STRATÉGIE DE TEST

---

1. Scénarios prévus

Tableau — Fonctionnalité / Exigences / Scénario prévu

Fonctionnalité	***Exigences***	***Scénario***
Création de comptes d'utilisateurs sécurisés	*Sprint-1 – Critique*	*Pouvoir créer mon compte utilisateur*
Téléversement des pièces justificatives	*Sprint-1 – Majeur*	*Pouvoir téléverser mes pièces justificatives*
Authentification double facteur (2FA)	*Sprint-1 – Majeur*	*Demander une authentification à double facteur pour les nouvelles connexions*
Authentification biométrique	*Sprint-1 – Critique*	*Demander une authentification biométrique*
Interconnexions bancaires	*Sprint-2 – Critique*	*M'interfacer avec les API bancaires françaises*
Récupération des données bancaires (API BDF)	*Sprint-3 – Critique*	*Récupérer les informations bancaires via la BDF avec timeout et limite 5 min*

Consultation des documents bancaires	*Sprint-3 – Majeur*	*Consulter mes informations bancaires synchronisées*
Téléchargement des relevés bancaires	*Sprint-4 – Majeur*	*Télécharger mes relevés bancaires (PDF)*
Consultation des transactions	*Sprint-4 – Critique*	*Consulter mes transactions avec tri et filtres*
Suivi de consommation – Synthèse	*Sprint-5 – Critique*	*Visualiser la synthèse de mes dépenses multi-banques*
Graphique des dépenses (camembert)	*Sprint-5 – Majeur*	*Afficher le graphique des dépenses par catégorie*
Conseiller virtuel	*Sprint-6 – Critique*	*Obtenir des conseils “Consommations” et “Épargne”*
Interaction → Prise de RDV conseiller	*Sprint-6 – Majeur*	*Être redirigé vers la page de prise de rendez-vous*

---

2. Méthodes de test adaptées

Voici le tableau

Scénario lié à la fonctionnalité	Méthode de test	Justification
Création de comptes sécurisés	**Tests automatisés**	Ce parcours est critique et très sensible à la régression. L'automatisation permet de rejouer rapidement tous les contrôles de sécurité (mot de passe, validation des champs, messages d'erreur) à chaque nouvelle livraison, ce qui garantit la fiabilité du processus d'inscription.
Téléversement des justificatifs	**Tests automatisés**	Le téléversement implique des règles strictes (formats, tailles, erreurs serveur). Automatiser ces vérifications assure une validation constante, rapide, et permet de détecter immédiatement une régression sur un format ou une taille de fichier.
Authentification double facteur (2FA)	**Tests automatisés**	Le 2FA est un point de sécurité majeur. Comme toutes les étapes sont répétitives (saisie code, expiration, renvoi), l'automatisation permet de simuler différents scénarios (code valide, expiré, invalide) de manière fiable et continue.
Authentification biométrique	**Tests exploratoires**	La biométrie dépend fortement du matériel (PC, mobile, capteurs). Il est difficile d'automatiser ces tests car chaque appareil peut réagir différemment. Une évaluation exploratoire permet de valider l'intégration réelle et l'expérience utilisateur.
Interconnexions bancaires	**Tests automatisés**	Les appels API sont répétables et doivent respecter des formats stricts. Automatiser ces tests permet de contrôler rapidement les réponses, les statuts HTTP, et la cohérence des données, tout en testant la stabilité des API.
Récupération API BDF (TOM-10)	**Tests API + Automatisés**	Ce scénario nécessite de tester plusieurs éléments : délai maximum (30s), erreurs serveur, format des balises, mise à jour en base. Les tests API sont essentiels pour valider le comportement exact de l'API, tandis que l'automatisation permet de répéter ces contrôles à chaque sprint.

