Stratégie de tests

Table des matières

Scénarios prévus

Exigence 1 : Gestion des comptes utilisateurs

Description : Permet la création, modification, suppression, et gestion des informations des comptes utilisateurs.

Scénarios :

Scénario 1 *: Création d'un compte utilisateur*

Objectif : Vérifier que l'utilisateur peut créer un compte avec les informations requises et que les contrôles sont bien appliqués.
Exigences associées : TOM-1.1, TOM-1.4, TOM-6.1, TOM-6.2, TOM-6.3, TOM-6.4, TOM-6.5, TOM-6.6, TOM-6.7

Étapes	Action	Résultat attendu
1	Aller sur la page d'inscription	La page s'affiche correctement
2	Saisir un email déjà utilisé	Un message d'erreur indiquant que l'email est déjà pris est affiché
3	Saisir un email valide et un mot de passe conforme aux règles de sécurité	Le formulaire accepte les entrées
4	Saisir un prénom contenant des chiffres (ex. : "Jean123")	Un message d'erreur s'affiche : "Le prénom ne doit contenir que des lettres"
5	Saisir une date de naissance dans un format invalide (ex. : 01-01-2000)	Un message d'erreur s'affiche : "Format invalide. Utilisez JJ/MM/AAAA"
6	Saisir une date de naissance après 2007	Un message d'erreur s'affiche : "Vous devez avoir 18 ans pour vous inscrire"
7	Ne remplir que certains champs (laisser un ou plusieurs champs obligatoires vides)	Un message d'erreur s'affiche : "Veuillez remplir tous les champs obligatoires"
8	Saisir tous les champs avec des données valide (notamment une date de naissance valide (ex. : 01/01/2000))	Le formulaire accepte la date
9	Valider l'inscription	Un email de confirmation est envoyé
10	Cliquer sur le lien de confirmation dans l'email	Le compte est activé et l'utilisateur peut se connecter

Scénario 2 : Évolution automatique de la date limite de naissance

Objectif : Vérifier si la limite d'âge évolue automatiquement chaque année.
Exigences associées : TOM-6.2

Étapes	Action	Résultat attendu
1	Se connecter avec un compte existant	L'utilisateur accède à son espace personnel
2	Aller dans les paramètres du compte	La page des paramètres s'affiche
3	Modifier le prénom et enregistrer	Le changement est sauvegardé et visible après rafraîchissement
4	Modifier l'email et enregistrer	Un email de validation est envoyé pour confirmer le changement
5	Modifier la date de naissance	Un message d'erreur s'affiche indiquant que ce champ ne peut pas être modifié

Scénario 3 : Modification des informations d'un compte

Objectif : Vérifier si un utilisateur peut modifier certaines informations après la création de son compte.
Exigences associées : TOM-1.2

Étapes	Action	Résultat attendu
1	Se connecter avec un compte existant	L'utilisateur accède à son espace personnel
2	Aller dans les paramètres du compte	La page des paramètres s'affiche
3	Modifier le prénom et enregistrer	Le changement est sauvegardé et visible après rafraîchissement
4	Modifier l'email et enregistrer	Un email de validation est envoyé pour confirmer le changement
5	Modifier la date de naissance	Un message d'erreur s'affiche indiquant que ce champ ne peut pas être modifié

Scénario 4 : Suppression d'un compte utilisateur

Objectif : Vérifier que la suppression d'un compte est bien prise en charge et qu'elle suit les règles définies.
Exigences associées : TOM-1.3

Étapes	Action	Résultat attendu
1	Se connecter avec un compte existant	L'utilisateur accède à son espace personnel
2	Aller dans les paramètres et cliquer sur "Supprimer mon compte"	Une boîte de dialogue de confirmation s'affiche
3	Confirmer la suppression	Le compte est désactivé et l'utilisateur est déconnecté
4	Tenter de se reconnecter avec les identifiants supprimés	Un message d'erreur indique que le compte n'existe plus

Scénario 5 : Téléversement d'un justificatif

Objectif : Vérifier que l'utilisateur peut téléverser un justificatif dans les formats et limites autorisés.
Exigences associées : TOM-7.1, TOM-7.2, TOM-7.4

Étapes	Action	Résultat attendu
1	Aller sur la page de téléversement	La page s'affiche correctement
2	Sélectionner un fichier au format PNG	Le fichier est accepté
3	Sélectionner un fichier au format non supporté (ex : .exe)	Un message d'erreur s'affiche indiquant que le format n'est pas accepté
4	Sélectionner un fichier de 6 Mo (si la limite est 5 Mo)	Un message d'erreur indique que la taille est trop grande
5	Téléverser un fichier valide	Le fichier est ajouté avec succès

Scénario 6 : Vérification du consentement et du stockage des données

Objectif : Vérifier que les données personnelles sont stockées avec le consentement explicite de l'utilisateur et qu'elles ne sont pas conservées inutilement.

Exigences associées : TOM-1.5

Étapes	Action	Résultat attendu
1	Aller sur la page d'inscription	La page affiche une case à cocher "J'accepte la politique de confidentialité"
2	Ne pas cocher la case	Un message d'erreur empêche la soumission du formulaire
3	Cocher la case et compléter le formulaire	L'utilisateur peut soumettre le formulaire
4	Valider l'inscription	Les données sont enregistrées en base, un horodatage du consentement est conservé
5	Vérifier en base ou via API	Les données sont bien présentes, le consentement est enregistré (timestamp ou champ associé)
6	Vérifier la documentation RGPD	Une politique de conservation des données est clairement définie et accessible

Scénario 7 : Suppression automatique des données après inactivité (durée de conservation)

Objectif : Vérifier que les données utilisateur sont supprimées ou anonymisées après la durée légale de conservation (ex. : 3 ans d'inactivité).

Exigences associées : TOM-1.5

Étapes	Action	Résultat attendu
1	Créer un compte utilisateur	Le compte est actif et stocké en base
2	Simuler 3 ans d'inactivité (modification manuelle de la date dernière activité en base ou via un script)	La date est modifiée sans interaction de l'utilisateur
3	Lancer manuellement (ou attendre) la tâche de nettoyage	Une tâche automatique ou planifiée supprime/anonymise le compte
4	Vérifier la base	Les données ne sont plus présentes ou sont rendues anonymes (ex : "utilisateur supprimé")
5	Essayer de se connecter	L'accès au compte n'est plus possible
6	Vérifier les logs	Une trace de suppression (ou anonymisation) est visible dans les journaux du système ou du traitement automatisé

Exigence 2 : Authentification et sécurité

Description : Garantir que l'utilisateur puisse s'authentifier en toute sécurité, y compris l'authentification biométrique et à double facteur.

