

Rapport Individuel

SAE R 3.01c

Semaine 3

Ilane Riotte

Synthèse :

Cette troisième et dernière semaine de projet MobiList a été consacrée à la finalisation des fonctionnalités manquantes, à la correction des derniers bugs ergonomiques et, surtout, à l'implémentation de la partie "Assureur" de l'application.

Mon rôle est resté principalement technique et transversal. J'ai veillé à ce que les fondations posées en semaine 2 (le "Profile Pattern") soient capables de supporter un deuxième type d'utilisateur. Mes contributions majeures ont été :

- Gestion du multi-profil : Adaptation des vues de connexion et de redirection pour différencier un compte "Assuré" d'un compte "Assureur".
- Développement des vues Assureur : Création et correction de la logique permettant à un assureur de consulter les dossiers et sinistres des clients sans pouvoir les modifier.
- Optimisation de l'expérience utilisateur : Intégration de fonctionnalités de confort comme l'affichage/masquage des mots de passe (JavaScript) et l'harmonisation des styles CSS (boutons, messages flash, formulaires).
- Maintenance et Debugging : Résolution des erreurs de logique persistantes sur le changement de mot de passe et le lien entre les pièces et les logements.
- Soutien technique : Assistance continue auprès des membres du groupe pour intégrer leurs templates et corriger les problèmes de routage Flask.

Analyse :

Le défi majeur de cette semaine a été la gestion de la dualité des profils utilisateurs. Le système devait être capable de rediriger correctement l'utilisateur après le login selon son rôle et d'afficher des interfaces radicalement différentes (gestion d'inventaire pour l'un, consultation de sinistres pour l'autre).

Problème rencontré : Lors de l'ajout du profil "Assureur", nous avons rencontré des erreurs de type AttributeError car certaines vues communes essayaient d'accéder à current_user.assure_profile alors que l'utilisateur connecté était un assureur (possédant un assureur_profile). L'application plantait dès qu'un assureur tentait d'accéder aux paramètres ou au tableau de bord.

Démarche et Solution :

- **Logique de branchement :** J'ai modifié les fonctions de vue (notamment login et paramètres) pour inclure des tests de type if current_user.assure_profile:. Cela permet d'adapter dynamiquement le contenu du template ou la destination de la redirection.
- **Sécurisation des accès :** J'ai renforcé les vues avec des décorateurs et des vérifications manuelles pour empêcher un assuré d'accéder aux pages réservées aux assureurs, et vice versa.
- **Uniformisation :** Pour faciliter la maintenance, j'ai centralisé le style des formulaires dans des fichiers CSS dédiés (modification_mdp.css), permettant une cohérence visuelle parfaite entre les pages créées par différents membres du groupe.

Démonstration de compétence :

AC21.01 | Élaborer et implémenter les spécifications fonctionnelles : J'ai finalisé l'implémentation du changement de mot de passe sécurisé et mis en place le système de consultation pour les assureurs, répondant ainsi aux exigences de gestion des sinistres par des tiers professionnels.

AC21.03 | Adopter de bonnes pratiques de conception et de programmation : J'ai optimisé l'utilisation des macros Jinja2 (_formhelpers.html) pour y intégrer des fonctionnalités JavaScript (toggle mot de passe), évitant ainsi la duplication de code sur plusieurs pages.

AC22.01 | Choisir des structures de données complexes adaptées au problème : J'ai géré la complexité du modèle User lié à deux tables de profils distinctes (Assure et Assureur), garantissant que les relations SQLAlchemy restent intègres même lors de modifications de mot de passe simultanées sur les deux profils.

AC23.01 | Concevoir et développer des applications communicantes : J'ai assuré la gestion des flux de redirection HTTP complexes, s'assurant que chaque type d'utilisateur reste dans son environnement dédié (Tableau de bord assuré vs Interface assureur).

AC25.02 | Formaliser les besoins du client et de l'utilisateur : En ajoutant des retours visuels (messages flash de succès/erreur stylisés) et des aides à la saisie (indicateurs de longueur de mot de passe), j'ai traduit les besoins de clarté de l'utilisateur final en solutions techniques concrètes.

AC26.03 | Mobiliser les compétences interpersonnelles pour travailler dans une équipe : J'ai servi de "point de contrôle technique" pour le groupe, en reprenant les styles CSS et les structures HTML de mes camarades pour les rendre compatibles avec le back-end que j'avais stabilisé.