Tableau de bord	**Manuel + Automatisé**	Une partie du tableau (affichage, visibilité, lisibilité) nécessite une validation visuelle manuelle. En revanche, les scénarios répétitifs (ex : rafraîchissement, navigation) peuvent être automatisés pour sécuriser le parcours.
Téléchargement des relevés	**Tests manuels**	Le téléchargement est un comportement lié au navigateur/système de fichier. Il nécessite une validation visuelle du PDF, du nom du fichier et de la cohérence du contenu.
Suivi des consommations	**Manuel + Exploratoire**	Les calculs multi-banques et multi-devises doivent être vérifiés avec précision. Une approche exploratoire permet aussi d'identifier des incohérences ou des limites non documentées.
Graphiques	**Tests manuels**	Les graphiques nécessitent une évaluation visuelle incontournable : couleurs, proportions, légendes, comportement si peu ou pas de données.
Conseiller virtuel	**Tests manuels**	Les tests portent sur la navigation, la visibilité de la bulle et les redirections. Ce sont des interactions UX, difficiles à automatiser de manière fiable.

---

3. Ressources nécessaires

Ressources humaines

- **Consultant Test** : pilote l'ensemble de la stratégie, réalise les tests critiques, remonte les anomalies et suit leur résolution.
- **PO (Elsa)** : indispensable pour lever toutes les ambiguïtés fonctionnelles et valider la conformité métier.
- **Bureau d'étude (Quentin)** : apporte les précisions sur les règles bancaires et API externes.
- **Équipe dev (Alice)** : fournit les versions, corrige les anomalies, et explique les comportements techniques attendus.

Outils employés

- **GitHub** : centralise les versions et facilite l'analyse des changements, important pour comprendre l'origine d'une régression.
- **Jira** : permet de suivre précisément l'évolution des anomalies, leur criticité et leur traitement.
- **Teams** : communication en continu avec le PO et les devs, essentielle en mode Agile.
- **Cypress / Playwright** : permettent de créer des scripts stables pour les parcours critiques (inscription, login, 2FA).
- **Postman** : idéal pour tester les réponses brutes des API bancaires et valider le respect du format.
- **Snagit / Notion** : facilitent le reporting visuel et la traçabilité des tests exécutés.

Environnements

- **Docker** : garantit un environnement identique à chaque livraison, évitant les erreurs dues à la configuration locale.
- **Base SQL / Firebase de test** : permet de manipuler et réinitialiser les données sans impacter la production.
- **Mocks API** : indispensables pour tester même lorsque les API réelles sont indisponibles ou instables.

Jeux de données nécessaires

- Comptes utilisateurs (valide / invalide / âge < 18 / email déjà pris)
- Codes 2FA mockés
- Justificatifs (valide / invalide / tailles différentes)

Données bancaires simulées (banques compatibles, non compatibles, timeout)

---

4. Étapes clés de la stratégie

Phase 1 – Préparation (Semaines 1-3)

- Analyse des spécifications
- Revue d'exigences
- Rédaction de la stratégie de test
- Début du cahier de recette (Sprints 1 & 2)

Phase 2 – Exécution par sprint (Semaines 4–12)

- Conception des tests du sprint suivant
- Exécution des tests du sprint en cours (Sprints 1 à 6)
- Suivi et traitement des anomalies
- GO/NO GO en fin de chaque sprint

Phase 3 – Non-régression (Semaines 13–14)

- Intégration du module de l’équipe externe
- Réalisation de la campagne TNR
- Correction des anomalies éventuelles

Phase 4 – Clôture (Semaines 15–16)

- Rédaction du bilan de test
- Décision GO/NO GO production

- Archivage des livrables

---

5. Préconisations

- **Clarifier les exigences manquantes** : sans réponse claire (stockage, format API), les tests risquent d'être incomplets ou faux.
- **Sécuriser les échanges API** : passer systématiquement en HTTPS + clé API pour réduire les risques.
- **Définir formats/taille max des justificatifs** pour éviter les dérives.
- **Anticiper les mocks API** pour tester même lorsque les services externes sont indisponibles.
- **Automatiser les parcours critiques** pour réduire les risques et accélérer la TNR.
- **Uniformiser la règle d'âge** (≥18 ans, basée sur la date du jour) pour éviter les incohérences.