Scénarios :

Scénario 8 : Authentification avec double facteur

Objectif : Vérifier que l'authentification à double facteur fonctionne correctement et que les erreurs sont bien gérées.
Exigences associées : TOM-8.1, TOM-8.3, TOM-8.4

Étapes	Action	Résultat attendu
1	Se connecter avec un compte ayant le double facteur activé	Une demande de validation par code est affichée
2	Saisir un code correct	L'utilisateur est connecté avec succès
3	Saisir un code incorrect trois fois	Un message d'erreur indique que la tentative a échoué
4	Cliquer sur "Renvoyer le code"	Un nouveau code est envoyé par SMS/email
5	Ne pas recevoir le code et demander de l'aide	Une option de récupération est proposée

Scénario 9 : Activation de l'authentification biométrique

Objectif : Vérifier que l'utilisateur peut activer l'authentification biométrique et l'utiliser pour se connecter.
Exigences associées : TOM-9.1, TOM-9.2, TOM-9.4

Étapes	Action	Résultat attendu
1	Aller dans les paramètres de sécurité du compte	La page des paramètres s'affiche correctement
2	Activer l'option "Authentification biométrique"	Le système détecte les capteurs biométriques disponibles
3	S'authentifier avec le mot de passe pour valider l'activation	La biométrie est activée avec un message de confirmation
4	Se déconnecter et tenter de se reconnecter avec la biométrie	L'accès est accordé via l'authentification biométrique

Scénario 10 : Échec de l'authentification biométrique

Objectif : Vérifier la gestion des erreurs lorsque l'authentification biométrique échoue.
Exigences associées : TOM-9.3

Étapes	Action	Résultat attendu
1	Tenter de se connecter en utilisant une empreinte non enregistrée	Un message d'erreur s'affiche, demandant de réessayer ou d'utiliser un autre moyen
2	Réessayer avec la bonne empreinte	L'accès est accordé
3	Échouer l'authentification biométrique trois fois de suite	Un message propose de s'authentifier avec le mot de passe

Scénario 11 : Désactivation de l'authentification biométrique

Objectif : Vérifier que l'utilisateur peut désactiver la biométrie et revenir à une connexion classique.
Exigences associées : TOM-9.6

Étapes	Action	Résultat attendu
1	Aller dans les paramètres de sécurité	La page des paramètres s'affiche correctement
2	Désactiver l'option "Authentification biométrique"	Un message de confirmation est affiché
3	Se déconnecter et tenter de se reconnecter avec la biométrie	La connexion biométrique est désactivée, et un mot de passe est demandé

Scénario 12 : Gestion des données biométriques

Objectif : Vérifier que les données biométriques sont stockées de manière sécurisée et ne peuvent pas être compromises.
Exigences associées : TOM-9.5

Étapes	Action	Résultat attendu
1	Activer l'authentification biométrique	La biométrie est activée avec succès
2	Vérifier le stockage des données biométriques	Les données ne sont pas accessibles en clair et sont sécurisées selon les normes en vigueur
3	Désactiver l'authentification biométrique et tenter d'accéder aux données biométriques stockées	Les données sont supprimées ou restent inaccessibles

Scénario 13 : Sécurisation et gestion des justificatifs

Objectif : Vérifier la sécurité et la gestion des justificatifs après leur téléversement.

Exigences associées : TOM-7.3, TOM-7.5, TOM-7.6, TOM-7.7

Étapes	Action	Résultat attendu
1	Téléverser un justificatif	Le fichier est accepté
2	Vérifier si une analyse automatique est réalisée (OCR, métadonnées)	Le système extrait les informations si applicable
3	Vérifier le stockage du fichier	Le fichier est chiffré et stocké de manière sécurisée
4	Supprimer le justificatif	L'utilisateur ne peut plus y accéder
5	Vérifier s'il existe une date d'expiration sur le fichier	Si oui, un message d'alerte doit informer l'utilisateur avant expiration

Scénario 14 : Configuration et sécurisation du double facteur

Objectif : Vérifier si le double facteur peut être désactivé et comment se passe la récupération en cas de perte.
Exigences associées : TOM-8.2, TOM-8.5, TOM-8.6

Étapes	Action	Résultat attendu
1	Activer l'authentification à double facteur	L'activation est confirmée
2	Se connecter et tenter de désactiver le double facteur	Vérifier si l'option est disponible et si une validation supplémentaire est requise
3	Perdre l'accès au moyen de validation (ex : téléphone perdu)	Un processus de récupération est proposé (email, questions de sécurité...)
4	Saisir un mauvais code plusieurs fois	Après un certain nombre d'échecs, le compte est temporairement bloqué

Exigence 3 : Connexion sécurisée et intégration bancaire

Description : Garantir la sécurité des échanges de données entre l'application et les services bancaires partenaires. S'assurer de la fiabilité de la connexion aux API externes (ex. : API BDF) ainsi que de la continuité de service en cas d'interruption.

Scénarios :

Scénario 15 : *Connexion et sécurisation des échanges avec les banques*

Objectif : Vérifier que l'utilisateur peut se connecter à une banque compatible et que les échanges sont sécurisés.
Exigences associées : TOM-2.1, TOM-2.2, TOM-2.3

Étapes	Action	Résultat attendu
1	Accéder à la liste des banques compatibles	La liste des banques supportées s'affiche correctement
2	Sélectionner une banque et tenter de s'y connecter sans être authentifié	Un message d'erreur demande une authentification préalable
3	Se connecter à l'application et sélectionner une banque	La connexion à la banque est possible
4	Capturer les échanges réseau lors de l'interconnexion	Les données sont chiffrées (HTTPS, OAuth, etc.)
5	Vérifier les logs pour s'assurer qu'aucune donnée sensible n'est enregistrée	Les logs ne contiennent pas d'informations personnelles en clair

Scénario 16 : *Gestion des interruptions et récupération des données*

Objectif : Vérifier la gestion des interruptions bancaires et la récupération des transactions.
Exigences associées : TOM-2.4, TOM-2.5, TOM-2.6

