout se passe bien on publie un résumé dans un microservice (nom, prenom, identifiant, type, garantie, prix)
* On doit gérer les rôles (pour l’instant seulement 2 (USER et ADMIN)

**Description de l’écran des devis véhicules**

|  |  |  |
| --- | --- | --- |
| Wizard page 1 | Nom / prenom de l’utilisateur | Non modifiable |
| Wizard page 1 | Nom du devis | Obligatoire |
| Wizard page 1 | Marque | Obligatoire |
| Wizard page 1 | Modèle | Obligatoire |
| Wizard page 1 | Carburant | Obligatoire |
| Wizard page 1 | Cheveaux fiscaux | Obligatoire |
| **Un contrôle de cohérence doit être effectué (combox box lié)** | | |
| Wizard page 2 | Date du permis | Obligatoire |
| Wizard page 2 | Nombre d’accident | Obligatoire |
| Wizard page 2 | Bonus malus | Obligatoire et a valider |
|  | | |
| Wizard page 3 | Conducteur principal | Obligatoire |
| Wizard page 3 | Conducteur secondaire | Juste nom et prénom si différent de l’utilisateur connecté |
| Wizard page 3 | Dors dans un garage ? | Obligatoire |
| Wizard page 3 | Adresse | Obligatoire |
|  | | |
| Wizard page 4 | Résumé du devis |  |
| Wizard page 4 | Tous risques ? | Obligatoire |
| Wizard page 4 | Au tiers | Obligatoire |
| Wizard page 4 | Montant du devis |  |

**Description de l’écran des devis habitations**

|  |  |  |
| --- | --- | --- |
| Wizard page 1 | Nom / prenom de l’utilisateur | Non modifiable |
| Wizard page 1 | Nom du devis | Obligatoire |
| Wizard page 1 | Type d’habitation | Obligatoire |
| Wizard page 1 | Surface | Obligatoire |
|  | | |
| Wizard page 2 | Nombre de pièces | Obligatoire |
| Wizard page 2 | Etage (si appartement) | Obligatoire |
| Wizard page 2 | Nombre de salle de bain | Obligatoire |
| Wizard page 2 | Garage ? | Obligatoire |
|  | | |
| Wizard page 3 | Surface du terrain | Obligatoire si il y a |
| Wizard page 3 | Surface de la terrasse | Obligatoire si il y a |
| Wizard page 3 | Type de chauffage | Obligatoire |
|  | | |
| Wizard page 4 | Résumé |  |
| Wizard page 4 | Formule 1 | Obligatoire |
| Wizard page 4 | Formule 2 | Obligatoire |
| Wizard page 4 | Prix |  |

**Ecran d’administration (liste en temps réel)**

On doit pouvoir voir au travers d’une liste les utilisateurs qui sont en train de réaliser un devis, cette liste doit mentionner l’utilisateur, le nom ainsi que le type du devis et l’étape dans laquelle il se trouve.

Cette liste étant en temps réelle, lorsqu’un utilisateur commence un devis ou en finis un, la liste doit se mettre à jour sans rechargement de page.

Bonus secondaire, au clic sur un nom de la liste, on doit pouvoir envoyer un message en temps réel à cette utilisateur, ceci sous la forme d’une fenêtre non modale en bas de l’écran de l’utilisateur.

Afin de réaliser la liste d’admin, vous avez le droit d’utiliser un autre framework UI (Angular ou autre par exemple). Ce framework doit s’intégrer dans l’environnement des applications (sécurité, etc …)

Astuce : La communication en temps réelle entre un system back office et front office peut se faire via le bus de Vert.x (il existe un connecteur SockJs entre du javascript et un bus Vert.X).

Il existe d’autres solutions, vous avez toute liberté pour le faire.