Étapes	Action	Résultat attendu
1	Simuler une coupure de connexion avec une banque	Un message d'erreur s'affiche
2	Attendre quelques secondes pour observer une reconnexion automatique	Une tentative de reconnexion est effectuée
3	Vérifier si l'historique des transactions est accessible	Les transactions récentes et passées sont affichées selon la banque
4	Tester la mise à jour automatique et manuelle des données bancaires	Les données sont rafraîchies selon la méthode prévue (automatique ou manuelle)

Scénario 17 : Connexion à l'API BDF et gestion des erreurs

Objectif : Vérifier l'authentification et la gestion des erreurs lors de l'utilisation de l'API BDF.
Exigences associées : TOM-10.1, TOM-10.2, TOM-10.3

Étapes	Action	Résultat attendu
1	Tenter de se connecter à l'API BDF sans authentification spécifique	Un message d'erreur s'affiche
2	Se connecter avec une authentification correcte	L'accès est accordé
3	Simuler une latence de 30 secondes de l'API	Un message d'erreur apparaît et un mécanisme de gestion est déclenché
4	Effectuer plusieurs appels en moins de 5 minutes	Un blocage temporaire ou un message d'erreur s'affiche si une limite est atteinte

Scénario 18 : Gestion des données bancaires et conservation des informations

Objectif : Vérifier la gestion des données bancaires stockées et affichées.
Exigences associées : TOM-10.4, TOM-10.5, TOM-11.1, TOM-11.2, TOM-11.3, TOM-11.4, TOM-11.5

Étapes	Action	Résultat attendu
1	Vérifier si les données bancaires sont conservées en base selon une politique définie	Une durée de conservation est appliquée
2	Consulter les anciennes données après un échec de mise à jour de l'API BDF	Les données précédentes restent accessibles
3	Ajouter plusieurs banques et vérifier l'affichage du tableau de bord	Le nombre de banques affichées est limité si nécessaire
4	Ajouter plusieurs comptes d'une même banque	Tous les comptes sont bien affichés sans erreur
5	Ajouter un grand nombre de comptes et vérifier la pagination	L'affichage est paginé si besoin
6	Simuler une banque temporairement indisponible	Un message ou une icône indique que les données sont inaccessibles
7	Vérifier si l'utilisateur peut personnaliser l'affichage des comptes	Tri, filtres et mise en avant d'un compte sont possibles

Scénario 19 : Téléchargement et gestion des relevés bancaires

Objectif : Vérifier le bon fonctionnement du téléchargement des relevés et la traçabilité des actions.
Exigences associées : TOM-12.1, TOM-12.2, TOM-12.3, TOM-12.4, TOM-12.5, TOM-12.6

Étapes	Action	Résultat attendu
1	Tenter de télécharger plusieurs relevés en même temps	Le téléchargement est autorisé ou un message précise une limite
2	Essayer de télécharger un relevé indisponible	Un message d'erreur informe de l'indisponibilité
3	Télécharger un relevé sur différents systèmes (Windows, Mac, Android, iOS)	Le téléchargement fonctionne sur tous les OS supportés
4	Vérifier le nom du fichier téléchargé	Le fichier suit le format standard (ex. banque_nom_date.pdf)
5	Vérifier si une confirmation s'affiche après un téléchargement réussi	Un message de succès apparaît
6	Consulter l'historique des téléchargements	Un journal des relevés téléchargés est disponible pour l'utilisateur

Exigence 4 : Affichage et traitement des transactions

Description : Offrir à l'utilisateur un affichage clair, fiable et actualisé de ses transactions. Gérer les erreurs liées aux appels API et assurer une bonne expérience utilisateur, y compris en cas de forte charge ou de regroupement d'informations.

Scénarios :

Scénario 20 : Gestion de l'affichage des transactions

Objectif : Vérifier que les transactions sont bien paginées, triées et filtrées.

Exigences associées : TOM-3.1, TOM-3.2, TOM-3.3

Étapes	Action	Résultat attendu
1	Accéder à la liste des transactions	Les transactions s'affichent correctement
2	Vérifier la pagination si les transactions sont nombreuses	La liste est paginée
3	Appliquer un tri par montant, date et destinataire	Les transactions sont triées correctement
4	Rechercher une transaction par mot-clé ou utiliser un filtre	Les transactions correspondantes sont affichées

Scénario 21 : Gestion des erreurs API

Objectif : Vérifier le comportement du système en cas d'échec de l'API BDF.

Exigences associées : TOM-3.4

Étapes	Action	Résultat attendu
1	Simuler une panne de l'API BDF	Un message d'erreur est affiché
2	Vérifier le comportement si l'API renvoie une erreur	Une alerte informe l'utilisateur
3	Vérifier si une solution de repli est proposée	Une alternative est disponible (ex: affichage des dernières données connues)

Scénario 22 : Gestion de l'historique des consultations

Objectif : Vérifier que l'utilisateur peut consulter son historique.

Exigences associées : TOM-3.5

Étapes	Action	Résultat attendu
1	Accéder à l'historique des consultations	L'historique s'affiche correctement
2	Vérifier la conservation des consultations passées	Les données sont bien enregistrées

Scénario 23 : Rafraîchissement des transactions

Objectif : Vérifier si les transactions sont mises à jour en temps réel ou à intervalle régulier.

Exigences associées : TOM-3.6

Étapes	Action	Résultat attendu
1	Vérifier si une mise à jour automatique se déclenche	Les transactions sont rafraîchies selon la fréquence définie
2	Forcer une mise à jour manuelle	La liste est mise à jour immédiatement

Scénario 24 : Stockage et surcharge de l'API

Objectif : Vérifier la gestion du stockage des transactions et la surcharge de l'API.

Exigences associées : TOM-13.1, TOM-13.2

Étapes	Action	Résultat attendu
1	Vérifier si les transactions sont stockées en BDD	La réponse du système est cohérente avec la décision prise
2	Simuler plusieurs utilisateurs consultant en même temps	L'API gère correctement la surcharge

Scénario 25 : Expérience utilisateur et affichage des transactions

Objectif : Vérifier l'UX en cas de latence et la présence d'un indicateur de chargement.

Exigences associées : TOM-13.3, TOM-13.4

Étapes	Action	Résultat attendu
1	Simuler une latence de l'affichage des transactions	Un indicateur de chargement est visible
2	Vérifier l'affichage progressif des données	Les données sont chargées progressivement sans bloquer l'interface

Scénario 26 : Regroupement et alerte des transactions

Objectif : Vérifier le regroupement par type de dépense et la détection des transactions suspectes.

Exigences associées : TOM-13.5, TOM-13.6

Étapes	Action	Résultat attendu
1	Vérifier si les transactions sont bien regroupées par type	Les groupes sont corrects
2	Simuler une transaction suspecte	Une alerte est affichée

Exigence 5 : Suivi, analyse et visualisation des données

Description : Permettre à l'utilisateur de suivre sa consommation et d'analyser ses données via des représentations graphiques compréhensibles. Inclure des fonctionnalités d'export et assurer la compatibilité avec différents formats.

Scénarios :

Scénario 27 : Analyse et affichage du suivi de consommation

Objectif : Vérifier la gestion et l'affichage des dépenses.

Exigences associées : TOM-4.1, TOM-4.2, TOM-4.3, TOM-4.4, TOM-4.6

Étapes	Action	Résultat attendu
1	Vérifier le regroupement des dépenses par catégorie	Le regroupement est correct
2	Sélectionner une période personnalisée	L'analyse se met à jour
3	Vérifier la mise à jour en temps réel	Les données sont rafraîchies selon la fréquence définie
4	Afficher un grand nombre de dépenses	L'affichage est optimisé sans surcharge
5	Comparer avec une période précédente	La comparaison s'affiche correctement

Scénario 28 : Export et compatibilité

Objectif : Vérifier l'export des données et la gestion des devises.

Exigences associées : TOM-4.5, TOM-14.2

Étapes	Action	Résultat attendu
1	Exporter les données en CSV et PDF	L'export est fonctionnel
2	Vérifier la prise en compte des devises	Les taux de conversion sont appliqués

Scénario 29 : Gestion des requêtes simultanées et affichage des graphiques

Objectif : Vérifier la gestion des requêtes et les formats d'affichage.

Exigences associées : TOM-14.1, TOM-14.3, TOM-14.4, TOM-14.5

Étapes	Action	Résultat attendu
1	Simuler plusieurs requêtes simultanées	Le système gère correctement la charge
2	Filtrer par catégorie	Les filtres fonctionnent
3	Changer le format du graphique	L'affichage s'adapte
4	Vérifier le comportement si une banque ne fournit pas ses données	Une gestion d'erreur est mise en place

Exigence 6 : Recommandations et accompagnement utilisateur

Description : Fournir une assistance personnalisée à l'utilisateur via un conseiller virtuel. Adapter les recommandations en fonction des données collectées dans le respect de la confidentialité et des préférences de l'utilisateur.

Scénarios :

Scénario 30 : Fonctionnement et personnalisation du conseiller virtuel

Objectif : Vérifier la disponibilité et le fonctionnement du conseiller virtuel.

Exigences associées : TOM-5.1, TOM-5.2, TOM-5.3, TOM-5.4, TOM-5.5

Étapes	Action	Résultat attendu
1	Tester sur mobile et desktop	Le conseiller fonctionne sur les plateformes prévues
2	Simuler une charge élevée	Le conseiller reste disponible
3	Vérifier les réponses automatiques et la redirection	Le comportement est conforme
4	Tester la personnalisation des conseils	L'utilisateur reçoit des recommandations adaptées
5	Désactiver la bulle du conseiller	L'option fonctionne

Scénario 31 : Gestion des recommandations et des données collectées

Objectif : Vérifier les règles de recommandation et la collecte des données.

Exigences associées : TOM-5.6, TOM-15.1, TOM-15.2, TOM-15.3, TOM-15.4, TOM-15.5

Étapes	Action	Résultat attendu
1	Vérifier les règles de recommandation	Les conseils sont cohérents
2	Tester l'adaptation en fonction des finances	Les conseils évoluent
3	Refuser une recommandation	L'option fonctionne
4	Tester la prise de rendez-vous	La procédure est claire
5	Vérifier la mise à jour des conseils	L'adaptation est dynamique

Méthodes de test adaptées

Comme on peut le constater, certains scénarios de test se retrouvent dans plusieurs types de tests.

Cela s'explique par la nature de notre projet : beaucoup de fonctionnalités critiques doivent à la fois être validées du point de vue fonctionnel (tests boîte noire), maintenues intactes après des évolutions (tests de non-régression) et exécutées régulièrement pour sécuriser les futures versions (tests automatisés).

Cette approche permet de garantir à la fois la qualité fonctionnelle immédiate, la stabilité sur le long terme et l'efficacité des campagnes de tests.

C'est pourquoi certains scénarios, essentiels pour l'expérience utilisateur ou la conformité réglementaire (comme la création de compte, la sécurité d'authentification ou la gestion des données personnelles), ont été associés à plusieurs méthodes de test complémentaires.

Tests boîte noire

Les tests boîte noire sont des tests fonctionnels visant à vérifier que le système se comporte comme attendu sans avoir connaissance de son fonctionnement interne (code, structure, algorithme). On se concentre uniquement sur les entrées et les sorties. Cette approche se place du point de vue de l'utilisateur final.

Scénarios concernés

- **Scénario 1 : Création d'un compte utilisateur** → Vérification fonctionnelle de la création avec différents cas d'entrées utilisateurs.
- **Scénario 2 : Évolution automatique de la date limite de naissance** → Validation du comportement dynamique de la limite d'âge sans accès au code.
- **Scénario 3 : Modification des informations d'un compte** → Vérification fonctionnelle des modifications décrites depuis l'interface utilisateur.
- **Scénario 4 : Suppression d'un compte utilisateur** → Contrôle de la suppression d'un compte via les actions utilisateur.
- **Scénario 5 : Téléversement d'un justificatif** → Validation des contrôles de format et de taille à partir de l'interface.
- **Scénario 6 : Vérification du consentement et stockage des données** → Validation de l'obtention du consentement RGPD à l'inscription.
- **Scénario 7 : Suppression automatique des données après inactivité** → Vérification du respect des durées de conservation sans accès à l'intérieur du traitement.
- **Scénario 8 : Authentification avec double facteur** → Validation du processus 2FA du point de vue utilisateur.
- **Scénario 9 : Activation de l'authentification biométrique** → Contrôle du bon fonctionnement de l'activation biométrique.
- **Scénario 10 : Échec de l'authentification biométrique** → Vérification de la gestion des erreurs d'authentification biométrique.
- **Scénario 11 : Désactivation de l'authentification biométrique** → Contrôle de la désactivation et retour au mot de passe.
- **Scénario 12 : Gestion des données biométriques** → Contrôle de la sécurisation des données biométriques.

- **Scénario 13 : Sécurisation et gestion des justificatifs** → Vérification du stockage sécurisé des fichiers téléversés.
- **Scénario 14 : Configuration et sécurisation du double facteur** → Validation de la configuration du 2FA et de sa récupération.
- **Scénario 15 : Connexion et sécurisation des échanges avec les banques** → Vérification fonctionnelle des échanges API bancaires.
- **Scénario 16 : Gestion des interruptions et récupération des données** → Test fonctionnel des reconnexions et sauvegardes de données.
- **Scénario 17 : Connexion à l'API BDF et gestion des erreurs** → Validation du comportement de l'authentification API.
- **Scénario 18 : Gestion des données bancaires et conservation des informations** → Contrôle du stockage et de l'affichage des données bancaires.
- **Scénario 19 : Téléchargement et gestion des relevés bancaires** → Vérification du téléchargement et de l'historique.
- **Scénario 20 : Gestion de l'affichage des transactions** → Vérification de la pagination, tri, et recherche de transactions.
- **Scénario 21 : Gestion des erreurs API** → Test fonctionnel de l'affichage d'erreurs API.
- **Scénario 22 : Gestion de l'historique des consultations** → Contrôle du bon stockage et affichage des consultations.
- **Scénario 23 : Rafraîchissement des transactions** → Test du rafraîchissement manuel et automatique.
- **Scénario 25 : Expérience utilisateur et affichage des transactions** → Validation de l'affichage sous latence.
- **Scénario 26 : Regroupement et alerte des transactions** → Vérification du regroupement par type et détection d'anomalies.
- **Scénario 27 : Analyse et affichage du suivi de consommation** → Test de l'affichage et de la comparaison des dépenses.
- **Scénario 28 : Export et compatibilité** → Vérification des exports CSV/PDF et de la gestion des devises.
- **Scénario 29 : Gestion des requêtes simultanées et affichage des graphiques** → Contrôle de l'affichage sous forte charge.
- **Scénario 30 : Fonctionnement et personnalisation du conseiller virtuel** → Test de la disponibilité et de la personnalisation des conseils.
- **Scénario 31 : Gestion des recommandations et des données collectées** → Validation des règles d'adaptation des recommandations.

Tests exploratoires

Les tests exploratoires reposent sur la créativité et l'intuition du testeur. Ils ne suivent pas de script détaillé, mais visent à découvrir des anomalies en explorant librement l'application. Ils complètent les tests formels en cherchant les comportements inattendus.

Scénarios concernés

- **Scénario 2 : Évolution automatique de la date limite de naissance** → Exploration du calcul dynamique de l'âge.
- **Scénario 25 : Expérience utilisateur et affichage des transactions** → Détection d'anomalies d'affichage sous différentes latences.
- **Scénario 29 : Gestion des requêtes simultanées et affichage des graphiques** → Découverte d'éventuels problèmes d'affichage ou de lenteur en forte charge.
- **Scénario 30 : Fonctionnement et personnalisation du conseiller virtuel** → Vérification libre de la personnalisation et adaptabilité.

Tests de non-régression

Les tests de non-régression visent à s'assurer que les fonctionnalités existantes continuent à fonctionner correctement après des modifications du système. Ils permettent d'éviter que de nouvelles erreurs soient introduites involontairement.

Scénarios concernés

- **Scénario 1 : Création d'un compte utilisateur** → S'assurer que la création reste fonctionnelle après évolutions.
- **Scénario 3 : Modification des informations d'un compte** → Contrôle de la persistance des modifications utilisateur.
- **Scénario 4 : Suppression d'un compte utilisateur** → S'assurer de la suppression correcte malgré les changements.
- **Scénario 7 : Suppression automatique des données après inactivité** → Validation de la récurrence du nettoyage automatique.
- **Scénario 8 : Authentification avec double facteur** → Maintenir le bon fonctionnement du 2FA.
- **Scénario 9 : Activation de l'authentification biométrique** → Contrôler la continuité de la fonctionnalité biométrique.
- **Scénario 10 : Échec de l'authentification biométrique** → Maintenir la gestion des erreurs biométriques.
- **Scénario 11 : Désactivation de l'authentification biométrique** → Contrôle de la désactivation sans régression.
- **Scénario 15 : Connexion et sécurisation des échanges avec les banques** → Continuité de la sécurisation des échanges API.
- **Scénario 16 : Gestion des interruptions et récupération des données** → S'assurer de la récupération des données en cas d'erreurs.
- **Scénario 17 : Connexion à l'API BDF et gestion des erreurs** → Maintenir les traitements d'erreurs de connexion.
- **Scénario 18 : Gestion des données bancaires et conservation des informations** → S'assurer de la conservation conforme des données bancaires.
- **Scénario 23 : Rafraîchissement des transactions** → Continuité du rafraîchissement automatique et manuel.

- **Scénario 24 : Stockage et surcharge de l'API** → Vérification que la surcharge est toujours gérée correctement.

Tests automatisés

Les tests automatisés permettent de réaliser rapidement et régulièrement des tests répétitifs, notamment pour les parcours critiques, afin de gagner du temps et garantir la stabilité du système.

Scénarios concernés

- **Scénario 1 : Création d'un compte utilisateur** → Automatiser la création pour tester rapidement divers cas d'entrées.
- **Scénario 7 : Suppression automatique des données après inactivité** → Simuler et valider les suppressions sur le long terme.
- **Scénario 8 : Authentification avec double facteur** → Automatiser les vérifications de sécurité 2FA.
- **Scénario 9 : Activation de l'authentification biométrique** → Automatiser les activations et connexions biométriques.
- **Scénario 15 : Connexion et sécurisation des échanges avec les banques** → Tester automatiquement les différentes banques.
- **Scénario 16 : Gestion des interruptions et récupération des données** → Simuler des interruptions répétées.
- **Scénario 17 : Connexion à l'API BDF et gestion des erreurs** → Automatiser les cas d'erreurs et de timeout.
- **Scénario 19 : Téléchargement et gestion des relevés bancaires** → Répéter les téléchargements pour plusieurs configurations.
- **Scénario 20 : Gestion de l'affichage des transactions** → Automatiser les tests de tri, filtres et pagination.
- **Scénario 21 : Gestion des erreurs API** → Détecter rapidement les problèmes d'erreurs API.
- **Scénario 23 : Rafraîchissement des transactions** → Valider régulièrement la mise à jour automatique.
- **Scénario 25 : Expérience utilisateur et affichage des transactions** → Automatiser le test sous différentes conditions de réseau.
- **Scénario 29 : Gestion des requêtes simultanées et affichage des graphiques** → Simuler des charges élevées pour détecter les ralentissements.

Ressources nécessaires

Pour mener à bien la stratégie de test définie, plusieurs ressources doivent être mobilisées.

Ces ressources seront à la fois humaines, techniques et matérielles, afin de garantir la couverture complète des tests et respecter les objectifs qualité du projet.

Ressources humaines

- **Testeurs fonctionnels :** Les tests manuels (boîte noire) représentent une part importante des validations. Des testeurs ayant une bonne compréhension fonctionnelle de l'application sont nécessaires pour couvrir les parcours utilisateur et les scénarios critiques.
- **Testeurs techniques :** Pour mettre en œuvre les scénarios automatisés (tests de non-régression, validation de charges API), au moins un profil ayant des compétences en scripting de tests automatisés sera nécessaire (par exemple avec Cypress, Postman, ou un autre outil adapté à notre environnement).
- **Compétences attendues :**
  - Connaissance des pratiques de tests fonctionnels (tests boîte noire, tests exploratoires).
  - Maîtrise de bases en SQL pour valider certaines données stockées en base.
  - Connaissance générale des API REST pour interagir et tester les appels bancaires et API BDF.
  - Sensibilité forte aux aspects RGPD (protection et conservation des données personnelles).
  - Compréhension fonctionnelle du secteur bancaire et assurance (transactions, relevés bancaires, sécurisation des flux).

Outils de test

- **Cahier de recette** : Le cahier de recette sera structuré par sprint, en cohérence avec l'organisation agile du projet, pour permettre un suivi précis de l'avancement et une traçabilité complète entre exigences, cas de test et résultats.
- **Gestion des tests** : Un outil de gestion des campagnes de tests est nécessaire pour structurer l'exécution et assurer la traçabilité (par exemple TestRail, Xray, ou une organisation via Jira/Notion sous forme de tableaux Kanban).
- **Automatisation** : Un outil d'automatisation type Cypress pour les tests UI et Postman/Newman pour les tests API sera utilisé pour automatiser les scénarios critiques identifiés dans la stratégie de test.
- **Gestion des anomalies** : L'utilisation d'un outil de bug-tracking (type Jira, GitLab Issues) sera indispensable pour remonter, suivre, et prioriser efficacement les anomalies détectées pendant les campagnes.

Données de test

- **Comptes utilisateurs simulés** : Données fictives nécessaires pour tester l'inscription, la connexion, la suppression et la mise à jour des informations personnelles.
- **Jeux de données bancaires** : Simulations de transactions, historiques de relevés, et identifiants bancaires fictifs seront utilisés pour tester l'interconnexion bancaire et la récupération de données.
- **Données sensibles protégées** : Des jeux de données anonymisées ou fictives seront obligatoires pour respecter les exigences RGPD lors de l'exécution des tests.

Environnements de test

- **Environnement de préproduction** : Un environnement isolé, stable et représentatif de la production sera indispensable pour exécuter les tests sans impacter les utilisateurs réels. Il devra refléter fidèlement l'environnement final.
- **Accès aux environnements partenaires** : Connexion aux environnements de test des partenaires bancaires ou fournisseurs API sera nécessaire (par exemple sandbox bancaire, sandbox BDF). En cas d'indisponibilité de ces partenaires, la mise en place de mocks ou de simulateurs devra être envisagée pour garantir la continuité des tests.
- **Disponibilité des environnements** : Le calendrier d'exécution des tests devra être aligné avec la disponibilité de ces environnements pour éviter les conflits et interruptions.

Contraintes à prendre en compte

- **Complexité du périmètre** : Le projet traite des thématiques sensibles comme la sécurité bancaire et la protection des données personnelles, ce qui exigera une vigilance particulière sur les tests critiques et les parcours sensibles.
- **Dimensionnement de l'équipe** : Selon le nombre de tests pouvant être réalisés en parallèle, le dimensionnement de l'équipe devra être adapté afin d'optimiser les ressources, éviter les périodes d'inactivité et limiter les risques de surcharge.
- **Contraintes budgétaires et temporelles** : Pour respecter les délais et maîtriser les coûts, l'équipe devra prioriser les tests les plus critiques et automatiser dès que possible les scénarios les plus stables et récurrents.

Étapes clés de la stratégie

Afin de garantir une couverture complète des tests tout en respectant les délais fixés pour le projet TOMSEN, une planification structurée autour de jalons clés a été établie.

Elle s'appuie sur l'organisation agile de l'équipe et tient compte des contraintes calendaires spécifiques du mois de mai 2025, marqué par plusieurs jours fériés et ponts potentiels.

Approche de planification

La stratégie repose sur un découpage classique des phases de test, calé sur le rythme des sprints :

Phase de conception :

- Analyse des exigences fonctionnelles du sprint,
- Rédaction des scénarios de test,
- Préparation du cahier de recette.

Phase d'exécution :

- Lancement des tests sur l'environnement de préproduction,
- Suivi des anomalies, retests, et validation des exigences.

Phase de bilan :

- Rédaction d'un PV de recette ou d'un bilan de campagne,
- Analyse des indicateurs de couverture, de conformité et de performance,
- Validation qualité finale du sprint.

Chaque sprint est jalonné d'étapes de conception et d'exécution, avec un point de validation avant le passage au sprint suivant.

La flexibilité est essentielle : la planification est conçue pour pouvoir s'adapter aux aléas projet, aux ajustements de périmètre ou aux indisponibilités d'environnement.

Planning prévisionnel par sprint

Le projet est structuré en six sprints fonctionnels suivis d'un sprint de finalisation. Chaque sprint débute par une phase de développement logiciel menée par l'équipe de développement. En parallèle ou en décalé selon les sprints, les activités de test s'enchaînent : analyse des exigences, conception des tests, puis exécution. Cette organisation permet une synchronisation progressive entre les développements et les validations, tout en maintenant un rythme itératif conforme aux principes agiles.

Le tableau suivant synthétise la planification par sprint avec les phases de test associées.

Sprint	**Période**	**Phases de test et dates associées**
Sprint 1	du 01/04/25 au 22/04/25	Analyse : du 01/04/25 au 03/04/25 Conception : du 08/04/25 au 10/04/25 Exécution : du 16/04/25 au 22/04/25
Sprint 2	du 10/04/25 au 12/05/25	Analyse : du 23/04/25 au 24/04/25 Conception : du 30/04/25 au 05/05/25 Exécution : du 06/05/25 au 12/05/25
Sprint 3	du 05/05/25 au 30/05/25	Analyse : du 13/05/25 au 14/05/25 Conception : du 19/05/25 au 20/05/25 Exécution : du 26/05/25 au 30/05/25
Sprint 4	du 21/05/25 au 13/06/25	Analyse : du 26/05/25 au 27/05/25 Conception : du 02/06/25 au 03/06/25 Exécution : du 10/06/25 au 13/06/25
Sprint 5	du 04/06/25 au 27/06/25	Analyse : du 10/06/25 au 11/06/25 Conception : du 16/06/25 au 18/06/25 Exécution : du 24/06/25 au 27/06/25
Sprint 6 (Finalisation)	du 03/07/25 au 01/08/25	Non-régression : du 03/07/25 au 09/07/25 PV de recette : du 11/07/25 au 15/07/25 Mise en production : le 01/08/25

Jours fériés

- Lundi 21/04/2025 : Lundi de Pâques
- Jeudi 01/05/2025 : Fête du Travail
- *Vendredi 02/05/2025 : Pont probable*
- Jeudi 08/05/2025 : Victoire 1945
- *Vendredi 09/05/2025 : Pont probable*
- Jeudi 29/05/2025 : Ascension
- *Vendredi 30/05/2025 : Pont probable (écoles fermés)*
- Lundi 09/06/2025 : Lundi de Pentecôte
- Lundi 14/07/2025 : Fête nationale

Ces dates ont été prises en compte dans la construction du planning pour éviter toute interruption critique dans les phases de test.

Intégration de marges dans le planning : anticipation et gestion des risques

Le planning prévisionnel intègre volontairement des périodes de latence entre certaines tâches, sous la forme de quelques jours de décalage entre la fin d'une activité et le début de la suivante. Ces temps de respiration sont pensés comme des marges de sécurité, et participent à la robustesse de la stratégie de test.

Ils remplissent plusieurs fonctions essentielles dans une démarche de gestion de projet :

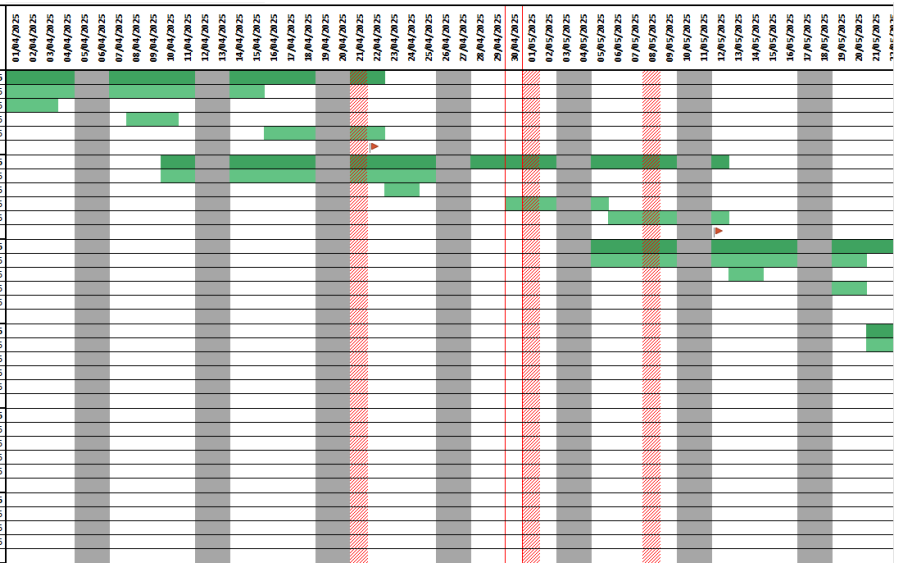
- **Anticiper les aléas** : en cas de retard de développement, de difficultés techniques, ou d'indisponibilité ponctuelle, ces jours permettent d'absorber les imprévus sans mettre en péril l'ensemble du planning.
- **Faciliter la relecture et l'ajustement** : un délai entre la conception des tests et leur exécution, par exemple, permet de valider les cas de test, de finaliser les jeux de données ou de clarifier certaines implémentations avec les développeurs.
- **Laisser place aux itérations** : Ces marges permettent d'intégrer sereinement les retours sur anomalies critiques ou les ajustements de dernière minute, sans décaler les étapes suivantes.
- **Tenir compte des jours fériés** : ces décalages permettent également de lisser l'impact des jours fériés (par exemple, le lundi de Pâques, le 1er mai ou le lundi de Pentecôte) sur le bon déroulement des activités de test.

Ces temps d'intervalle ne sont donc pas des temps perdus, mais bien des éléments clés d'une planification réaliste, souple et adaptée à la complexité d'un projet informatique.

À cela s'ajoute une marge spécifique en fin de projet, pensée comme une véritable soupape de sécurité avant la mise en production.

Cette marge temporelle, située entre la fin du PV de recette (15 juillet) et la mise en production (1er août), permet d'absorber d'éventuels retards, de finaliser des tests complémentaires ou de traiter les dernières anomalies critiques. En cas de déroulé nominal, elle pourra également être utilisée pour renforcer la documentation ou effectuer des tests de performance additionnels. Elle joue ainsi un rôle central dans la maîtrise des risques projet et la sécurisation de la qualité du produit livré.

WBS	Type	Tâche	Début imp.	Fin imp.	Lien	Antécédent	Décalage	Progression	Durée	Date début	Date fin
1	Sprint	Sprint 1								01/04/2025	22/04/2025
1.1	Tâche	Développement	01/04/2025						11 jours ouvrés	01/04/2025	15/04/2025
1.2	Tâche	Analyse des exigences			DD	1.1			3 jours ouvrés	01/04/2025	03/04/2025
1.3	Tâche	Conception des tests			FD	1.2	2		3 jours ouvrés	08/04/2025	10/04/2025
1.4	Tâche	Exécution des tests			FD	1.3	3		4 jours ouvrés	16/04/2025	22/04/2025
1.J	Jalon	Bilan	22/04/2025								
2	Sprint	Sprint 2								10/04/2025	12/05/2025
2.1	Tâche	Développement			DD	1.3	2		11 jours ouvrés	10/04/2025	25/04/2025
2.2	Tâche	Analyse des exigences			FD	1.4			2 jours ouvrés	23/04/2025	24/04/2025
2.3	Tâche	Conception des tests			FD	2.2	3		3 jours ouvrés	30/04/2025	05/05/2025
2.4	Tâche	Exécution des tests			FD	2.3			4 jours ouvrés	06/05/2025	12/05/2025
2.J	Jalon	Bilan	12/05/2025								
3	Sprint	Sprint 3								05/05/2025	30/05/2025
3.1	Tâche	Développement			DD	2.3	2		11 jours ouvrés	05/05/2025	20/05/2025
3.2	Tâche	Analyse des exigences			FD	2.4			2 jours ouvrés	13/05/2025	14/05/2025
3.3	Tâche	Conception des tests			FD	3.2	2		2 jours ouvrés	19/05/2025	20/05/2025
3.4	Tâche	Exécution des tests			FD	3.3	3		4 jours ouvrés	26/05/2025	30/05/2025
3.J	Jalon	Bilan	30/05/2025								
4	Sprint	Sprint 4								21/05/2025	13/06/2025
4.1	Tâche	Développement			DD	3.3	2		11 jours ouvrés	21/05/2025	05/06/2025
4.2	Tâche	Analyse des exigences			DD	3.4			2 jours ouvrés	26/05/2025	27/05/2025
4.3	Tâche	Conception des tests			FD	4.2	2		2 jours ouvrés	02/06/2025	03/06/2025
4.4	Tâche	Exécution des tests			FD	4.3	3		4 jours ouvrés	10/06/2025	13/06/2025
4.J	Jalon	Bilan	13/06/2025								
5	Sprint	Sprint 5								04/06/2025	27/06/2025
5.1	Tâche	Développement			DD	4.3	2		11 jours ouvrés	04/06/2025	19/06/2025
5.2	Tâche	Analyse des exigences			DD	4.4			2 jours ouvrés	10/06/2025	11/06/2025
5.3	Tâche	Conception des tests			FD	5.2	2		3 jours ouvrés	16/06/2025	18/06/2025
5.4	Tâche	Exécution des tests			FD	5.3	3		4 jours ouvrés	24/06/2025	27/06/2025
5.J	Jalon	Bilan	27/06/2025								
6	Sprint	Finalisation								03/07/2025	01/08/2025
6.1	Tâche	Non-régression			FD	5.4	3		5 jours ouvrés	03/07/2025	09/07/2025
6.2	Tâche	PV de recette			FD	6.1	1		2 jours ouvrés	11/07/2025	15/07/2025
6.3	Tâche	Mise en production		01/08/2025	FD	6.2	10		1 jour ouvré	01/08/2025	01/08/2025
6.J	Jalon	Mise en production	01/08/2025								



Préconisations

Afin d'assurer le bon déroulement de la campagne de test du projet TOMSEN, plusieurs points de vigilance ont été identifiés. Ces recommandations s'inscrivent dans une démarche préventive, pour anticiper les obstacles potentiels et optimiser la qualité des livrables.

Coordination inter-projets

Problème identifié

Un autre projet doit intégrer l'une de ses fonctionnalités dans l'application testée. À ce stade, les modalités techniques de cette intégration ne sont pas connues.

Risque

Incohérences entre les deux périmètres, conflits d'implémentation, effets de bord non anticipés.

Préconisation

Organiser une réunion technique dédiée dès que les spécifications du second projet seront disponibles. Prévoir un test de compatibilité entre les deux modules dans la campagne de non-régression. Intégrer un jalon intermédiaire pour vérifier l'impact croisé.

Environnement de test

Problème identifié

La stratégie repose sur un environnement de préproduction stable et fidèle à la future production, notamment pour les connexions API bancaires et BDF.

Risque

Tests retardés ou partiellement exécutables si les environnements sont instables, non prêts, ou mal configurés.

Préconisation

Mettre en place une vérification technique de l'environnement avant chaque phase d'exécution. Désigner un responsable technique côté infrastructure. Prévoir un environnement miroir si nécessaire.

Réalisme du planning

Problème identifié

Certains sprints (notamment 4 et 5) sont impactés par des jours fériés et ponts. Des retards de développement pourraient également décaler les phases de test.

Risque

Accumulation de retard, surcharge sur les phases critiques, baisse de la couverture.

Préconisation

Renforcer les marges prévues dans le planning sur les sprints sensibles. Prioriser les tests critiques. Maintenir la flexibilité dans l'organisation des campagnes, en mobilisant notamment la marge prévue entre le PV de recette et la mise en production si nécessaire.

Sécurité des données sensibles

Problème identifié

L'application manipule des données personnelles et sensibles : données bancaires, authentification à double facteur, biométrie, justificatifs.

Risque

Non-conformité RGPD, exposition de données, perte de confiance.

Préconisation

Valider en amont les jeux de données fictifs à utiliser. Prévoir une revue sécurité en fin de campagne avec l'ensemble des parties prenantes. Ajouter un audit spécifique avant la mise en production, notamment sur les aspects RGPD, authentification et stockage des justificatifs.

Automatisation et capacité à faire

Problème identifié

Les scénarios critiques sont nombreux. Certains sont éligibles à l'automatisation, mais celle-ci reste à mettre en œuvre.

Risque

Temps d'exécution trop long si tous les tests restent manuels. Sous-utilisation des ressources techniques.

Préconisation

Lancer un chantier d'automatisation ciblée sur les scénarios de non-régression. Intégrer cette dimension dans les prochaines estimations de charge. Prévoir une première version automatisée avant la fin du sprint 5, en priorisant les scénarios critiques et les parcours de non-régression.

Clarification des exigences restantes

Problème identifié

Certaines exigences manquent de précision (gestion fine du 2FA, options du conseiller virtuel, gestion des erreurs API).

Risque

Tests incomplets ou mal ciblés. Risque d’ambiguïté lors de l'exécution.

Préconisation

Organiser une revue des exigences restantes en début de chaque sprint. Capitaliser sur le document de revue d’exigence pour prioriser les levées de doute avec le PO.